READ

Recognition and Enrichment of Archival Documents

D7.15
Keyword Spotting Engines: QbE, QbS P3

Ioannis Pratikakis, Konstantinos Zagoris DUTH
George Retsinas, George Sfikas, Basilis Gatos, George Louloudis, Nikolaos Stamatopoulos NCSR
Alejandro H. Toselli, Joan Puigcerver, Enrique Vidal, Verónica Romero, Joan A. Sánchez, UPVLC

Distribution:

http://read.transkribus.eu/

READ
H2020 Project 674943

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943

Project ref no.	H2020 674943
Project acronym	**READ**
Project full title	**Recognition and Enrichment of Archival Documents**
Instrument	H2020-EINFRA-2015-1
Thematic Priority	EINFRA-9-2015 - e-Infrastructures for virtual research environments (VRE)
Start date / duration	01 January 2016 / 42 Months

Distribution	Public
Contractual date of delivery	31.12.2018
Actual date of delivery	
Date of last update	
Deliverable number	D7.15
Deliverable title	Keyword Spotting Engines: QbE, QbS P3
Type	Demonstrator
Status & version	
Contributing WP(s)	WP7
Responsible beneficiary	DUTH
Other contributors	NCSR, UPVLC
Internal reviewers	Joan Andreu Sánchez
Author(s)	Ioannis Pratikakis, Konstantinos Zagoris DUTH George Retsinas, George Sfikas, Basilis Gatos, , George Louloudis, Nikolaos Stamatopoulos NCSR Alejandro H. Toselli, Joan Puigcerver, Enrique Vidal, Verónica Romero, Joan A. Sánchez, UPVLC
EC project officer	
Keywords	Keyword Spotting, Query by Example, Query by String

Table of Contents

Executive Summary

Handwritten keyword spotting is the task of detecting query words in handwritten document image collections without involving a traditional OCR step. Recently, handwritten word spotting has attracted the attention of the research community in the field of document image analysis and recognition since it has been proved to be a feasible solution for indexing and retrieval of handwritten documents in the case where OCR-based methods fail to deliver proper results. This deliverable reports on the achievements concerning the tasks of keyword spotting for handwritten document image collections at the end of the third year of the READ project that have been realized by three (3) distinct frameworks which correspond to partners DUTH, NCSR and UPVLC, respectively.

I. The Query by Example (QbE) Engines

1. Introduction

A promising strategy to deal with unindexed documents is a keyword matching procedure that relies upon a low-level pattern matching called word spotting by example [Manmatha1996]. In the literature, word spotting appears under two distinct strategies wherein the fundamental difference concerns the search space which could be either a set of segmented word images (segmentation-based approach) or the complete document image (segmentation-free approach). 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 which is the common case in historical documents. Nevertheless both strategies use an operational pipeline where feature extraction and matching have prominent roles.

2. DUTH Keyword Spotting Framework

During the third year of the project, DUTH focused on minimizing memory and computational power requirements of year 2's method which it would enable us to search in large document collections.

2.1. Segmentation-Free Keyword Spotting

The focus of the work during the third year has been about minimizing memory and computational power requirements. That was of high priority since it would enable us to search in large document collections. The current method provides some unique advantages that stems from the capacity to search the whole document and not just applying a word segmentation method. Those advantages are:

1. Good handling of complex document layouts.
2. Ability to match partial words or phrases.
3. It can locate not only words but also symbols.

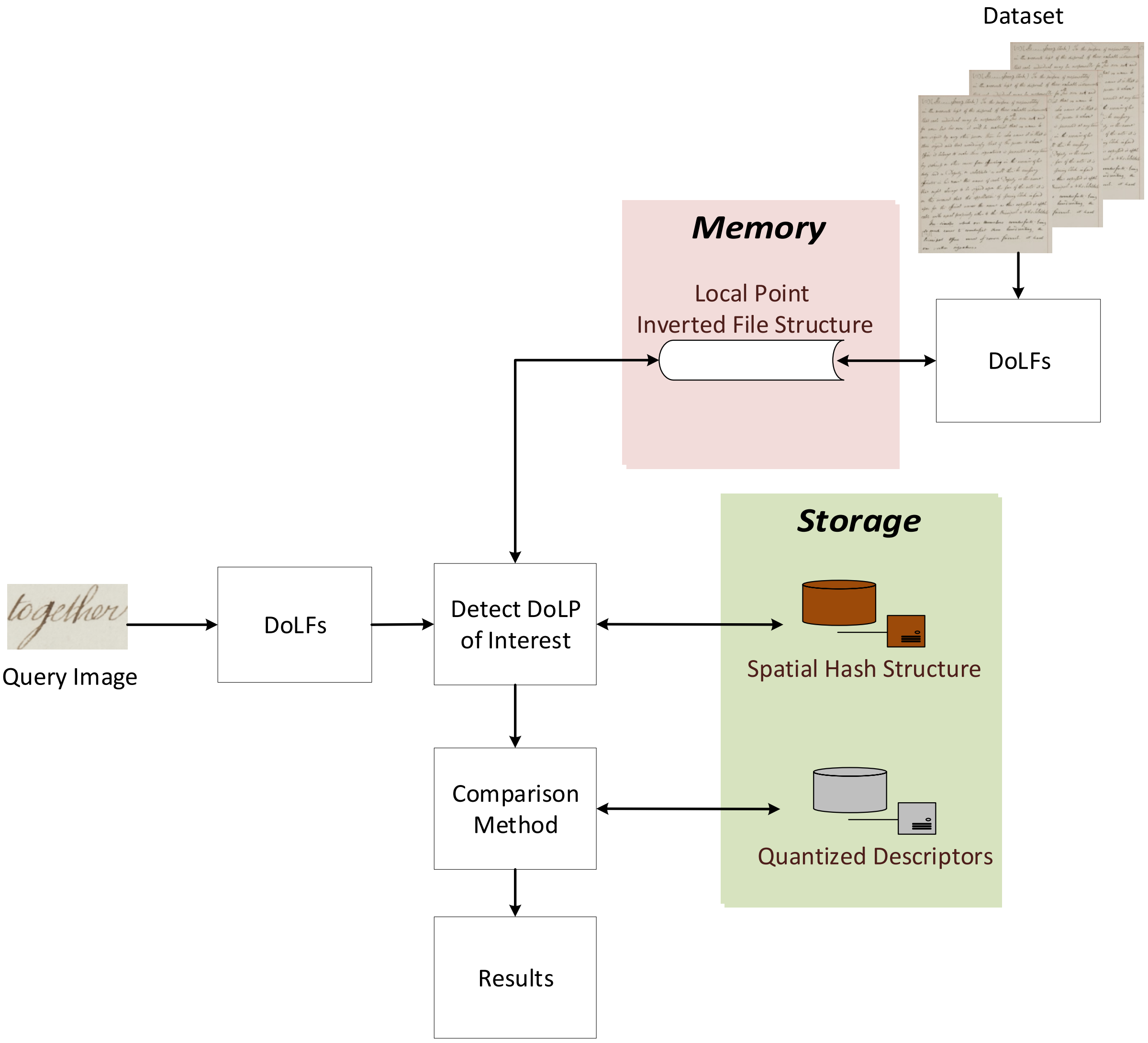

Figure I.2.1 The architecture of the DUTH Segmentation-Free Keyword Spotting.

Therefore, DUTH concreated at the following out aspects:

- Architectural Improvements
  - Implement a spatial hashing structure that encodes both the location and the id of the local point and incorporating it to the comparison method.
  - Implement a different descriptor quantization procedure which allows to store more information while decreasing the quantization time and storage cost.
- Implementation Enhancements:
  - Using a custom solution for storing the quantized descriptors.
  - Use of newer C# constructs such as Span<T> that provide performance parity close to not-managed languages (such as C++).

Figure I.2.1 shows the update year 3's architecture of the DUTH method. It uses Document Oriented Local Points (DoLFs)[ZAG2017] to detect meaningful points on a dataset and one type of Inverted File Structures to describe them which is the only required memory-based data since the DoLF descriptors are quantized and hashed by using a unique hash function as Figure 1.2.1 shows.

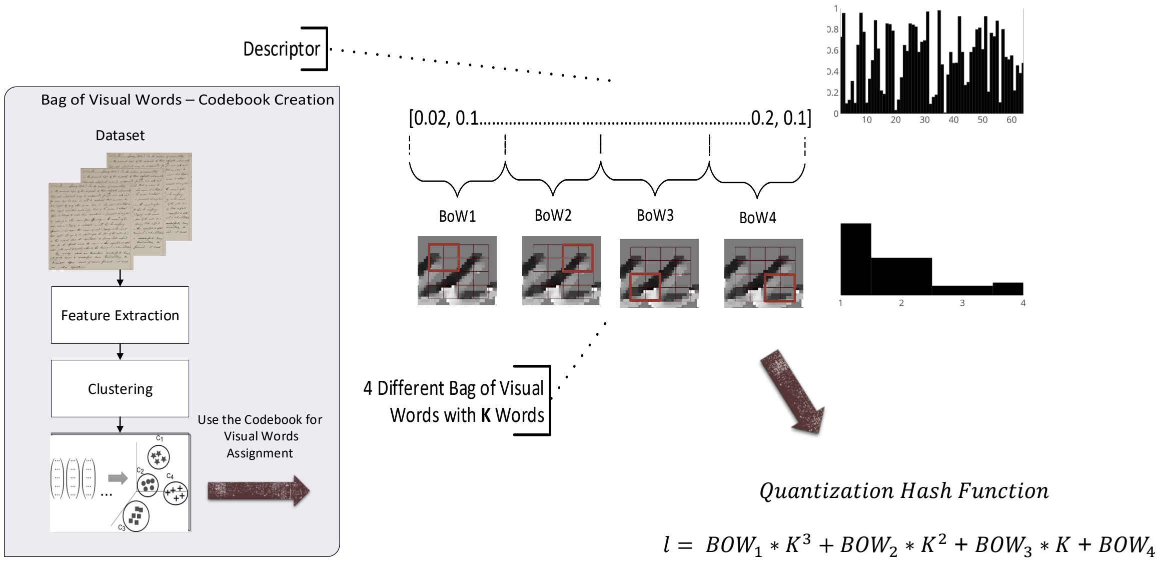
Descriptor
Bag of Visual Words – Codebook Creation
Dataset
Feature Extraction
Clustering
Use the Codebook for Visual Words Assignment
[0.02, 0.1.............0.2, 0.1]
BoW1
BoW2
BoW3
BoW4
4 Different Bag of Visual Words with K Words
Quantization Hash Function
$l = BOW_1 * K^3 + BOW_2 * K^2 + BOW_3 * K + BOW_4$

Figure I.2.2 The quantization architecture.

When the user searches for a word, the DoLFs are calculated and based on Invert File Structure the most meaningful DoLFs are identified and retrieved from the storage. Finally, for comparative purposes the efficiency of the proposed method compared to the original method [ZAG2017] is used to draws the final conclusions. Section 4.2 describes the experimental results.

2.2. KeyWord Spotting Demonstrator

In order to showcase the above segmentation-free word spotting method, a cross-platform word-spotting application was created. It is based on Angular 5, Material Design and Electron frameworks for the front-end (GUI) and the back-end is created by the C#/.NET Core framework.

The KeyWord Spotting Demonstrator supports the following main tasks:

- Creation (Indexing) of new Datasets.
- User interactive word image query selection.
- Presentation of the spotted words.

Figure I.2.2 shows some representative screenshots.

Moreover, the communication between the front-end and the back-end is defined by a REST API which is freely available at:

https://github.com/Transkribus/WSBackend-API

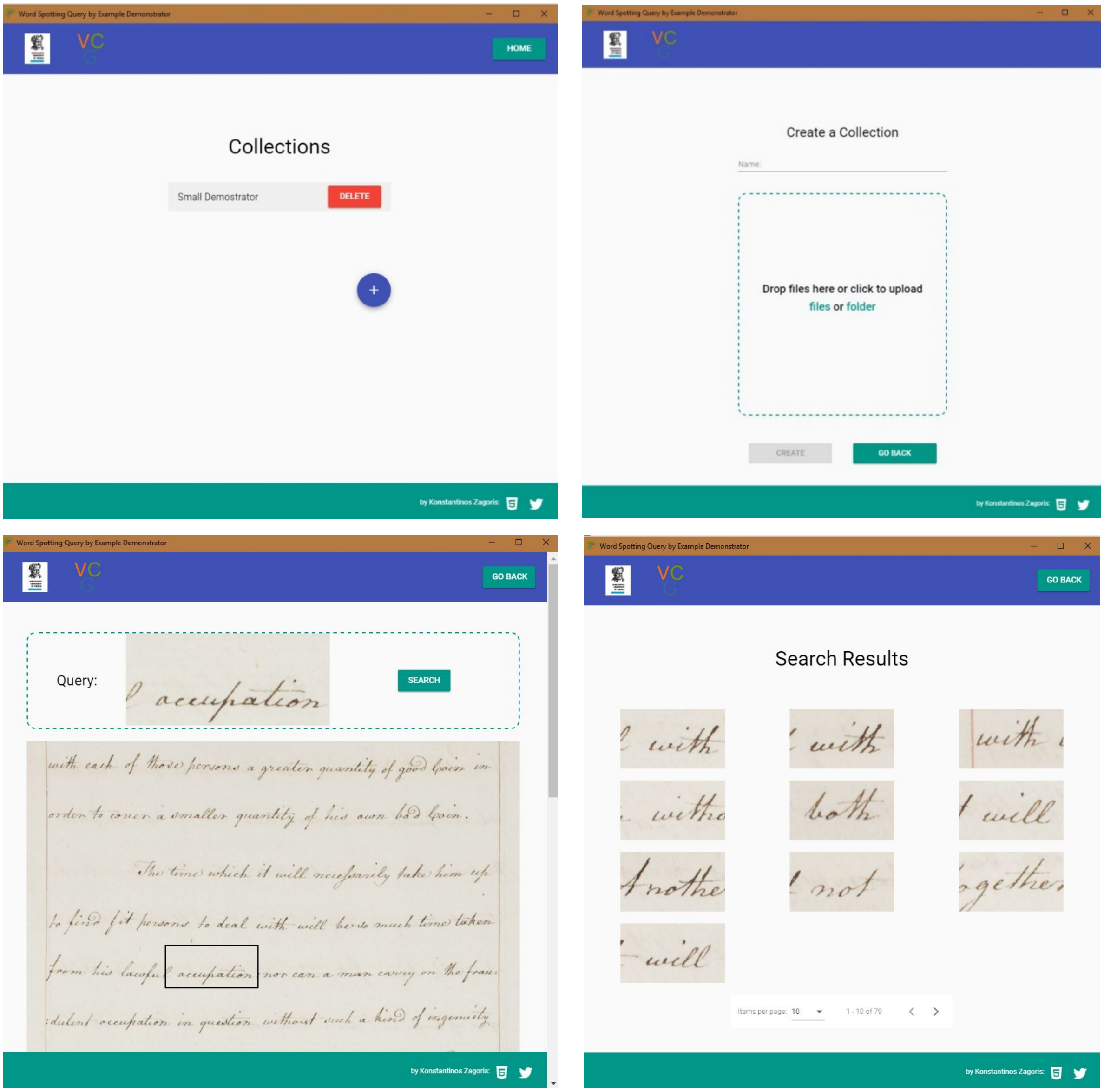

Figure I.2.2 Screenshots of the KeyWord Spotting Demonstrator.

2.3. Evaluation

The presented methodologies are evaluated on three datasets (Figure I.4.1):

- English Dataset which contains 115 Pages and 15923 words
- Konzilsprotokolle (German) Dataset which contains 100 Pages and 15579 words
- Finnish Dataset which contains 56 pages and 16201 words

Please note that the punctuation marks and capitals are considered in the ground truth corpora.

The queries consist of words with length greater than 3 and frequency greater than 2. Therefore, the English dataset query set size is 4790, Konzilsprotokolle dataset query set size is 7119 and the Finnish is 5731.

The performance of the word spotting methods was recorded in terms of the Precision at Top 5 Retrieved words (P@5) as well as the Mean Average Precision (MAP) [Pratikakis2014]. Time and memory requirements are recorded in terms of the following metrics: Retrieval Time per Query, Memory requirements per Document, and Storage requirements per Document.

The evaluation of DUTH methods is performed on an 8-core Intel i7-4770K at 3.50GHz with 16Gb of RAM for parallel computation (4 cores). All DUTH methods are currently implemented in C#/.NET.

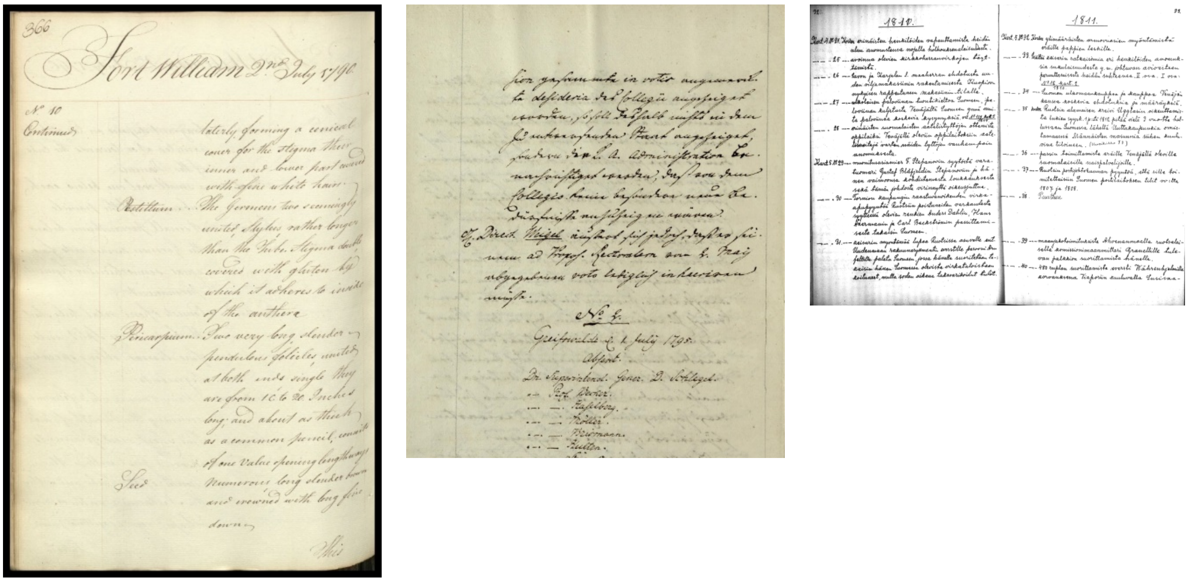

Figure I.4.1 Example documents from the English (left), Konzilsprotokolle (middle) and Finnish (right) Datasets

2.4. Conclusive remarks

Table I.4.2.1 shows the segmentation-free evaluation results for the original method [Zagoris2017], as well as for the methods 'DUTH-M12', 'DUTH-M24', 'NCSR-M12', 'NCSR-M24' and new 'DUTH-M36'. The time, memory and storage requirements are presented in Table I.4.2.2 by averaging the corresponding metrics over the three datasets.

Table I.4.2.1 Experimental Results Segmentation-Free Evaluation

Method	English		Konzilsprotokolle		Finnish	
	P@5	**MAP**	**P@5**	**MAP**	**P@5**	**MAP**
Original[ZAG2017]	0.35	0.22	0.59	0.42	0.58	0.43
DUTH M12	0.38	0.25	0.46	0.24	0.35	0.23
DUTH M24	0.34	0.22	0.51	0.27	0.55	0.35
DUTH M36	0.35	0.22	0.57	0.38	0.56	0.39
NCSR-M12	0.36	0.36	0.64	0.54	0.67	0.62
NCSR-M24	0.44	0.42	0.77	0.66	0.76	0.75

Table I.4.2.2 shows that the ‘DUTH-M36’ method manages to keep the same or in some cases achieve better performance than ‘DUTH-M24’ with a big reduction in memory requirements enabling the capability to search in large datasets. Moreover, ‘DUTH-M36’ has the same performance with our original method at the P@5 in much less query time and memory requirements.

Table I.4.2.3 shows the performance of ‘DUTH-M24’ and ‘DUTH-M36’ in relation to the dataset size. The results reveal that the retrieval time per query is increased in a non-linear manner thus making search feasible in terms of time consumption for large scale datasets.

Table I.4.2.2. Time, Memory and Storage Requirements for Segmentation-free Scenario

Method	Retrieval Time per Query (sec)	Memory requirement per Document (KB)	Storage requirement per Document (KB)
Original	15.84	19800	19800
DUTH M12	0.36	1410	1410
DUTH M24	0.67	366	2187
DUTH M36	0.66	49	1676
NCSR-M12	0.0080	101	101
NCSR-M24	0.0653	424	424

Table I.4.2.3. Comparative Evaluation Results for big datasets for Segmentation-free Scenario using **DUTH-M24** and **DUTH-M36** method.

Dataset (Documents)	Retrieval Time per Query (sec)		Overall Memory requirement (MB)		Overall Storage requirement (MB)	
	DUTH-M24	DUTH-M36	DUTH-M24	DUTH-M36	DUTH-M24	DUTH-M36
50	0.47	0.61	12	2.1	35	69
5000	1.55	0.89	1125	213	3438	2693
50000	3.21	1.1	11648	448	34966	5843

3. NCSR Keyword Spotting Framework

On the third year of the READ project, NCSR mainly focused on exploring deep learning (Convolutional Neural Networks - CNNs) for the task of keyword spotting, providing outstanding results. NCSR developed/explored the following KWS subtasks: 1) Segmentation-based QbE applied on word images 2) Segmentation-free QbE (input corresponds to the whole document image) 3) QbE and QbS applied on text line images using a single framework for both modalities.

3.1. Word-level QbE KWS

The NCSR method developed during the second year of the READ project (NCSR-M24) was the method achieving the best performance for the QbE segmentation-based KWS scenario, evaluated not only on the READ datasets but also on the keyword spotting competitions [RET2017a]. In order to evaluate the capability of deep learning techniques, NCSR developed a CNN-based feature extraction method combined with manifold learning approaches for dimensionality reduction [RET2017b]. The backbone CNN was trained on the publicly available IAM dataset (http://www.fki.inf.unibe.ch/databases/iam-handwriting-database), which consists of English text written by 657 writers. The evaluation was performed on the KWS Competition of ICFHR 2014 [PRA2014] which is completely unrelated to the training set. Table I.3.1 presents comparative experimental results of i) state-of-the-art approaches, ii) NCSR-M24 method and iii) original DUTH method [ZAG2017] which is the best method in terms of performance among DUTH methods developed during the READ project. The results indicate that the use of CNN as a feature extraction method creates discriminative features for handwritten text recognition. These features can be used efficiently for different languages (ICFHR14 KWS Modern dataset contains 4 different languages) or historical documents (ICFHR14 KWS Bentham dataset) without re-training. The reported performance is assisted by a non-linear dimensionality reduction method (for further details see [RET2017b]) which creates an embedding of 5 dimensions (the dimensions are pre-defined). As a result, each word image is stored as a 5-d feature vector. Due to the low dimensionality of the feature space, storage requirements are greatly reduced (e.g. for a document collection of 10.000-word images, only 1.6 MB is required). Retrieval time can also be significantly reduced since the low dimensionality of the feature vectors allows for an efficient indexing. The latter is currently being investigated by the NCSR group.

NCSR also explored the use of CNNs on a segmentation-based KWS scenario by experimenting on different architecture choices of the neural network and alternative losses [RET2018a] [RET2018b].

Table I.3.1 Experimental results on ICFR14 KWS dataset (Segmentation-based QbE)

Method	Bentham	Modern
	MAP	MAP
Kovalchuck[PRA2014]	52.4	33.8
Almazan[PRA2014]	51.3	52.3
Howe[PRA2014]	46.2	27.8
DUTH[ZAG2017]	60.0	-
NCSR-M24	71.1	49.1
NCSR-M36 (CNN)	**87.8**	**91.1**

3.2. Segmentation-free QbE

Although the segmentation-based scenario is very practical for the development of reliable word image representations, it could not be applied to real world keyword spotting applications since until recently most of these methods work and report results using correctly segmented words (ground truth). However, NCSR developed a segmentation pipeline which made the generation of word regions a simple task and at the same time it has proven to be a compelling alternative of the time-consuming segmentation-free techniques. During the second year of the READ project, NCSR KWS methods were applied on entire document images using the abovementioned NCSR segmentation pipeline in order to produce candidate word regions which are used as input to the NCSR KWS system (the NCSR segmentation pipeline was presented on deliverable D6.10).

It should be noted that the aforementioned pipeline produces unique candidate regions, which ideally correspond to the actual words of the document. However, this hard assignment of document image parts to words is subjectable to possible errors which affect the final word spotting performance. The aforementioned erroneous procedure can be avoided by using multiple hypotheses of candidate regions for each word starting from the initial document image. This idea is presented in deliverable D6.12 where a novel word segmentation method is presented which not only produces multiple hypotheses for the words that compose the document but also works directly on the initial document image without needing any prior segmentation. The latter is the reason for the significant reduction of the processing time which is necessary for the production of the final word segmentation result.

The evaluation of segmentation-free KWS is reported at Table I.3.2, where a comparison is made among i) DUTH's best method so far, ii) NCSR's best KWS method using the segmentation pipeline described in deliverable D6.10 (NCSR-M24 + segm. pipeline) and iii) NCSR's best KWS method using the novel word segmentation method developed during the third year of the READ project (deliverable D6.12 - NCSR-M24 + candidate regions). The usage of the candidate regions constantly improves the MAP metric, at the expense of a slight reduction of the P@5 metric. The greatest improvement was observed at the English dataset, which was the most challenging with many erroneous segmentations.

Concerning the KWS time and memory requirements of the new segmentation-free pipeline, they are proportional to the number of candidate regions, since each region is a possible word match. Experiments on all three datasets provide on average three times the actual number of words per document, i.e. each "real" word has 3 candidate regions. Therefore, the retrieval time per document, as well as the storage requirements per document are proportional to the number of candidate regions (see D6.12).

Table I.3.2 Segmentation-Free Evaluation

Method	English		Konzilsprotokolle		Finnish	
	P@5	**MAP**	**P@5**	**MAP**	**P@5**	**MAP**
DUTH[ZAG2017]	0.35	0.22	0.59	0.42	0.58	0.43
NCSR-M24 + segm. pipeline	0.44	0.42	0.77	0.66	0.76	0.75
NCSR-M24 + candidate regions	0.44	0.52	0.72	0.71	0.69	0.76

3.3. Line-level QbE & QbS KWS

One of the main concerns of NCSR during the 3rd year is to efficiently retrieve words using both query scenarios (QbE and QbS) using a single framework. Following the success of text-line segmentation on the READ project (see D6.12), we assume line-segmented images as input.

The main idea behind this new approach is based on the success of the CNN features, as it was mentioned previously. The goal is to transform a line image into a discriminative feature space, where a simple template matching can be performed successfully.

To this end, three distinct neural networks were trained:

1. Width Estimator: estimate average character width in an image (word or line image) with the use of a regression CNN. Using the estimation, one can rescale the images in order to have a fixed (pre-defined) average character width. This is important in order to known beforehand the corresponding width of a query into the image (e.g. the query "cat" corresponds to circa 40 pixels width). Such knowledge simplifies the matching procedure.

2. Feature Extractor: extract discriminative features over a text image (word or line) using a CNN trained on word images with Pyramidal Histogram of Characters (PHOC) as label. Only the convolutional layers are necessary for this part, since they can be applied on arbitrary sized images [RET2018a], [RET2018b].

3. Encoder: map a PHOC representation of a string into the same feature space created by the Feature Extractor using a neural network consisted of fully connected layers. This network enables QbS on the generated feature space.

It is important to distinguish the offline operations (the computation and storage of line features) and the online operations (query features and matching procedure):

Offline: Transform each line image into a sequence of features and store them. This is the line representation. (**Width Estimator + Feature Extractor**)

Online: If the query is a word image (QbE) use the feature extractor, alternatively if the query is a string (QbS) use the encoder. Both approaches result to the production of a single feature vector. Moreover, estimate the number of line features that correspond to the query (depends on the number of characters and the average character width of the width estimator). The matching procedure is then simplified into a kNN search (using the cosine distance) on the sequence of line-features. (**QbE : Width Estimator + Feature Extractor, QbS : Encoder**)

The overall pipeline is presented at Figure I.3.1. The method is described in detail in the submitted paper [RET2018c].

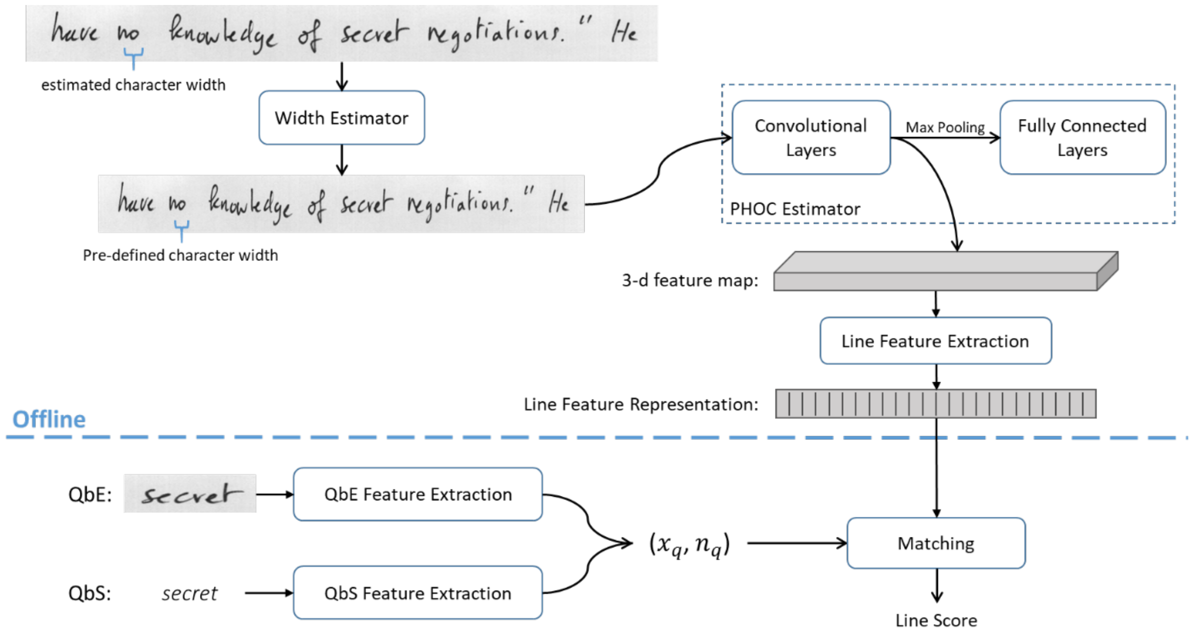

Figure I.3.1 Overview of line-level QbE and QbS KWS system.

This approach was evaluated using the IAM dataset (http://www.fki.inf.unibe.ch/databases/iam-handwriting-database) and the corresponding results are presented on Table I.3.3. In order to be comparable with the results reported in the bibliography, we follow the two most widely used KWS setups for the IAM dataset. **IAMDB1**: 882 queries selected using all non-stop words that appear at least once in the training set as well as the test set. **IAMDB2:** All non-stop words among the 4000 most frequent words that also occur in the training set are selected as queries, resulting in 3421 queries in total. In addition, for measuring the performance of the KWS system two possible scenarios were considered: **local** (a local threshold is used for each keyword separately) and **global** (a global threshold is used that is independent of the keyword). Note that the QbE scenario is only available on the IAMDB1 setup, since IAMDB2 setup contains queries that are out of vocabulary.

Table I.3.3 Experimental results for line-level KWS system (MAP %).

	IAMDB1		**IAMDB2**
methods	local	global	global
Fischer [FIS2012]	68.92	47.75	-
Toselli [TOS2016]	-	-	72.00
Frinken [FRI2012]	-	-	76.00
NCSR-QbS	88.73	83.15	75.31
NCSR-QbE	84.25	73.16	-

For the QbS scenario, it can easily be observed that the accuracy of the NCSR method is on par with the state-of-the-art method of Frinken et al. (**IAMDB2** column). However, the proposed method outperforms the method of Fischer et al. (**IAMDB1** columns). The great achievement of the proposed method concerns its astonishing performance for the QbE scenario (comparable to the QbS case). It should be noted that to the best of the authors' knowledge, it is the first KWS method who manages to apply the QbE scenario on line-level segmented images.

Finally, it should be stressed that this approach can be efficiently used on a large-scale scenario mainly due to its storage and time efficiency. For clarity, we report some indicative storage and time requirements on the IAM dataset: the line image features, computed offline, require 260KB storage (without any quantization), while it takes around 0.28 msec to compare a query to a line. This means that a document page consisting of 100 lines requires 26MB storage space and the related words in the page are retrieved in 28 msec.

II. The Query by String (QbS) case

1. Query by String (QbS) KWS work at UPVLC

In order to provide fast and effective textual access to large collections of handwritten images we compute "probabilistic indexes" (PIs), which allow for very accurate and efficient keyword spotting on non-transcribed handwritten images. See READ deliverable D7.14 for details.

Preparatory experiments aiming at probabilistically indexing large scale collections were conducted to optimize parameters of optical and language models. The following three collections were considered: Bentham Papers, Teatro del Siglo de Oro (TSO), PASSAU. On the base of these experiments, two of these collections, Bentham Papers and TSO, were indexed, as described in Deliverable D8.12. The resulting search interfaces are publicly available at http://prhlt-carabela.prhlt.upv.es/bentham and http://prhlt-carabela.prhlt.upv.es/tso.

In addition, preliminary work on probabilistic indexing and search for melody patterns in handwriting sheet music images, were carried out on the VORAU-253 Manuscript, written in German gothic notation.

1.1. Bentham Papers Experiments

18th-19th century manuscripts written in English by several hands. The full collection contains more than 80K images. See examples in Figure II.1. From this vast collection a representative, ground-truth dataset was compiled. basic statistics of the training and test partitions of this dataset are shown in Table II.1. Two test sets were used for evaluation; one referred to as "Easy", composed of images from several hands with relatively neat handwriting and the other, referred to as "Hard", containing only Bentham's hand draft manuscripts, extremely difficult for read even for expert paleographers.

Experimental conditions:

- Transkribus URO's line detection carried out by UIBK
- *Laia* Convolutional + RNN character optical modeling,
- Character 8-gram language model, trained on an external text (~1 million words)
- Transliteration: text and keywords were case-folded, diacritics-folded, etc.

Transcription accuracy (HTR) results are shown in Table II.2 and indexing and search precision-recall performance can be seen in Figure. II.2

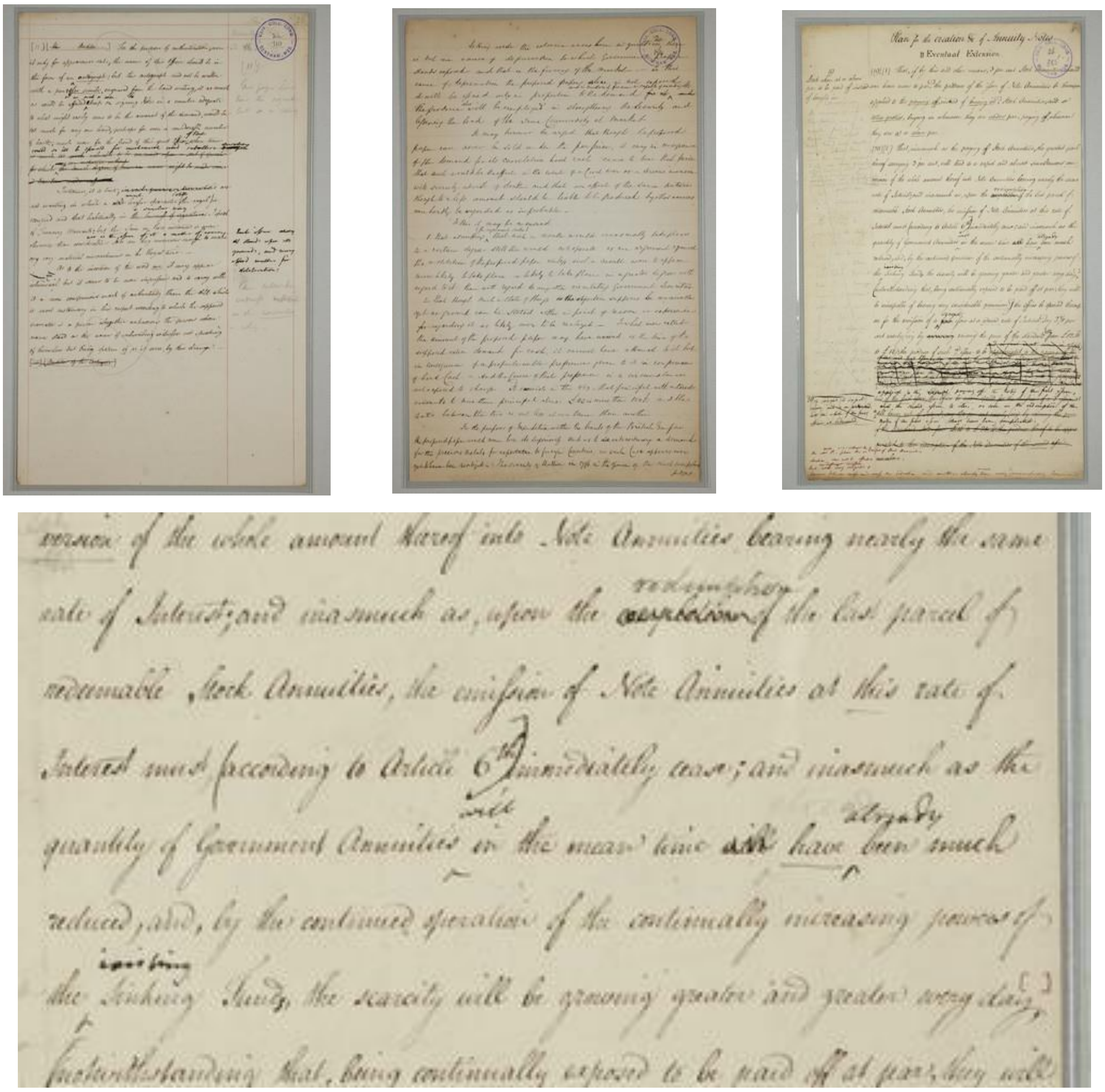

Figure II.1 Example of Bentham Papers images

Table II.1 Bentham papers experimental (transliterated) dataset. †Ignoring punctuation marks.

Manuscripts	Train-Val	Test-Easy	Test-Hard
Pages	846	212	155
Lines	23942	6440	5923
Running Words†	177692	48052	41398
Lexicon†	11510	5498	3744
Character set size	67	65	54

Table II.2 Betham papers HTR results: Character Error Rate (CER) and (transliterated) Word Error Rate (WER) both in %.

Test	CER	WER	CER8–gr	WER8–gr
Easy	6.8	14.4	6.1	10.5
Hard	18.1	38.2	15.5	26.4

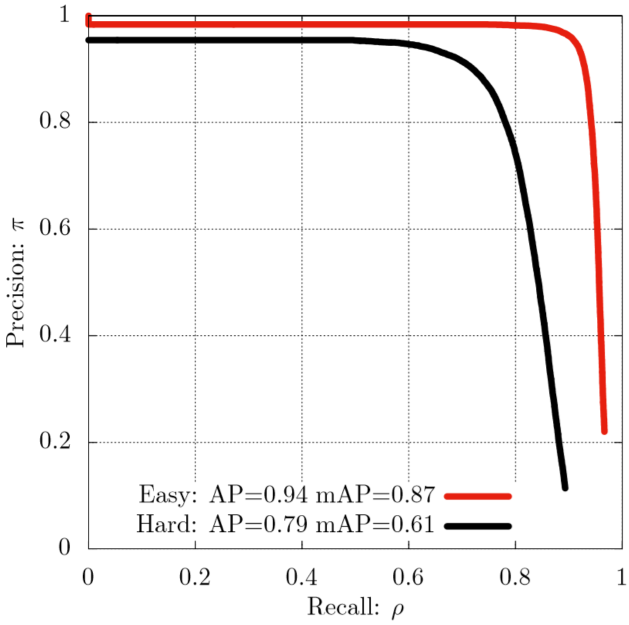

Figure II.2 Bentham papers Precision-Recall performance.

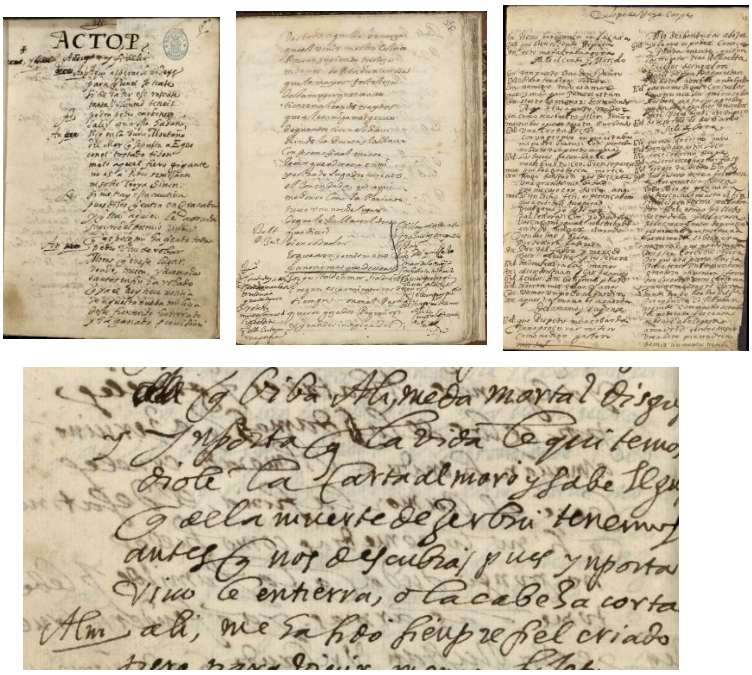

Figure II.3 Example of TSO images

Table II.3 TSO experimental dataset.

Manuscripts	**RES-166**	**RES-168**	**Total**
Pages	141	145	286
Lines	3320	3563	6883
Running Words	19801	23190	42991
Lexicon	3928	3813	6289
Character set size	89	86	92
OOV Characters	6	3	-

Table II.4 TSO CER and WER, both in %

CER	**WER**	**CER$_{8\text{-}gr}$**	**WER$_{8\text{-}gr}$**
26.3	58.8	23.2	48.8

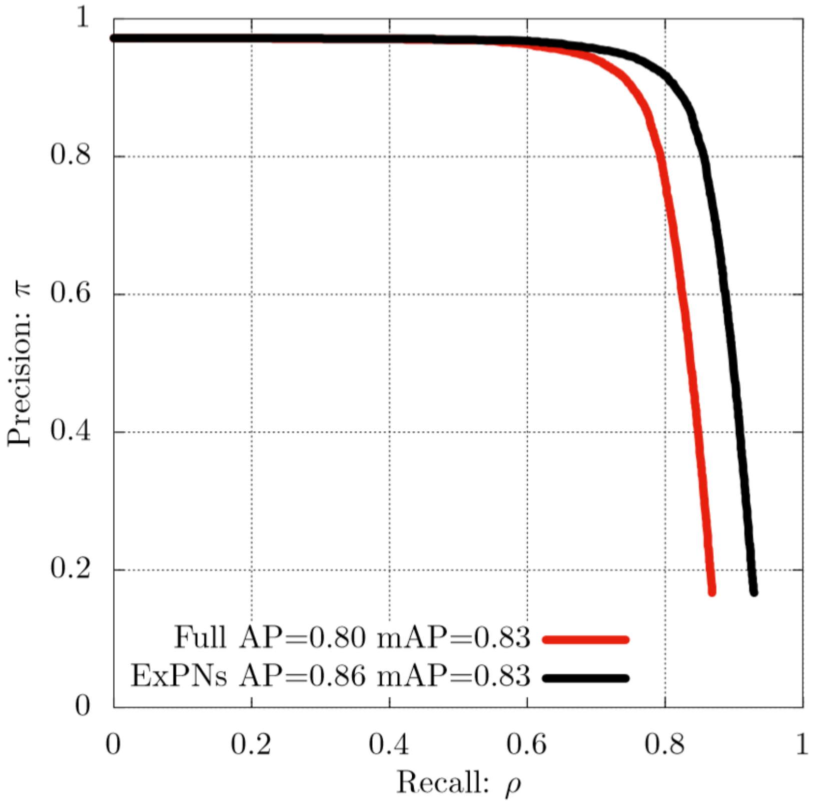

Figure II.4 TSO Precision-Recall performance.

1.2. TSO Experiments

15th-16th century manuscripts. More than 100K images, many hands. Example images are shown in Figure II.3. An experimental dataset of 286 images of two Lope de Vega's comedies was produced by ProLope and UPVLC with GT detected text lines and transcripts. Basic information of this dataset is reported in Table II.3.

Experimental conditions:

- *Laia* Convolutional + RNN optical models
- Character 8-gram language models trained on many TSO comedies
- Transliteration: text and keywords were case-folded, diacritics-folded, etc.
- Keywords (all with length > 1):
    - Full: all the test-set words (5 409)
    - ExPNs: excluding *personae names* (5 355)
- All the results obtained by 10-fold cross validation partition of the dataset.

Transcription accuracy (HTR) results are shown in Table II.4 and indexing and search precision-recall performance can be seen in Figure II.4.

1.3. PASSAU Experiments

Experiments like those carried out for the Bentham Papers and the TSO collections were carried out using a representative dataset of the large PASSAU collection of handwritten German parish records.

Probabilistic Indices may become prohibitively large for vast manuscript collections. Therefore, using this dataset we also analyzed simple index pruning methods to achieve adequate tradeoffs between memory requirements and search performance. We also studied how to adequately deal with the large variety of non-ASCII symbols and handwritten word spelling variations (accents, umlauts, etc.) which appear in this kind of historical collections.

This work was published in the proceedings of 2018 ICFHR and we refer to this paper for more details:

> Eva Lang, Joan Puigcerver, Alejandro H. Toselli and Enrique Vidal. "Probabilistic Indexing and Search for Information Extraction on Handwritten German Parish Records". In "Proceedings of the 16th International Conference on Frontiers in Handwriting Recognition (ICFHR 2018)". Pages 44-49, Niagara Falls, USA (August 2018). Published by IEEE Computer Society, ISBN-13: 978-1-5386-5875-8.

A demonstration search interface for the (small) test set of this dataset is publicly available at http://transcriptorium.eu/demots/kws-Passau.

1.4. Experiments with Handwritten Sheet Music Images

In this work we explored probabilistic indexing approaches for fast and accurate retrieval of music symbol sequences, representing melodic patterns, from collections of early music manuscripts. As a first test-bed collection we used the music manuscript referred to as Cod. 253 of the Vorau Abbey library, which was provided by the Austrian Academy of Sciences. It is written in German gothic notation and dated around year 1450. Figure II.5 show example images of the VORAU-253 manuscript.

Music symbol recognition: Experimental conditions:

- Automatic stave segmentation provided by Transkribus tools (which not always provided accurate results in this dataset)
- Convolutional + RNN optical music symbol model, trained with TensorFlow + 2-gram symbol language model estimated from training sequences of tokens representing vertical symbol positions within the staves.
- Query sets:
  - *Single symbols*: all the 15 symbols seen in the test set.
  - *Symbol sequences:* all the 615 sequences with lengths ranging from 3 to 15 which appear in the test set more than once.
- Precision-Recall performance evaluated at stave level for sequence queries and at relative symbol position level for single symbol queries.

Transcription accuracy results measured in terms of Symbol Error Rate (SER) are shown in Table II.6 and indexing and search precision-recall performance can be seen in Figure II.6.

A melody pattern search interface which demonstrates this approach is publicly available at: http://prhlt-carabela.prhlt.upv.es/music.

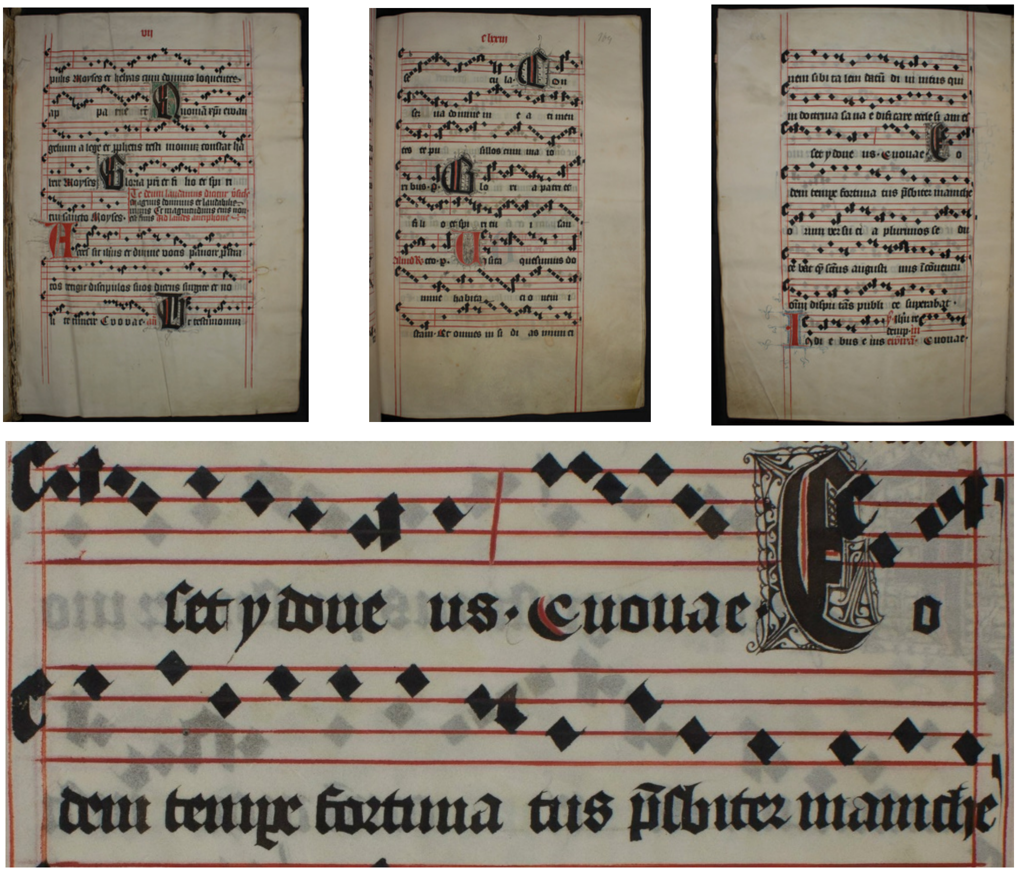

Figure II.5 Examples of page images of Vorau music manuscripts.

Table II.5 The VORAU-253 Dataset basic statistics

Manuscripts	**Train-Val**	**Test**
Pages	422	44
Staves	1000	97
4-line staves	882	97
5-line staves	118	0
Running symbols	13066	1086
Symbol set size	19	15
Vertical positions	12	9
Clefs, alterations, etc.	7	6

Table II.6: VORAU-253 Symbol Error Rate (SER) in %

SER	**SER$_{2\text{-}gr}$**
6.63	5.62

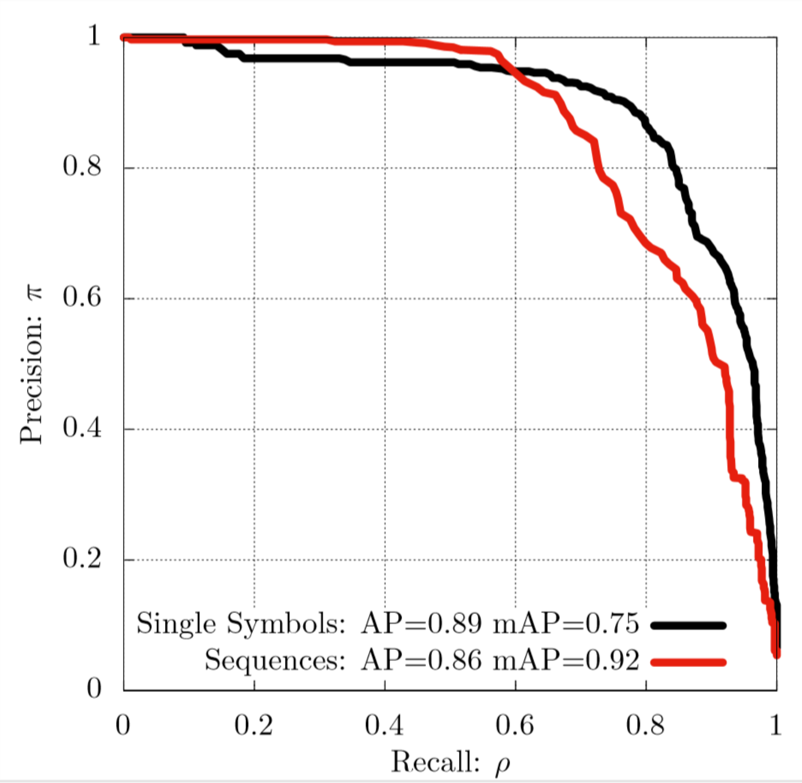

Figure II.6 VORAU-253 Precision-Recall performance.

2. Query by String (QbS) KWS work at NCSR

NCSR also developed a QbS framework, as a part of a line-level Keyword Spotting framework which can be used for both QbE and QbS scenarios. Details and results of this KWS approach are presented at section I.3.3.

References

[FIS2012] A. Fischer, A. Keller, V. Frinken, and H. Bunke, “Lexicon-free handwritten word spotting using character hmms”, Pattern Recognition Letters, 2012.

[FRI2012] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, “A novel word spotting method based on recurrent neural networks”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012.

[PRA2014] I. Pratikakis, K. Zagoris, B. Gatos, G. Louloudis and N. Stamatopoulos, “ICFHR 2014 Competition on Handwritten KeyWord Spotting (HKWS 2014)”, International Conference on Frontiers in Handwriting Recognition (ICFHR), 2014.

[RET2017a] G. Retsinas, G. Louloudis, N. Stamatopoulos and B. Gatos, “Efficient Learning-Free Keyword Spotting”, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[RET2017b] G.Retsinas, G.Sfikas and B.Gatos, "Transferable Deep Features for Keyword Spotting", International Workshop on Computational Intelligence for Multimedia Understanding, EUSIPCO 2017

[RET2018a] G. Retsinas, G. Sfikas, N. Stamatopoulos, G. Louloudis and B. Gatos, “Exploring critical aspects of CNN-based Keyword Spotting. A PHOCNet study”, International Workshop on Document Analysis Systems (DAS), 2018.

[RET2018b] G. Retsinas, G. Sfikas, G. Louloudis, N. Stamatopoulos and B. Gatos, “Compact Deep Descriptors for Keyword Spotting”, International Conference on Frontiers in Handwriting Recognition (ICFHR), 2018.

[RET2018c] G. Retsinas, G. Louloudis, N. Stamatopoulos, G. Sfikas and B. Gatos, “An Alternative Deep Feature Approach to Line Level Keyword Spotting”, submitted to CVPR 2018.

[TOS2016] A. H. Toselli, E. Vidal, V. Romero, and V. Frinken, “Hmm word graph based keyword spotting in handwritten document images”, Information Sciences, 2016.

[ZAG2017] Zagoris K, Pratikakis I, Gatos B. Unsupervised Word Spotting in Historical Handwritten Document Images using Document-oriented Local Features. IEEE Transactions on Image Processing. 2017