UAB
Universitat Autònoma de Barcelona

Texture of analysis for Robust Reading Systems

Angelos Nikolaou

Universitat Autònoma de Barcelona

Texture of analysis for Robust Reading Systems

Angelos Nikolaou

Computer Vision Center

This dissertation is submitted on September, 2020 for the degree of Doctor of Philosophy

Angelos Nikolaou
September, 2020

Abstract

Texture of analysis for Robust Reading Systems

Angelos Nikolaou

This thesis focuses into the use of texture analysis for Robust Reading Systems. In this thesis the use of texture analysis for text-image is explored. An in depth analysis of the established Local Binary Pattern descriptor is presented. The LBP descriptors are used in word-spotting and achieves top performance among learning-free methods. A custom variant called Sparse Radial Sampling LBP is developed to exploit the unique properties of text and is used to achive state-of-the-art performance in writer identification. The same feature descriptors are used in conjunction with deep Neural Networks in order address successfully the problem of script and language identification in multiple modalities.

Acknowledgements

I would like to thank everybody!

Contents

Glossary

ANN Artificial Neural Network.

BCN Barcelona Historical Handwritten Marriages Database.

BoVW Bag of Visual Words.

CNN Convilutional Neural Networtk.

CV Computer Vision.

CVSI Competition on Video Script Identification.

DIA Document Image Analysis.

DR Detection Rate.

DTW Dynamic Time Warping.

GMM Gaussian Mixture Model.

GSCM Grey-Scale Co-ocurrence Matrices.

GW George Washington.

HOG Histogram of Oriented Gradients.

ICDAR International Conference in Document Analysis and Recognition.

IoU Intersection over Union.

LBP Local Binary Patterns.

LPQ Local Phase Quantization.

LSTM Long Short Term Memory.

MLP Multi-Layered Perseptron.

MSPN Multi-stage Spatially-sensitive Pooling Network.

OCR Optical Character Recognition.

PCA Principal Component Analysis.

PDF Propability Density Function.

PHOC Pyramidal Histogram of Characters.

QBE Query by Example.

QBS Query by String.

QOS Quality of Segmentation.

RA Recognition Accuracy.

SIFT Scale-Invariant Feature Transform.

SotA State-of-the-Art.

SRS-LBP Sparse Radial Sampling Local Binary Patterns.

SVM Support Vector Machine.

VLAD Vector of Local Aggregate Descriptor.

Chapter 1

Introduction

1.1 Forward

In this thesis we propose the use of texture analysis as a general framework for extracting usefull information about text in images. This thesis has been realised in large span of time and in the midst of the deep learning evolution, it therefore is important to read this work in the context of dynamic and evolving field and contextualise the material in the time it was presented to the public.

The structure of this thesis is as follows:

- Chapter 1: In this chapter, we discuss the nature of text and our motivation in using texture analysis for it.
- Chapter 2: In this chapter, we dive into texturea analysis and more specifically to the specifics of LBP.
- Chapter 3: In this chapter, we investigate word spotting in general and propose an LBP method for word spotting.
- Chapter 4: In this chapter, we address writer identification and propose a general texture descriptor specialised for text.
- Chapter 5: In this chapter, we employ our texture analysis in script identification with use of deep MLP
- Chapter 6: In this chapter we conclude and sumarize our findings.

1.2 The nature of text

With respect to most other objects that are analysed by Computer Vision, text stands out as the principal example of an object that has evolved over milenia with visibillity, recognition, and saliency as some of the principal goals. Writing has evolved from drawing. Phonetic alphabets evolved from the notion of drawing homonym's initially, then gradually associating symbols with with sylables and finaly phonems[44]. The medium of writing had a profound effect on the evolution of text. With the exception of egyptian hieroglyphs, color does not have any specific meaning for a writing system and does not affect text transliteration.

1.2.1 The Visual Archetype

I propose the notion of the visual archetype a means of thinking about text in abstract and it's form.

The visual archetype is the ideal, intended, immaterial form of the document. Depending on the modality, handwritten, typeface printing, desktop publishing, etc., the visual archetype is different things. In all cases, the visual archetype, is the most mature, the last form the document takes in the production process before it goes on to the materialization stage. In the desktop publishing case, the visual archetype would be the vector form of the document; the raster version of the document, if such is rendered, we consider part of the material stages of production in the sense that discretization is strongly influenced by the computational resources available etc. In the typeface printing modality we consider the visual archetype to be the result of the typesetting process, what is codified on the compositing stick. In the handwriting modality, the visual archetype is harder to identify. The reason for the handwriting complication is that when one writes, he gets immediate feedback on how his text is rendered on paper. We could hypothesize that the feedback is taken into account and if what was written diverged from the visual archetype, the visual archetype is adapted to incorporate what was written. A scientific testing of such a hypothesis goes beyond the scope of this paper and might be unimportant, since our hypothesis was introduced to describe a concept and not a phenomenon. Even though there is probably no specific visual archetype for handwriting in the scope of a full document or a page, the concept can make sense on a smaller scale such as a stroke, a letter, or a word. We could maybe state that each grapheme is produced in the context of a specific visual archetype. I suggest that declination from the visual archetype happens because of a lack of skill or external derogation such as ink drops, crumbled paper, etc.. Skill is required for visualizing a feasible archetype, for anticipating how graphemes will be rendered on paper, but also for executing the visual archetype with precision. Taking the above assumptions, one could conclude that the more skilled a writer is, the more consistent during the writing of the document the visual archetype will be. As an abstract concept, we associate the visual archetype with the document creator's intention and define it as **the most precise form a document creator's intentions take**. The visual archetype's definition allows to consider any deviation from it as a kind of imperfection, error, or, noise. Note that

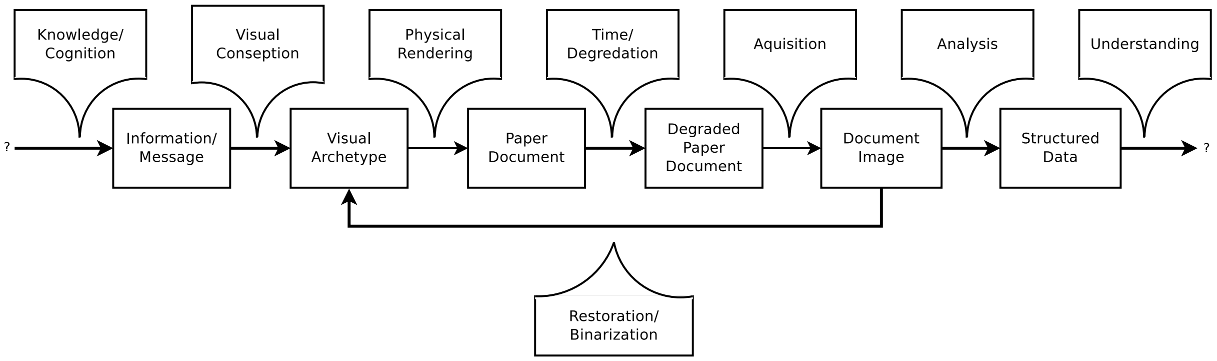


Figure 1.1: The Visual Archetype

visual archetypes should be put into reference with the work on cognitive models and prototype theory, i.e. they can be defined by idealized conceptual models, such as the typeface printing modality and the desktop printing modality, but are not limited to them. Furthermore, the visual archetypes could be a kind of idealized visual model, i.e. the idea of the visual appearance in mind which should be put on paper. This can be put into relation with the idealized conceptual model (ICM) of Lakoff [53], however a detailed analysis of this idea is beyond the scope of this work.

The visual archetype can to a certain extent, be considered when dealing with non-document text such as scene-text but a major limitation arises from the fact the creation of the text might not have control or even not be aware of the context in which the image is placed.

1.2.2 Two Tone Assumption

If we address the question of how colors are modelled in visual archetypes an interesting dichotomy arises. The visual archetypes for some document element types define colors in continuous ranges, while in other document element types as selections from a finite palette which in some cases contains only tones. The authors of this paper postulate that the visual archetypes of text and drawings[1] traditionally have discrete colors and in the overwhelming majority only two tones. In support of our postulation we can only provide some relevant remarks. In[11] Bringhurst, in reference to the engraving the printing process produces on the surface of the paper, states that *although early renaissance typographers were excited by the depth and the sense of touch they could achieve by the printing process, following neoclassical typographers such as Baskerville, would go as a far as to apply a process similar to ironing clothes on the printed document to produce perfectly flat surfaces.*

[1] A picture, image, etc., that is made by making lines on a surface with a pencil, pen, marker, chalk, etc., but usually not with paint. source: http://www.merriam-webster.com/dictionary/drawing , accessed: 2014-02-28

We suggest that text, even in the case that it wasn't, evolved to be bi-tonal because of the used media. Liquid ink on paper, which dominated writing for centuries, makes the color tones practically uncontrollable to the layman. When using a soft pen/plume, pressure variations result in line thickness variations. This way handwriting can transmit non transcribable meanings in the spatial domain. The bold and italic fonts could be described as descendants of the meanings that were associated with spatial information. Contemporary font systems such as TTF contain a purely spatial description of the fonts; tone related modifications, such as anti-aliasing, are automatically inferred during rendering. How could a gray-level document be transcribed without modern technology? What kind of document is untranscriptable?

1.3 Texture Analysis

1.3.1 General Texture Analysis

Defining what is texture analysis in CV is quite subjective, Shapiro and Stockman define it as *An image texture is a set of metrics calculated in image processing designed to quantify the perceived texture of an image. Image texture gives us information about the spatial arrangement of color or intensities in an image or selected region of an image* [85]. The above definition, clear as it may be, delegates to human perseption what texture is. One can intuitevely feel that the essence of texture is associated with repetition of small objects. As best exemplified by CNNs, image descriptions start by encoding at the local structure of pixels and their imediate neighborhood and than combine these to form higher level descriptions of structured elements in an image. Information can be combined either through spatial relationship preserving mechanisms eg. convolutional filters or can be pooled disregarding spatial relationships as eg. BoVW or pooling layers do.

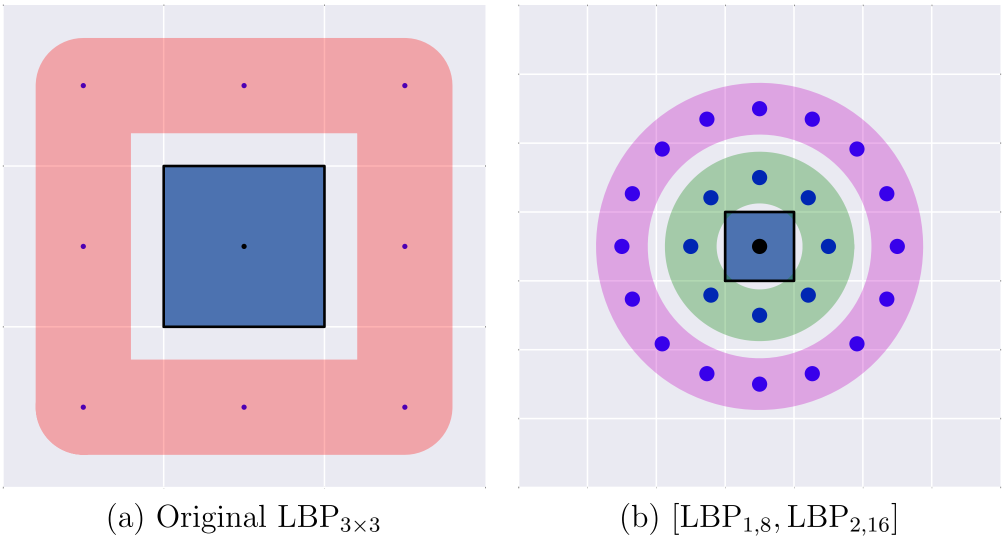
Chapter 2

Local Binary Patterns

The LBP descriptor has a vast set of variants in the literature. In order to be as inclucive as possible, I define single Local Binary Pattern as the *The bitstring constructed by concatenating the binary relationship of a central pixel and a sequence of points sampled around the central pixel.* Initially LBP where defined as the binary encoding of a 3×3 regionaround each pixel [73]. In later work, the 3×3 region was modified to a circle around each pixel[74]. In the inital case and most of the derived variants, the binary relationship d is simply the sign of the diference beetween the center pixel and the pixels sampled from the periphery as defined in 2.1.

$$d(x) = \begin{cases} 1 & : x \geq 0 \\ 0 & : x < 0 \end{cases}, \tag{2.1}$$

Bit-strings can be seen as integers and that is where the power of the LBP descriptor family lies, each pattern can be described as a single integer. So when patterns are pooled over a region the representation is a simple histogram. Both memory and computational complexity of methods using LBP depends mostly on the size of the histogram and therefore on the number of distinct patterns available. The actual set of distinct patterns can also be called a vocabulary of the LBP representation. The LBP are by nature robust against any global illumination variation since the sign of the difference beetween two pixels is preserved in almost all ilumination changes. In [10] an in depth analysis of the prior probabillities of LBP in presented. The analysis provides insights into the effectiveness of LBP but also prooves that the consistency the frequencies of patterns demonstrate are not a property natural texture images demonstrate but rather a property of the mathematical structure of the LBP operation it's self.



(a) Original $\mathrm{LBP}_{3\times3}$

(b) $[\mathrm{LBP}_{1,8}, \mathrm{LBP}_{2,16}]$

Figure 2.1: Popular LBP sampling patterns

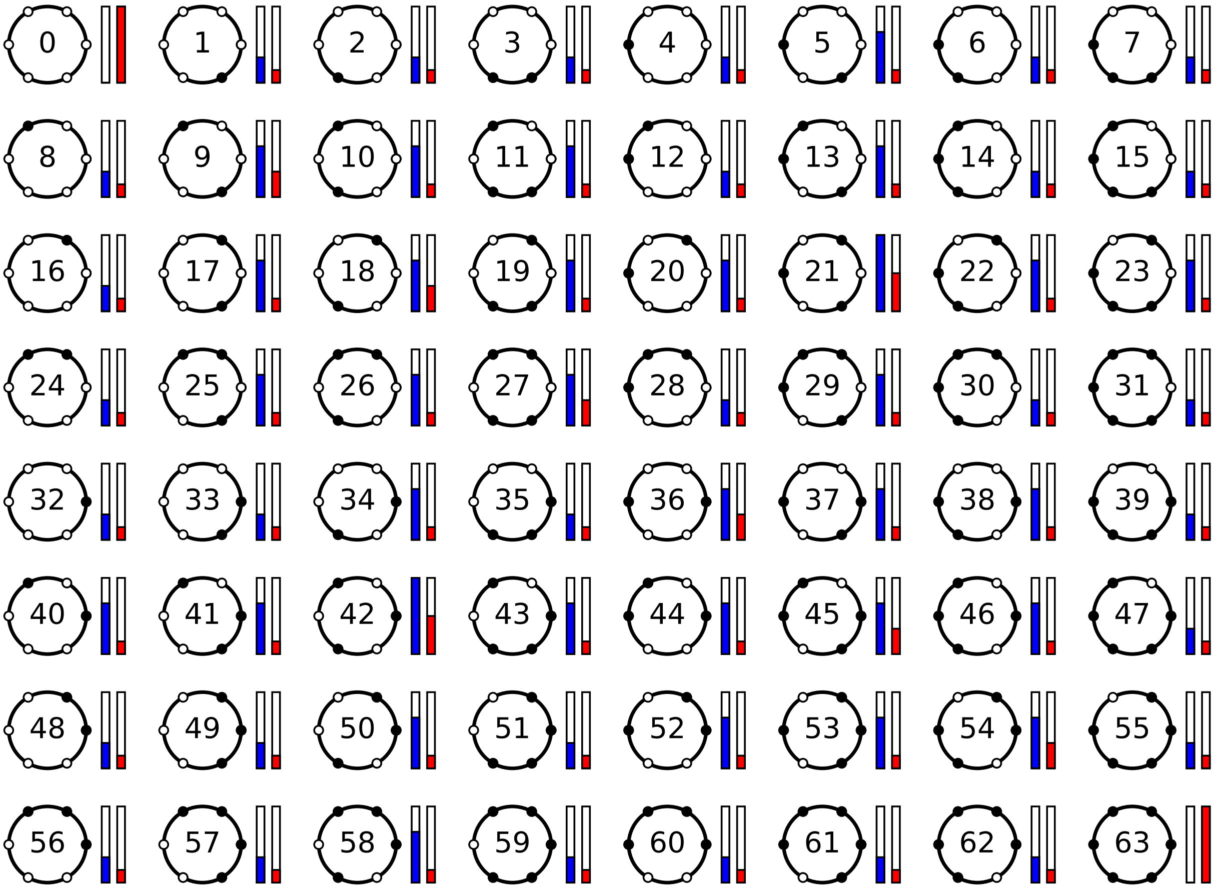


Figure 2.2: All $LBP_{1,6}$ patterns, their uniformity is marked as blue [0-6], and their rotational equivalence counts [0-6]

2.1 Compression Tecniques

If there are n points sampled on the periphery, the vocabury's size is 2^n. In [74] two tecniques were introduced to compress the vocabularies of LBP patterns, rotation invariance and uniformity. Compression can also be achieved by compressing the histograms after they are populated but has no computational benefits of any kind in the actual LBP transform.

2.1.1 Uniform Compression

Uniform compression is defining a set of patterns which will be ignored from the representation and in essence treated as noise. Uniformity is a measurement of smoothness of the local gradient. Specifically the uniformity is defined as number of transitions beetween ones and zeros as the bitstring is traversed in a clockwise maner. A small ambiguity can arise depending on whether only transitions from one to zero are counted or whether both one to zero and zeros to one are measured; for the remaining of this work whenever I refer to uniformity, I count both transitions so uniformity has to always be a even number. It is evident that in an LBP vocabulary of n points, the greatest uniformity is $2\lfloor\frac{n}{2}\rfloor$. In Fig. 2.2 all possible patterns of 6 points can be seen, while typically LBP are sampling multiples of 8 points, 6 points are shown in order to show all patterns. In the specific case of $LBP_{1,6}$ it can be seen that the number of patterns having a uniformity count of zero is always two, the all-zero pattern and the all-ones pattern this holds for any nuber of points. It can also be seen that only two patterns have 6, the highest uniformity possible, this is also true for any LBP lexicon with an even number of points but for an even number of n points the number of the highest uniformity patterns is $2n$. The way uformity is used to compress the vocabulary is to discard all patterns with a uniformity of more than 2; this in essence filters out patterns that lie on sadle points. Filtering the patterns with a Uniformity 2 constrain reduces the vocabualry size from 2^n to $2 + (n - 2)n$. In the case of 6 points, this reduction is from 64 to 26 but in 8 points it is from 256 to 50 which in most cases means saving computational resources by 80%. The implicit assumption behind uniformity 2 filetring is that sadle points at the pixel level are usually occuring because of noise. Uniform patterns were also demonstrated to be more probable regardless of the nature of the input images [10].

2.1.2 Rotation Invariance

Rotation invariance is compressing the pattern vocabularies by mapping patterns that can be converted from one to another by rotation to a single entry in the vocabulary. Other than sampling issues and aliasing when a single LBP is sampled for every point, this reduction makes the histogram representation rotation invariant which is highly desired in cases where nothing can be assumed about the orientation of the image. The number of rotationally equivalent patterns varies for every pattern. Rotation uniqueness of a pattern is the inverse of cardinality of the set of patterns that are equivalent to it by rotation. In Fig. 2.2 the rotational uniqueness of every pattern can be seen as the red bars. It can be seen that the uniqness varies quite a bit, this squees the histogram significantly. While uniformity compression has a simple formula for the vocabulary size, compression achieved on the vocabularies with respect to the number of points, in the case of rotation invariance it is not so simple. The formula given by [103] is given in 2.2 as $N(n, a)$ where n is the number of points and a is the number of distict colors a "bead" can have in the LBP case always 2, $v(n)$ is the number of divisors n has and ϕ is eulers totien function.

$$N(n,a) = \frac{1}{n} \sum_{i=1}^{v(n)} \phi(d_i) a^{n/d_i} \tag{2.2}$$

A major drawback of rotation invariance bining is that it discards information about the relative frequencies each rotationally equivalent pattern has to each-other. This can be easilly understood if one thinks of a pattern of rectangles, rotation invariance bins patterns on horizontaland vertical edges together making all rectangles that have the same circumference produce exactly the same histogram regarless of their width and heigh. In 2009 a histogram-level rotation invariance was proposed [2] while this doesn't economise computational resources as the orinal rotation invariance compression does, it provides roation invariance while preserving the information of the relative frequencies between rotationally equivalent patterns.

2.1.3 Multiple Radii

Multiple radii can encapsulate structure occuring at different scales

2.1.4 LBP Cardinalities

In their typical use LBP occurences are encoded as histograms contining their freequencies.

2.1.5 Evolution and applications of the LBP descriptor

In [106] the LBP was extened from images to volumes where the new dimension is time and thus they encode change of a pattern at a specific image location over time. LBP characterize local image patches using binary codes that encode the relationship between a central pixel and its neighbors [74]. LBP feature extraction usually consists of computing LBP descriptors at each pixel of an image to create an image of integer valued codes,

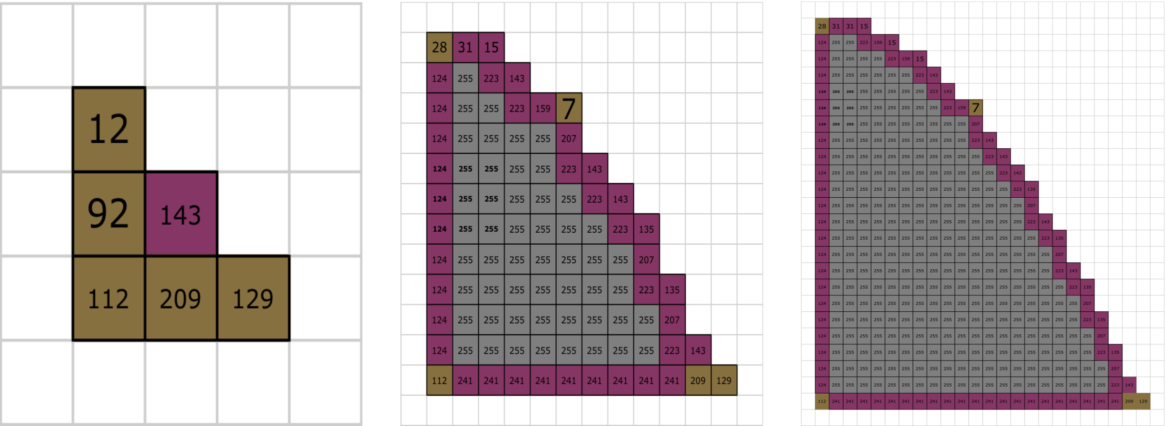


Figure 2.3: LBP patterns computed on a surface defined by a quadratic curve sampled at varius resolutions

followed by pooling of these codes into a histogram [74]. LBP have been successfully applied to many of the major computer vision problems such as face recognition [1], facial expression recognition [84], and human detection [69].

LBPs have also been applied to document image analysis in tasks other than writer identification. In DIA, LBP have been used for text detection of text in television streams [5], for printed script detection [29], and in [102] LBP were compared to other features in a feature selection process for historical document layout analysis and were selected as best across all datasets. The authors used an earlier proponent of the method presented in this paper for Arabic font recognition [71]. and were shown to dominate a feature selection process for use in historic page layout analysis [102]

2.1.6 Cardinalities

One of the reasons the LBP is so successfull is that a single pattern is represented as a bit-string of fixed width. In most cases a single pattern can be represented as an integer and therefore when pooling the memory complexity for storing the pooled patterns is $\mathcal{O}(\log n \times m)$. Where n represents the number of patterns and m represents the number of points the LBP descriptor.

Chapter 3

Word Spotting

3.1 Introduction

In recent years, many important and valuable documents have become accessible as digital images. A large amount of documents which were previously on paper, are also being digitized as images. These documents are quite precious and important for humanity, which need to be preserved permanently. These images contain printed or more often handwritten text. Most of these documents have been degraded to a great extent due to reading and ageing processes. These documents should be provided for users to access and retrieve including searching keywords throughout the documents. Word spotting has become a central tool in large scale document interpretation. Several document processing scenarios use word spotting as core technology for classification, annotation or to make them searchable. Word spotting has been proposed as an alternative to OCR, as a form of content-based retrieval procedure, which results in a ranked list of word images that are similar to the query word. The query can be either an example image Query by Example (QBE)[82] or a string containing the word to be searched Query by String (QBS)[3]. The basic idea of word spotting using query by example is that a template image is selected from a set of predefined keywords, i.e. words of interest and then search is initiated to find out its other instances in the target set of the digitized documents. This factoid makes the approach more flexible and suitable for indexing and retrieval of degraded and historical documents written in multiple languages. The principle of word spotting using query by string is that queries are typed. Textual and image features are jointly embedded and correlated into a n-dimensional space. In this space, textual and image features can be projected from one to the other.

In a typical word spotting pipeline, pre-processed patches/imperfectly segmented or cropped words are first obtained. This segmentation step is not always straightforward and might be prone to many errors. In fact, although word and text line segmentation is a highly cultured research topic, it is far from being a solved problem.

In the literature, word spotting evolves under two distinct sections: the segmentation-free approach and the segmentation-based approach. Any method classifying or retrieving words which are segmented beforehand from a given full page image are grouped under the category of segmentation-based method.

Most of the segmentation free word spotting methods use a separate segmentation technique ad-joint to the word spotting method. Segmentation based becomes segmentation-free given a word segmenter[4]. Depending on the amount of information localized, the performance of the word spotting method differs a lot. In a realistic scenario getting a perfectly segmented word from the database is a very rare phenomenon. On the other hand, the performance of the good state-of-art (SotA) methods degrades significantly if words are improperly segmented. In this chapter we will be confining ourselves to segmentation based approaches for better understanding the importance of a word segmenter or localiser.

The main motivation of this work is to provide an exhaustive analysis of different SotA methods in a practical scenario. This allows one to define a taxonomy on the convenience of each methods to the possibility of proper segmentation in target documents. Thus, a "difficult" document would require a spotting method that is robust. This analysis can bridge the gap between both the segmentation based and segmentation free word spotting methods. Segmentation errors have a cumulative effect on subsequent word representations and matching steps. This dependence on good word segmentation motivated the researchers from the keyword-spotting domain to recently move towards complete segmentation-free methods[82]. But most of these end to end word spotting methods comprise a segmenter whose quality of segmentation is not always perfect. In this chapter we analyze the robustness of different state of the art method to improper segmentation. We provide mean average precision measures of different levels of cropping. Additionally we provide some other measures like cross dataset performance by different methods too. We also present the orthogonality/independence of different methods with an intuition for the potential of optimal fusion. The final goal of the provided evaluation is to have a recommendation survey not only on the robustness of the methods in terms of quality segmentation, but on their complementary in fusion techniques. The principal focus of this chapter is not to introduce a new method neither for segmentation nor for word spotting but rather to provide insights on the evaluation and experimental procedure for word segmentation and word spotting methods. The key contributions of this work are:

- Demonstrating the limitations of using perfectly segmented input to evaluate word-spotting performance.

- Implementing an experimental pipeline that allows to compare learning-free, supervised learning, and unsupervised learning methods in a homogeneous manner.

- Performing experiments with many state-of-the art methods under the exact same conditions and pipeline.

- Demonstrating that the best word-spotting method depends on the quality of the segmentation.

- Demonstrating that Intersection over Union (IoU) is a valid quality metric for word segmentation methods.

3.2 State of the art

Word spotting can be broadly classified under two distinct sections: the segmentation-free approach and the segmentation-based approach. In the latter approach, there is a tremendous effort towards solving the word segmentation problem[76] [7][52] [9]. One of the main challenges of keyword spotting methods, either learning-free or learning-based, is that they usually need to segment the document images into words [45] [80][76] [55] or text lines [34] [95] using a layout analysis step. In critical scenarios dealing with handwritten text and highly degraded documents [56][58] segmentation is highly challenging. The work of Rusinyol et al.[81] avoids segmentation by representing regions with a fixed-length descriptor based on the well-known bag of visual words (BoVW) framework [21]. In this case, comparison of regions is much faster with the use of a dot product or Euclidean distance. Recent pieces of works on word spotting have proposed methods where a precise word segmentation is not required, or, in some cases, no segmentation at all. The recent works of Rodriguez et.al. [78] propose methods that relax the segmentation problem by requiring only segmentation at the text line level. In [37], Gatos and Pratikakis perform a fast and very coarse segmentation of the page to detect salient text regions. Rothacker et.al. [79] propose to generate hypotheses with text detectors based on SIFT contrast scores, CNNregion classification scores and attribute activation maps. In [40], Ghosh inspired by the success of bounding box proposal algorithms in object recognition, proposed a scheme to generate a set of word-independent text box proposal. They also propose to use [39] the whole input image and a set of word candidate bounding boxes and embeds all bounding boxes into an embedding space, where word spotting can be casted as a simple nearest neighbour search between the query representation and each of the candidate bounding boxes.

The most common approach relies in using a patch-based framework in which a window slides over the whole document. In such a framework expected segmentations may not be perfect and elements from surrounding words will appear within a patch. Automatic word segmentation, as presented in [92], is based on taking several features on either side of a potential segmentation point and then using a neural network for deciding whether or not the segmentation is between two distinct words. The segmentation free method attempts to perform spotting and segmentation concurrently. An entire line image acts as input in place of a candidate word image. The line is split into segments based on an algorithm similar to the ligature-based segmentation algorithm used in [50]. The text detection algorithms can be broadly classified into two categories: connected component based approach and sliding window-based approaches[54][101]. The sliding window based methods, approach to localize individual characters[101] or whole words [54] drawing inspiration from other object detection methods where this approach has been successfully applied [22] [100]. Strengths of such methods include robustness to noise and blur, because they exploit features aggregated over the whole region of interest. The main drawback is that the number of rectangles that it needs to be assessed grows rapidly when text with different scale, aspect, rotation and other variations are taken into consideration. Due to the variance of sliding window parameters is much lower, this kind of effect does not occur in general object detection tasks.

In the same spirit with the aforementioned approaches, this chapter concerns a study on the performance of word spotting methods, for recreating the accuracy achieved with different pipeline in the case of segmented words. The evaluation conducted in this work compares several state of the art methods of word spotting belonging to the segmentation based category. We will analyze their sensitivity to different levels of proper segmentation, i.e. how robust are they to incorrect segmentation in previous steps.

3.2.1 Almazan et al. [4] (FisherCCA)

The use of exemplar SVM's has created one of the best segmentation free methods in the literature in terms of accuracy and mean average precision. This supervised method represents the documents with a grid of Histogram of Gradients (HOG) descriptors. An exemplar SVM framework is used to produce a better representation of the query. They also use a more discriminative representation based on Fisher Vector to re-rank the best regions retrieved, which in turn is used to expand the Exemplar SVM training set and improve the query representation. Ultimately the document descriptor is pre-calculated and compressed with the Product Quantization.

3.2.2 Rusiñol et al. [82] (BoVW)

In the paradigm of segmentation free, query by example handwritten word spotting, this method has outperformed the recent state of the art keyword spotting approaches. In the experiments of this chapter, for the sake of comparison in the proposed framework, we have used a segmentation based variant. This method uses a patch based framework where local patches which are specified by bag of visual words (BoVW) model powered by SIFT descriptors. These descriptors are then projected to topic space with the Latent Semantic Analysis technique and then compressing of these descriptors is done with the Product Quantization method. This statistical approach in turn enables an efficient indexation of document information both in terms of memory and time.

3.2.3 Rath et al. [77] (DTW)

It can be considered as the baseline algorithm for matching handwritten words in noisy historical documents. The segmented word images are pre-processed to create sets of 1-dimensional features, which are then compared using Dynamic Time Warping (DTW).

3.2.4 Method based on Quad Tree (Quad Tree)

This method relies on an adaptive feature extraction technique [99] based on recursive subdivisions of the word images so that the resulting sub images at each iteration have balanced (approximately equal) numbers of foreground pixels, for two levels. This adaptive hierarchical decomposition technique which determines the pyramidal grid which is recursively updated through the calculation of image geometric centroids [91].

3.2.5 Dey et al. [25](LBP)

In this method the adaptive hierarchical decomposition technique employs the pooling of the Local Binary Patterns in the adaptive regions, that are determined by a pyramidal grid that is recursively updated through the calculation of image geometric centroids to calculate the feature vector for matching.

3.2.6 Method based on Histogram of Gradients (HOG)

Similar to the previous method, the feature vector is created by pooling the gradients in the similar pyramidal grid. Therefore, the feature vector describes the frequency of gradients and considering that the images contain hand written text, the HOG descriptor can roughly be associated to the characteristics of stroke fragments (curvature, smoothness, etc.).

3.2.7 Sudholt et al.[94] (PHOCNET)

In this method a deep CNN architecture is designed for word spotting. It is able to process input images of arbitrary size and predict the corresponding Pyramidal Histogram of Characters(PHOC). The authors show empirically that their architecture is able to outperform state of the art results for various word spotting benchmarks. They also showed that simple data augmentation and common regularization techniques can be used on small datasets to train CNN's from scratch.

3.2.8 Fusion

In this work, in addition to the robustness of each method to improperly segmented words, we also study how complementary are the methods. To assess how two methods are complementary and how the combination of them can overcome the lack of individual performance, a naive late fusion between methods is also proposed. Due to the fact that many of the methods analyzed in this chapter are learning-free, a learning-free late fusion was preferred. More specifically, the fundamental assumption is that any method that does retrieval, can provide a vector with all distances between the query sample and all samples in the retrieval database. The fusion method we propose is a weighted sum of such vectors that two or more methods provide. In the taxonomy of fusion methods described in [6], it would be described as a weighted linear fusion at the level of decision. One of the main benefits of this method is that it can be applied to feature representations of a variable size or even methods that don't have a feature representation such as DTW.

3.3 Experimental Analysis

The experiments performed had as a principal goal to obtain an in-depth analysis of the reliance each method has to high quality segmentation. The principal experiment consists of comparing retrieval of all methods under different levels of distortion on the retrieval database. A major constraint of designing the experimental procedure was making a fair and informative comparison between supervised learning, unsupervised learning, and learning free methods.

3.3.1 Datasets

In this work we have mainly focused in historical manuscripts due to the difficulties they convey. Nowadays it is a typical and well established scenario for the evaluation of word spotting. Two well established publicly available datasets were used: The George Washington (GW) dataset [31], a single-writer dataset, and the ground-truthed part of the Barcelona Historical Handwritten Marriages Database (BCN)[28] which is a multi-writer dataset. Both datasets were partitioned at the page-level having the first 75% of the pages as train-set and the last 25% of pages were designated as a test-set. The words occurring a single time are stemmed when calculating retrieval metrics. In Table 3.1 the specific word counts for the employed datasets can be seen. By default, when the dataset is not specified for a measurement, it is performed on GW. In every dataset any short word with a transcription of less than three characters was discarded. In the case of retrieval of test-set words from the train-set, all words not represented in both were discarded. The distribution of the word lengths also showed that words with less than three character even though had a lot more effect over the metrics during retrieval, it had a very less chance of getting queried in real scenarios. Trimming the dataset based on the distribution of the word lengths made the dataset less skewed.

Table 3.1: Employed Datasets

Dataset	Partition	Page#	Word#	Unique	Stemmed
				Word#	word#
GW[31]	Train	15	3696	967	265
	Test	5	1164	431	563
BCN[28]	Train	30	9879	1387	779
	Test	10	3051	607	367

3.3.2 Performance Evaluation and Metrics

Table 3.2: Analysed method performance

Method	Learning	mAP(GW)	mAP(BCN)	Cross Dataset	Retrieval sec.	Train time
Quad-Tree	Standardization	15.5	30.14	15.32	44.41	**0**
BoVW [81]	Unsupervised	68.26	-	-	-	1 hr.
FisherCCA$_{\mathrm{QBE}}$ [4]	Supervised	93.11	95.40	**72.42**	137.63	2 hrs.
FisherCCA$_{\mathrm{QBS}}$ [4]	Supervised	**96.29**	95.71	15.17	75.33	2 hrs.
PHOCNET$_{\mathrm{QBE}}$ [94]	Deep Learning	95.56	**97.01**	60.93	140.45	3 hrs.
PHOCNET$_{\mathrm{QBS}}$ [94]	Deep Learning	95.04	93.68	-	-	3 hrs.
DTW [77]	No	20.94	-	-	78095.89	**0**
HOG pooled Quad-Tree	No	48.22	66.66	66.66	45.34	**0**
LBP [25]	No	54.44	70.84	70.84	**43.17**	**0**

3.3.2.1 Performance Evaluation

The experimental pipeline was designed to perform consistent measurement of different methods under a variety of conditions. The used pipeline is equally well suited for comparing learning-free, unsupervised, or supervised learning methods. Each method is modeled as a feature extraction process which takes a word image as input and provides a feature vector as output. Any transformation of the feature space such as metric learning, normalization, standardization etc. is applied on the feature vectors. These vectors are used to generate a distance matrix using the cosine distance. A train-set is made available to all methods that need to learn the representation they produce. Each row in this matrix represents a sample from the test-set and each column represents a sample from the train-set. From the distance matrices a precision matrix is computed where each row refers to a query sample (test-set) and the n^{th} column refers to the precision of the retrieval for the n first samples. The precision matrices are computed considering only the exact same

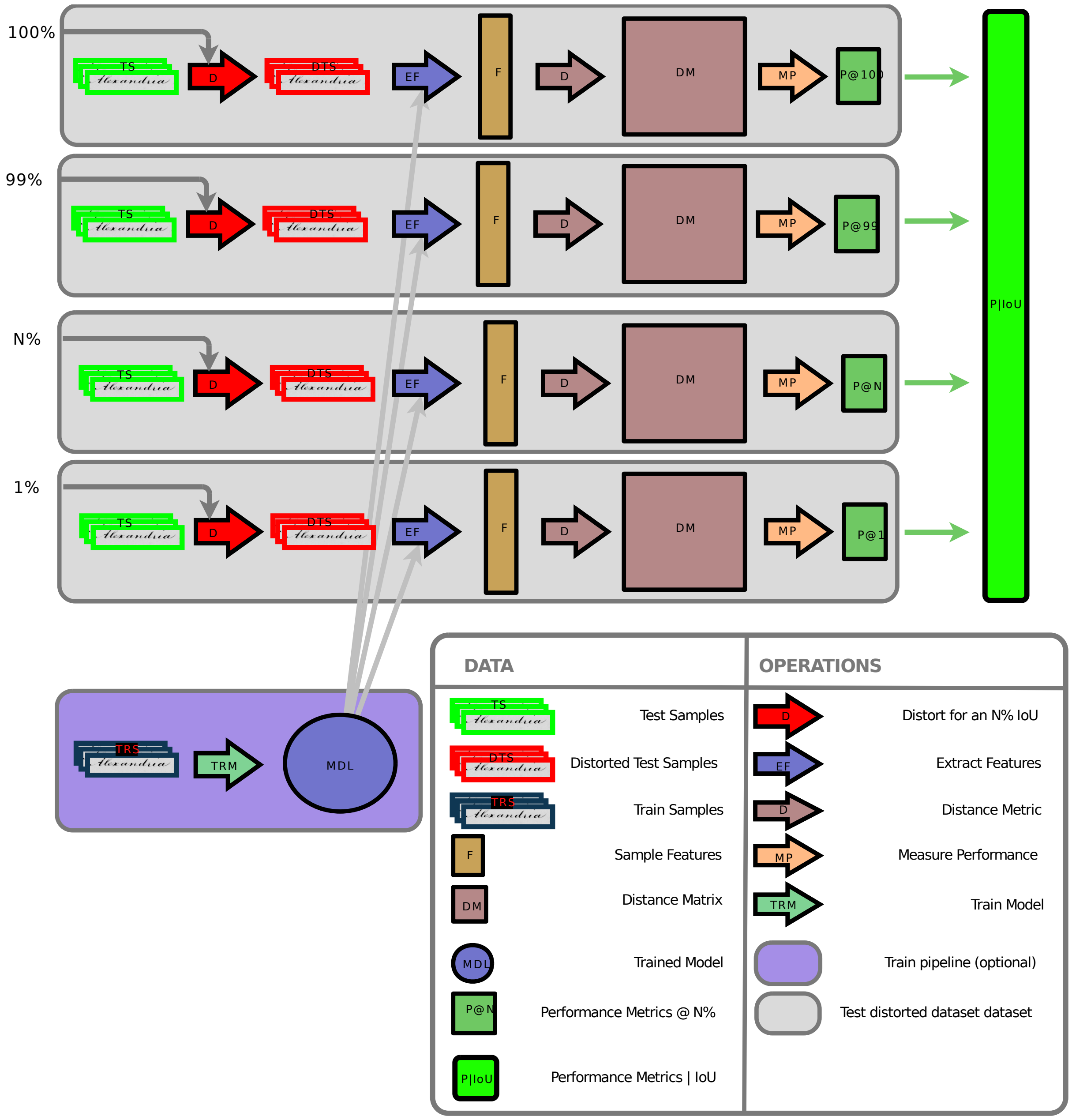


Figure 3.1: Experimental procedure for estimating $P(Correct|IoU)$ for a single word-spotting method. (Best viewed in electronic format.)

case-insensitive transcription between query and retrieved sample as a correct match. All word-spotting/retrieval performance metrics employed in this paper are computed based on these distance matrices.

Precision at any given index is defined as the percentage of correct retrievals for all samples at lesser or equal index. mAP and rPrecision metrics demonstrated remarkable correlation between them. Although rPrecision appears less saturated, mAP was preferred because it is more broadly used.

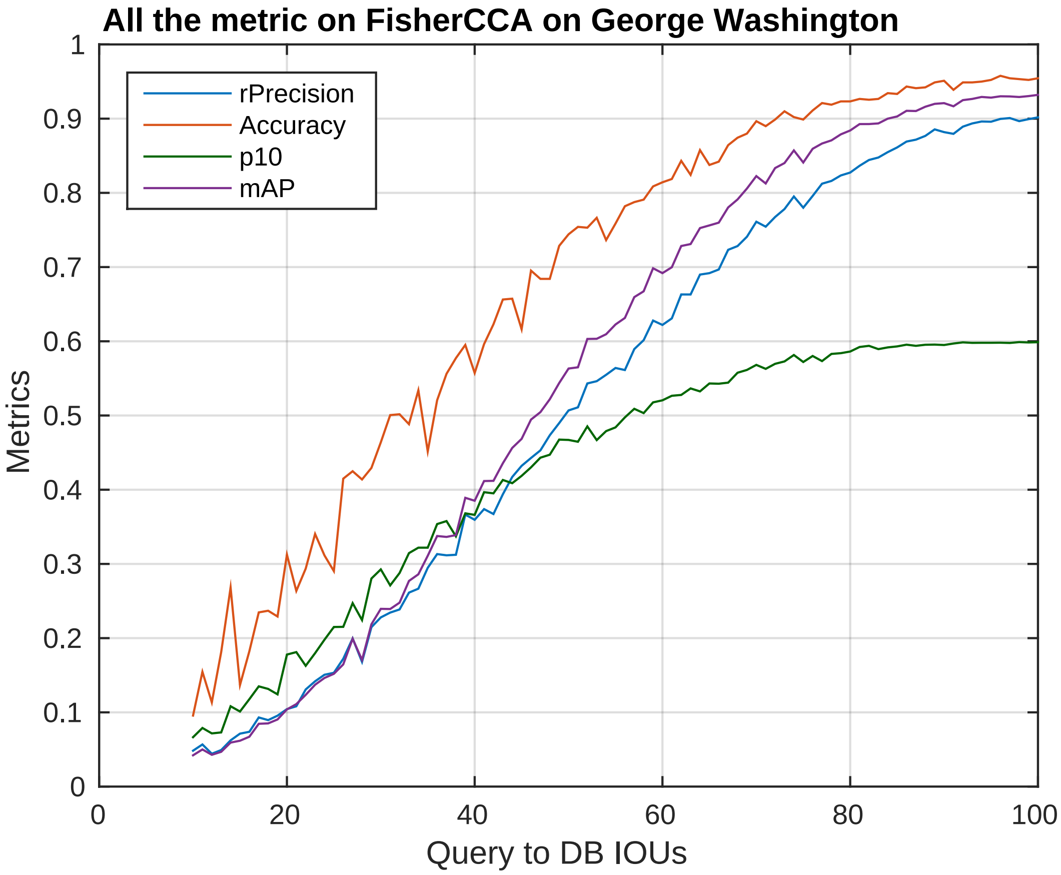


Figure 3.2: All metrics of Fisher CCA [4] on distorted segmentations. Best viewed in pdf.

3.3.2.2 Performance Metrics

All performance metrics are estimated on each query-sample and then averaged for all queries. The metrics that are used for the analysis are:

- Accuracy: The percentage of queries who's nearest retrieval contains the same transcription, it can be described as precision at index 1.
- rPrecision: The precision each query gets at the retrieval position where a perfect recall and precision scores are possible. Given a query, we label the set of relevant objects with regard to the query as *rel* and the set of retrieved elements from the database as *ret*. rPrecision is the precision at rank *rel*
- Precision @ 10: The precision each query gets for the 10 most relevant samples. The precision is defined in terms of *ret* and *rel* in Eq. 3.1.

$$Precision(P) = \frac{\mid ret \cap rel \mid}{\mid ret \mid} \tag{3.1}$$

- mAP: The average of precision each query gets at each correct retrieval as shown in Eq. 3.2, where $P@n$ is the precision till the n-th place of retrieval for a given query, and $r(n)$ is a binary function on the relevance of the n-th item returned in the ranked list.

$$mAP = \frac{\sum_{n=1}^{|ret|}(P@n \times r(n))}{\mid rel \mid} \tag{3.2}$$

- Self-classification Accuracy: It is obtained by allowing a query sample to retrieve itself as well. This metric makes sense when the test-set is a distorted version of the train-set. When there is no distortion, all samples by definition obtain a self-classification rate of 100%; it is therefore well suited as a metric that ignores performance allowing a comparison of methods with different performances such as learning-free and supervised learning. Other variants of this metric based on mAP or rPrecision can also be employed, but were not investigated as reporting on them would not be as informative.

In Fig. 3.2 the quality of segmentation plot can be seen using all the above metrics. The Query to DB IoUs is the retrieval of the actual query image from the dataset of different Intersection over Unions with respect to its groundtruth box. All the words in the database are distorted in accordance to the distortion model proposed in this paper to quantify the results of different segmenter. What stands out from that plot is that accuracy is quite unstable given that the x-axis is a quantification of the hardness of the dataset. The instability of accuracy also appears in self-classification accuracy. What also stands out is how non-informative precision at 10 is, this can be attributed to the fact that many samples have less than 10 examples per class.

3.3.3 State-of-the-art comparison

The methods analyzed are the state-of-the-art in segmentation based word spotting, or interesting baseline methods. Hence most types of word spotting methods are represented. Performance was measured on both the GW and the BCN datasets. In order to allow for a comparison between learning-free, unsupervised, and supervised learning methods, a cross-dataset experiment consists in also presented in Table 3.2. The cross-dataset experiment is retrieving the BCN test-set from the BCN train-set, but any training of the methods was performed on the GW trainset. The cross-dataset measurement is an indicator of how a method performs on unseen data.

In Table 3.3 the principal characteristics of all methods can be seen. PHOCNET performs better both in GW and BCN datasets and FisherCCA is a close second. It can be observed that supervised methods perform better by a large margin and BoVW which is an unsupervised learning method performs far better than any simpler and learning free methods. Although PHOCNET is the uncontested state-of-the-art, its cross-dataset performance is lower than learning free methods. FisherCCA on the other hand demonstrates tolerance to unseen data.

Concerning time, the training of the models was a highly heterogeneous process and a valid measurement was not possible. Retrieval of a single version of the GW test-set from the GW train-set was measured on a Intel Xeon E5-1620 system with an NVIDIA GeForce 1070 GPU. The column titled *Retrieval* contains the time cost of retrieving each sample in the test-set of GW from a database with all samples in the test-set of GW as queries. It is apparent that DTW is remarkably slow, while texture based learning free methods are the fastest. It should be pointed out that the PHOCNET employ GPU computing while all other methods were computed on a single CPU thread.

3.3.4 Quality of Segmentation (QOS)

In the literature word spotting methods, unless they are mentioned as segmentation free, report experiments where performance is measured on perfectly segmented words obtained from the groundtruth[94]. Word segmentation depends on totally different factors like irregularity in writing styles, skew, quality of degradation and etc. Historical documents can be challenging to evaluate the influence of these factors. In order to estimate experimentally the effect of improper segmentation might have on the state-of-the-art word spotting methods an experiment using synthetically generated improper word segmentation was performed. The following assumptions were made when designing the experiment.

- Intersection over Union (IoU) of two rectangles containing the same word is a valid quantification of the quality of segmentation as shown in Fig. 3.3.
- Modeling segmentation errors as over and under-segmentation's along several directions is realistic enough to allow to draw conclusions.
- Distorting all samples in a dataset by a fixed quantity should produce datasets that are equivalent to real-world poorly segmented datsets.

The quality of word segmentation is quantified as the Intersection over Union (IoU) of the two-point bounding boxes of the proposed word and the word in the groundtruth. The definition of IoU for bounding boxes is given in Eq 3.3.

$$\begin{aligned} w_I &= min(R_1, R_2) - max(L_1, L_2) \\ h_I &= min(B_1, B_2) - max(T_1, T_2) \\ IoU &= \frac{w_I \times h_I}{w_1 \times h_1 + w_2 \times h_2 - w_I \times h_I} \end{aligned} \tag{3.3}$$

Where $R_1(Right), L_1(Left), B_1(Bottom), T_1(Top)$ are the sides of the bounding box for the undistorted object, w_1, h_1 is its width and height. The dotted boundary is the undistorted word where as dark line is the distorted word which is an estimation of the segmented word that can be expected from a method.

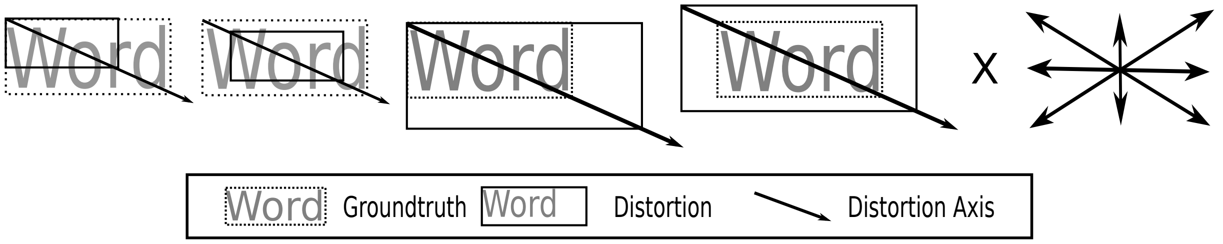
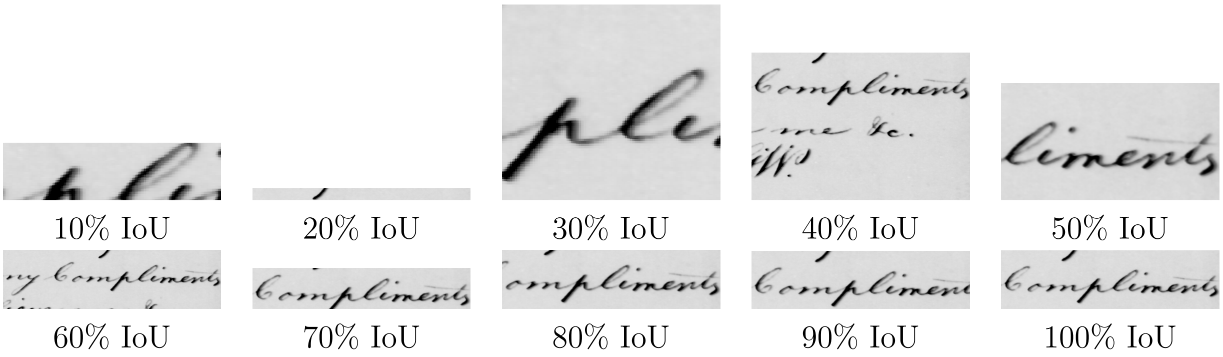


Figure 3.3: Visual depiction of degradation generated for a specific IoU on the top-left corner.


10% IoU 20% IoU 30% IoU 40% IoU 50% IoU

60% IoU 70% IoU 80% IoU 90% IoU 100% IoU

Figure 3.4: Different IoU of a specific sample.

3.3.4.1 Distortion Model

In Fig. 3.3 a visual representation of the distortion generator can be seen. The generator takes the groundtruth rectangles, the document page, and the desired IoU and provides improperly cropped word images. Given a direction and a specific IoU, four degradations were modeled, centered and uncentered, over and under segmentation. The degradation can happen on eight possible directions, which models errors on text-line segmentation, errors on word segmentation given correct text-lines, or errors on both. Given that centered distortions of opposing directions result in same rectangles, for a given IoU a rectangle can be distorted in 28 different ways. Since the precision is measured using the cosine distance which is symmetric, performance on retrieving distorted samples from an undistorted database, is identical to the performance of retrieving undistorted samples from a distorted database. In order to avoid training models for every degraded version of the datasets, degradation was only applied on the retrieval samples and the query databases were left undistorted. In order to avoid training models for every degraded version of the datasets, degradation was only applied on the query samples (test-sets) and the retrieval databases

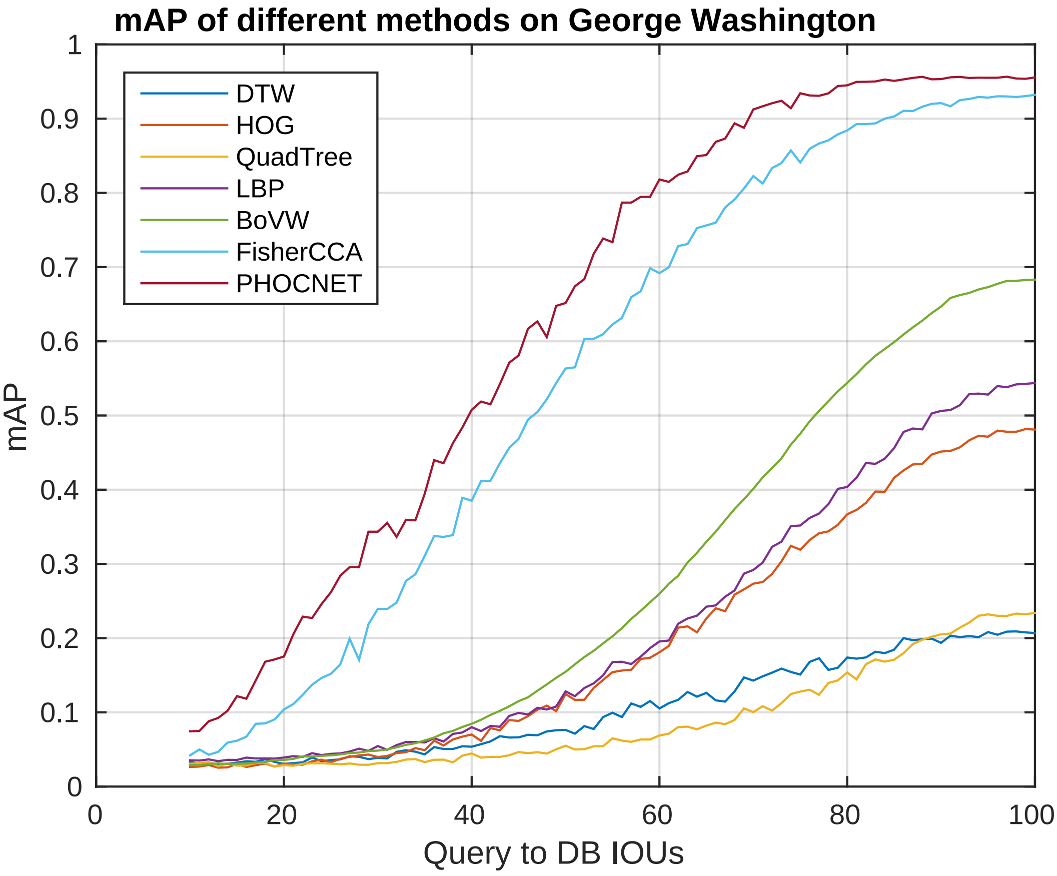


Figure 3.5: Effect of improper segmentation on mAP for GW.

(train-sets) were left undistorted. For every sample in the test-sets 100 different versions were cropped having IoU between 1% and 100% with groundtruth, selected from the 2800 variants possible. In Fig. 3.4 the same sample can be observed at different distortions, 100% IoU means no distortion. As the bounding boxes in the groundtruth are not tight, it can happen that samples with a non-zero IoU share no foreground pixels with the groundtruth rectangle. This is well demonstrated on the 10% sample of Fig. 3.4.

3.3.4.2 Quality of Segmentation Experimental Results

In Fig. 3.5 and Fig. 3.6 the effect of segmentation on the mAP of each method on GW and BCN can be seen. As for the undistorted measurements presented in Table 3.2 supervised methods demonstrate superior performance to other methods and unsupervised method (BoVW) follows and the texture learning-free methods follow. As distortions grow, supervised methods diverge from the rest, increasing the gap in performance with other methods. In the case of BCN, FischerCCA outperforms the PHOCNET for segmentation with an IoU less than 70%. In Fig. 3.7 self-classification accuracy on the GW dataset can be shown. Self-classification as a measurement quantifies robustness against distortion

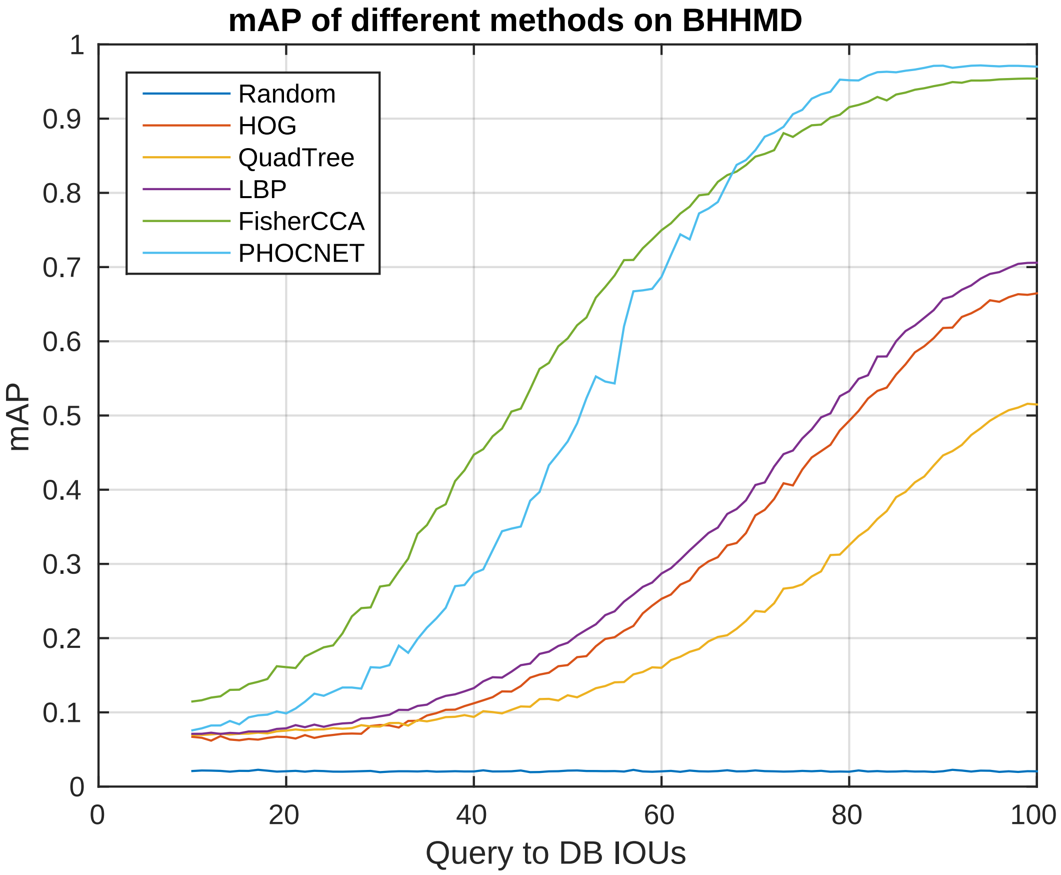


Figure 3.6: Effect of improper segmentation on mAP for BHHWD.

regardless of the performance each method demonstrates. What stands out is that BoVW seems to be more sensitive than other methods which it outperforms on small distortions. A qualitative example of a randomly selected query can be seen by all methods in Fig. 3.8. The numbers seen in the green boxes are the unique sample ids.

3.3.4.3 Query By String (QBS)

FicherCCA and PHOCNET methods both operate by embedding the visual features of a sample to a subspace common with a string embedding called the PHOC. This allows one to use the exact same models in either modality, QBE or QBS. A comparison between the accuracy achieved by both modalities for both FischerCCA and PHOCNET can be seen in Fig. 3.9. It is worth noticing that for FischerCCA QBS performs consistently better than QBE while for PHOCNET, QBS performs quite worst than QBE. Overall, on good segmentation, the best accuracy is achieved by the QBS modality of FischerCCA.

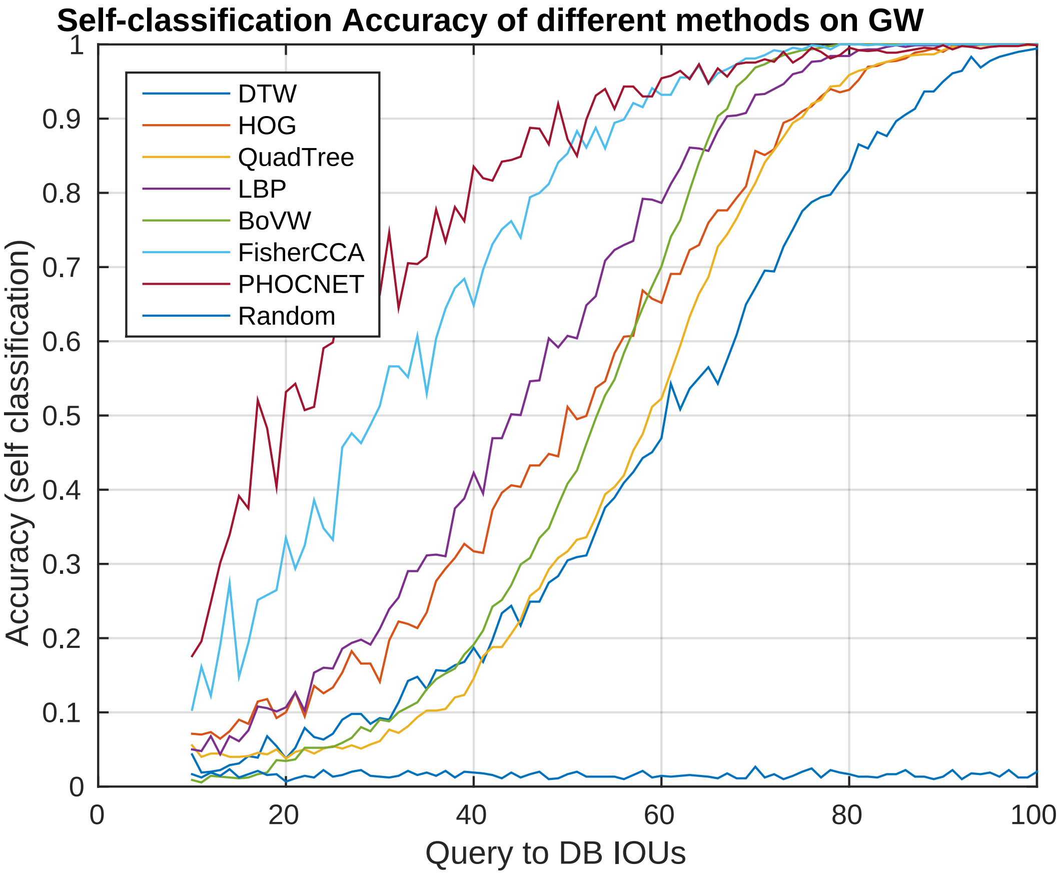


Figure 3.7: Effect of improper segmentation on self-retrieval for GW.

3.3.5 Available Segmentation

The use of a word segmentation method, allows for an estimate on the end-to-end performance a pipeline using segmented word spotting can demonstrate. An end-to-end system consists of text spotter, which jointly with extracted features detects and recognizes words in historical images. In order to put the QOS plots into perspective, an assessment of how an established method for historical document word-segmentation method performs on the datasets is also presented. A recent overview on state-of-the-art of word segmentation, can be inferred from the 2013 ICDAR competition [93], although the comparison is done on well binarized contemporary documents which are very different in nature from historical documents.

In the literature there are many segmentation algorithm such as [35] which have very close performance to the one we chose.

An established binarization-free method for automatic segmentation was used [66] and executed on the GW dataset (without tuning it for our dataset); a broader range of segmentation methods and a comparative analysis, would go beyond the scope of this paper. In Manmatha et. al. [66] the method was tuned with the GW dataset in mind. In order to

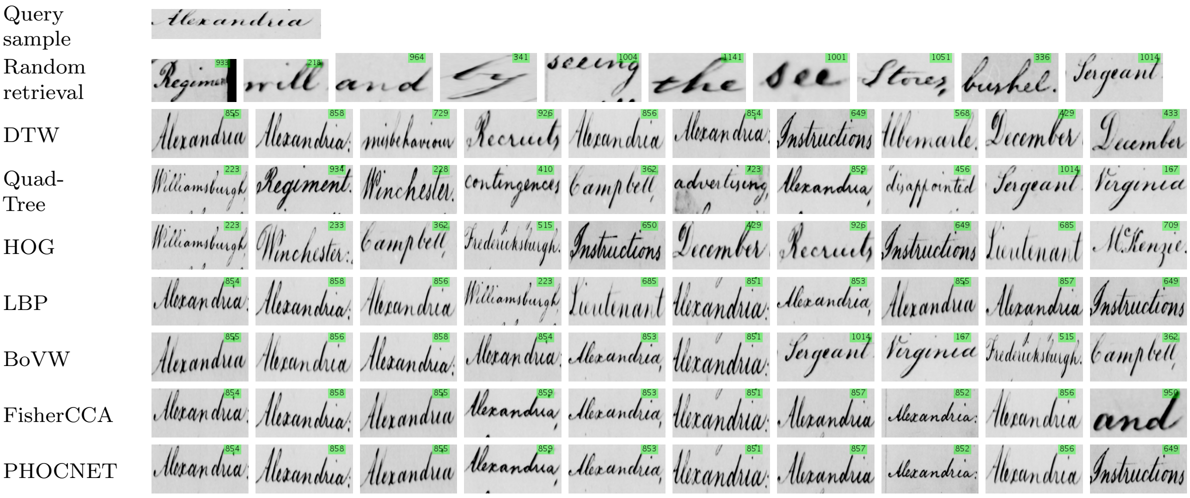


Figure 3.8: Qualitative example of retrieval results of the methods for a randomly selected query (Alexandria). From left to right: the nearest samples retrieved by each method. The id of each sample which distinguishes different samples with the same text is marked in superscript (green). Best viewed in pdf.

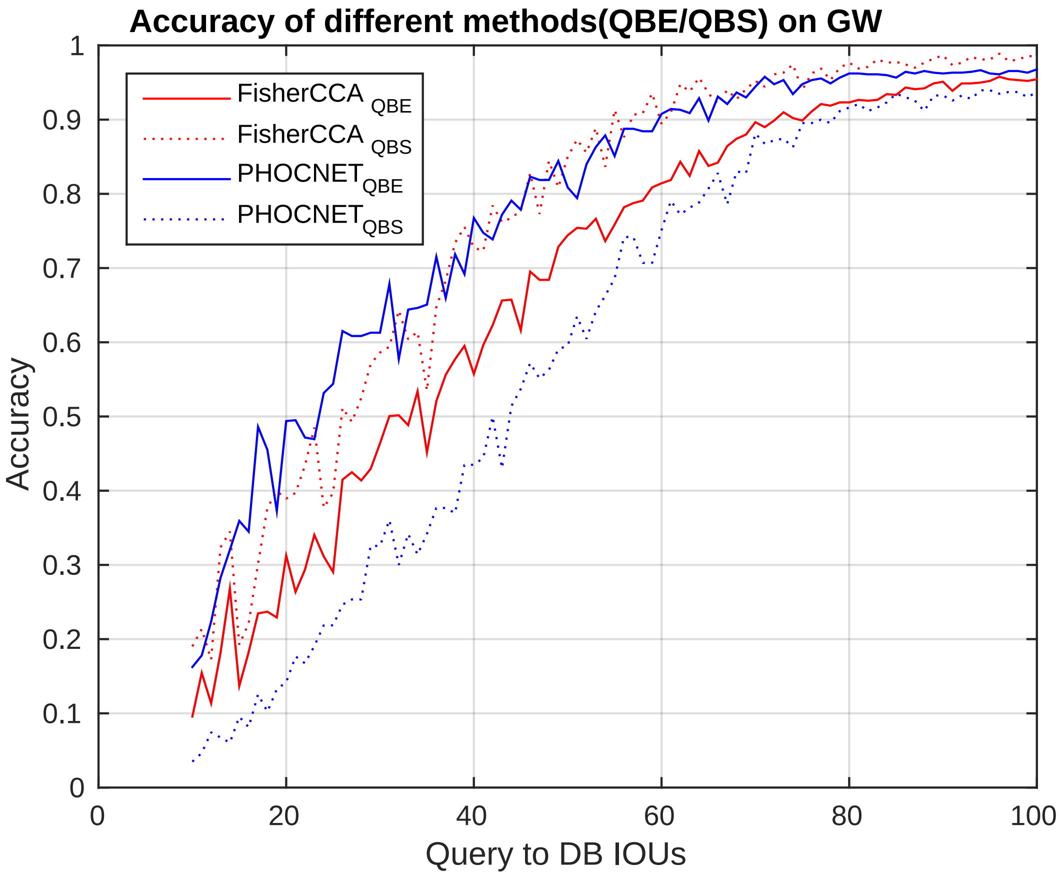


Figure 3.9: Accuracy of QBE for GW.

measure the IoU of the proposed words and the actual words, a sparse matrix is created where each row refers to a groundtruth word-box and each column refers to a proposed

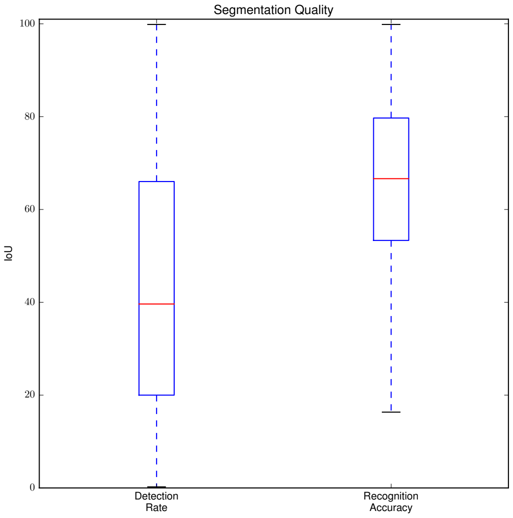


Figure 3.10: Distribution of the IoU obtained by application of [66] on GW dataset

word-box. Several measurements can be obtained from this matrix, the most informative with respect to the QOS curves are the column-wise maximum and the row-wise maximum which are related to the Detection Rate and Recognition Accuracy as defined in [37] the only difference being that there is no threshold at 90% and the actual IoU is used as a soft measurement. In Fig. 3.10 the statistics of IoU can be seen.

3.3.6 End-to-end performance

Table 3.3: Projected End-to-end performance

	Naive Extrapolation			Conditional Extrapolation			Experimental Measurement		
Method	Precision	Recall	FM	Precision	Recall	FM	Precision	Recall	FM
LBP [25]	12.45	32.48	18.01	9.89	28.31	14.66	9.45	27.06	14.01
HOG	15.24	31.14	15.24	9.61	27.52	14.25	8.78	25.13	13.01
Qaud Tree	7.23	22.13	10.90	6.8	19.54	10.11	-	-	-
Random	1.38	1.33	1.44	0.53	1.53	0.79	0.46	1.32	0.68
PHOCNET [94]	71.52	85.09	77.72	24.78	70.94	36.73	-	-	-
FischerCCA [4]	52.05	75.86	61.74	21.44	61.37	31.78	19.37	55.44	28.71
DTW [77]	12.56	21.91	15.97	6.75	19.34	10.01	-	-	-

Given a word segmentation method, segmentation-based word-spotting can be used in an end-to-end scenario. Experiments were conducted using an established word segmentation method[66]. Following the protocols defined in [49], end-to-end performance is measured by the F-Measure (harmonic mean) of the detection rate (DR) and the recognition accuracy (RA). A correct match is one that has the exact same transcription as the groundtruth and at the same time, the rectangle in which the word was detected and the rectangle of the groundtruth have an IoU of at least 50%. The extrapolation of end-to-end performance can happen under the assumption that the accuracy each method

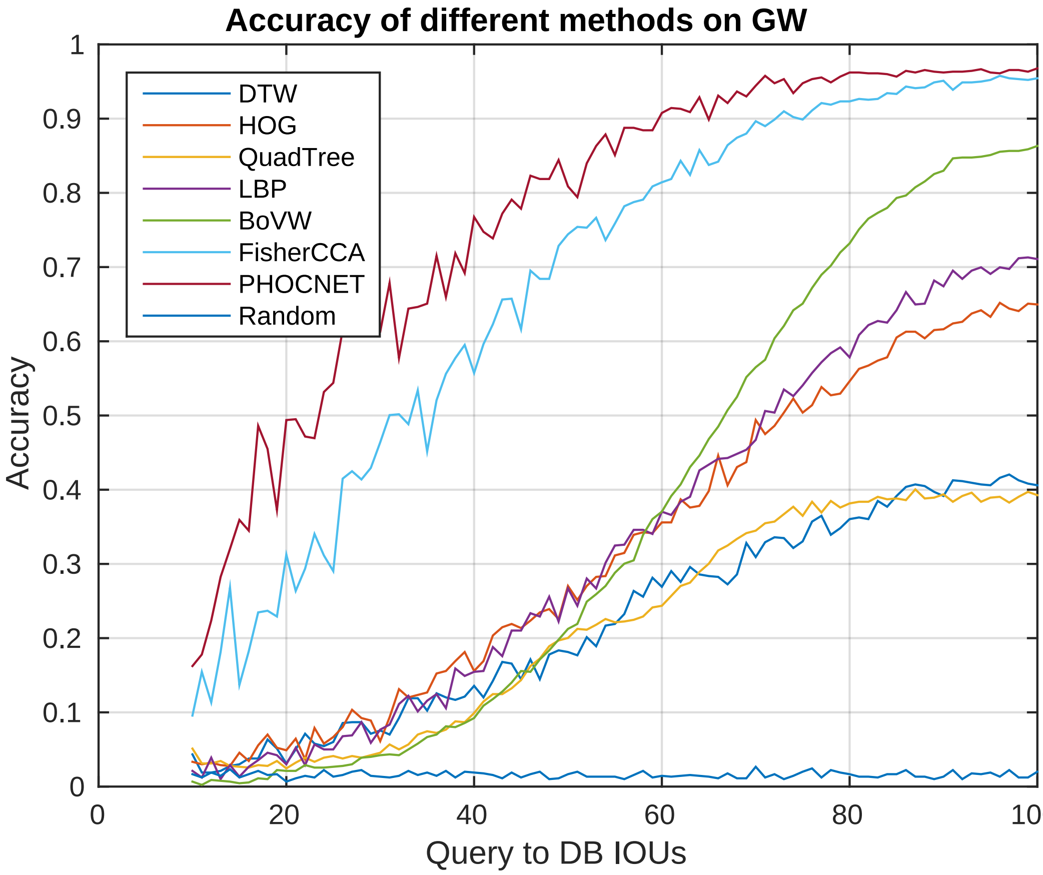


Figure 3.11: Accuracy for improper segmentation on GW dataset.

demonstrates at a given IoU distortion is also the probability of a correct retrieval given that IoU (see Equation 3.4).

$$\forall x \in [0,1], P(CR|IoU = x) = Acc_{method}(x) \tag{3.4}$$

where CR is having correct retrieval and $Acc_{method}(x)$ is the accuracy a method demonstrated for a specific x. In Table 3.3 the DR, RA, and F-Measure of end-to-end performance obtained by using each word-spotting method are presented for the GW dataset. The measurements are organized in three groups.

The first group of measurements is called naive extrapolation. The DR is approximated by taking the mean IoU of the best match that all groundtruth rectangles obtained and measuring the accuracy each method demonstrated for the given IoU. Respectively, the RA of the naive extrapolation for a method is the accuracy of the method for an IoU equal to the mean of the best IoU each detected sample obtained against all groundtruth samples. The F-Measure based on the naive extrapolation is the harmonic mean of the naive DR and the naive RA. With the use of (3.4), the naive extrapolated DR and RA are computed with (3.5) and (3.6) respectively.

$$DR_{naive} = Acc_{method}(\int_0^1 f_{DR}(x)dx) \tag{3.5}$$

$$RA_{naive} = Acc_{method}(\int_0^1 f_{RA}(x)dx) \tag{3.6}$$

The naive extrapolation assumes a perfect co-variance between the IoU of the segmentation each sample has and the probability of that sample being correctly retrieved. The above assumption is the same as assuming that all accuracy plots unlike that of Fig.3.11 are straight lines. The actual plot Fig.3.11 do demonstrates how weak this assumptions is.

A more elaborate extrapolation can be obtained by estimating the conditional probability of a correct retrieval given a Probability Density Function (PDF). Given a matrix M where rows represent grountruth rectangles, columns represent detected rectangles, and each cell contains the IoU of those rectangles. The PDF of IoU for every detected sample $f_{RA}(IoU)$ can be estimated as the distribution of the maximum elements in every column. Respectively the PDF of IoU for every sample in the groundtruth $f_{DR}(IoU)$ can be estimated as the distribution of the maximum elements in every row. In connection with (3.4) the extrapolated DR_{cond} and RA_{cond} can be estimated by Equations 3.7 and 3.8.

$$DR_{cond} = \int_0^1 P(CR|f_{DR}(x))dx \tag{3.7}$$

$$RA_{cond} = \int_0^1 P(CR|f_{RA}(x))dx \tag{3.8}$$

The third group of measurements, called the experimental measurement, consists of the empirical measurement. To obtain these measurements, the word segmentation method was used as the first stage and each segmentation-based method was plugged in as the second stage of a two level end-to-end pipeline. The resulting detected words and bounding boxes are passed to the evaluation protocol.

As can be seen, conditional extrapolation predicts quite well the empirical observation. The naive extrapolation, on the other hand, is a bad predictor of the empirical measurement. The fact that the conditional extrapolation has been demonstrated to be a good predictor of the empirical end-to-end performance indicates that IoU is enough to predict the quality of the segmentation.

It should be pointed out that in a broad experimental setup, containing several segmentation methods and several word-spotting methods, the conditional extrapolation and the empirical measurement have different computational costs. Computing the best end-to-end system for n segmentation methods and m word-spotting methods would require $m \times n$ executions of the word spotting methods. On the other hand, conditional extrapolation can obtain the best end-to-end performance with $100 \times m$ regardless of how many segmentation methods there are. As conditional extrapolation demonstrated to be a good predictor, it can be used whenever we want to select among many segmentation variants.

In Table 3.3 the DR, RA, and F-Measure of each word-spotting method employed in an end-to-end scenario is presented for the GW dataset. Both experimental and estimates of end-to-end performance are presented. As can be seen in the table, the conditional extrapolation predicts performance that is quite similar to the empirical measurements, while the predictions from the naive extrapolation are consistently much worse. It is worth noticing that both extrapolation methods, when compared to empirical evidence, demonstrate a bias, predicting higher performance than actually achieved. The conditional extrapolation prediction uses the IoU between a segmented word and the best matching ground-truth box as the sole factor that determines the end-to-end performance. The extent to which the conditional extrapolation prediction is correlated with the empirical end-to-end measurements, indicates the validity of the assumption that *IoU is a valid quantification of the quality of segmentation* for the given methods, data, and evaluation protocol employed.

3.3.7 Method Independence

In this section we analyze the independence of the methods. First, we study the independence in terms of retrieval, i.e. how two methods are complementary. Second, we apply such complementarity in a late fusion scheme.

3.3.7.1 Independence Measurement

An other part of the experimental analysis of the methods is about gaining insights on the independence of the proposed methods. The motivation for such an analysis is that it can provide an intuition for the potential that the optimal fusion of methods could provide. Two measurements are provided as indicators of the independence of the examined methods. The first is Spearman's footrule [26] measured between the retrievals of two methods. In Fig. 3.12 Spearman's footrule applied on all pairs of methods for the GW test-set can be seen. Spearman's footrule quantifies the similarity between rankings of samples for every query and therefore it is directly influenced by a methods performance. The lower the value the more similar it is and vice-versa. The diagonal in the matrix for this reason has

Spearman's footrule does not account for the performance two methods demonstrate. In order to provide a measurement of how similarly do two methods perform a second

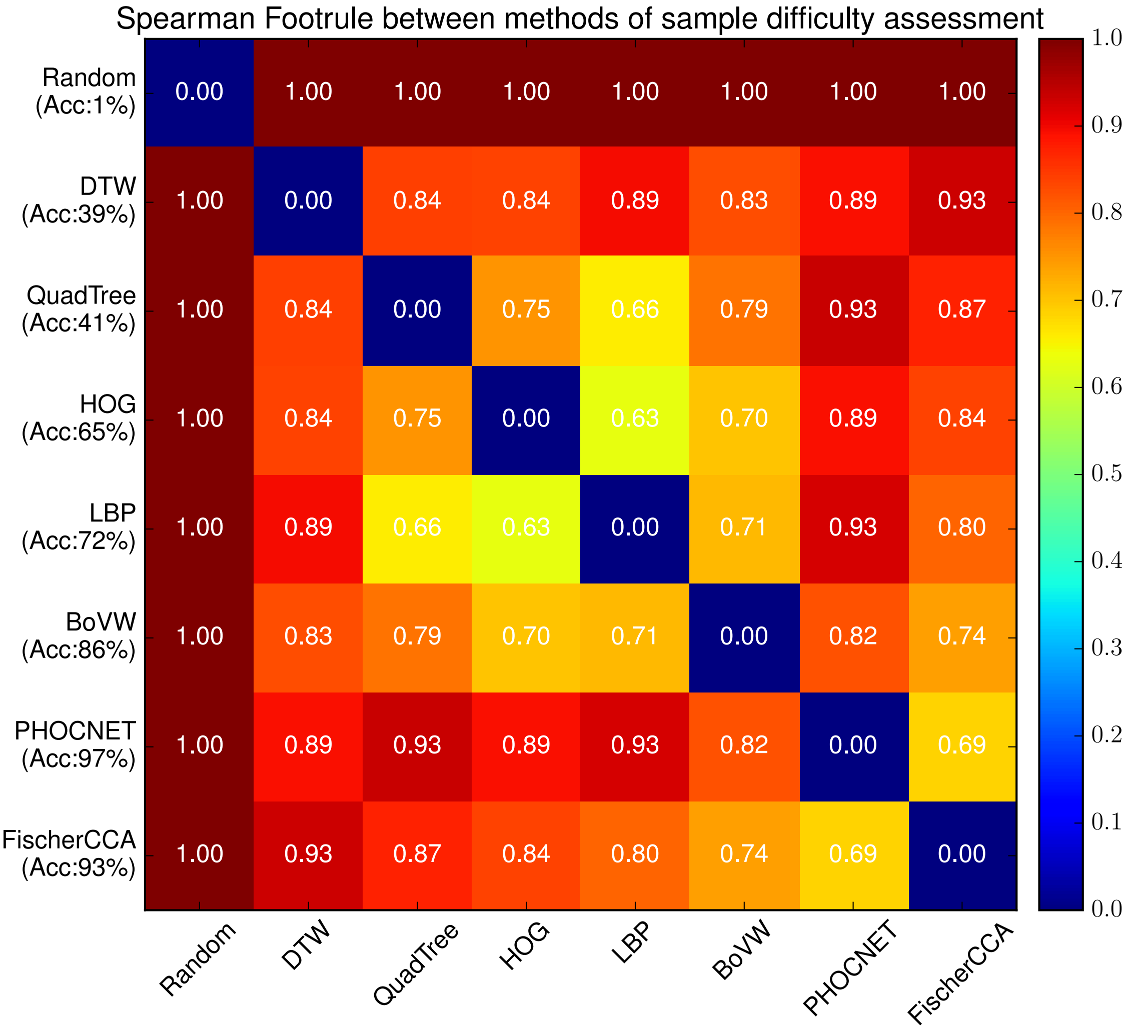


Figure 3.12: Normalised Spearman's footrule between methods. Higher values mean more disagreement in ranking the samples from easy to hard.(The figure is best viewed in color and with PDF magnification)

quantification of the independence of analyzed methods is introduced, which compensates differences in the methods performance. For every method, the average precision is calculated for every query. The query samples are then labelled as 0 if their average precision is lower than the median and as 1 if their average precision is greater than the median average precision. As an estimate of the dependence, Pearson's correlation is measured on the query-samples labellings. This measurement in effect quantifies the agreement of two methods on which are the easy samples to retrieve and which the hard. In Fig. 3.13 the correlations on the labelling of the GW test-set can be seen. In both figures, a random retrieval is added to provide context and scale.

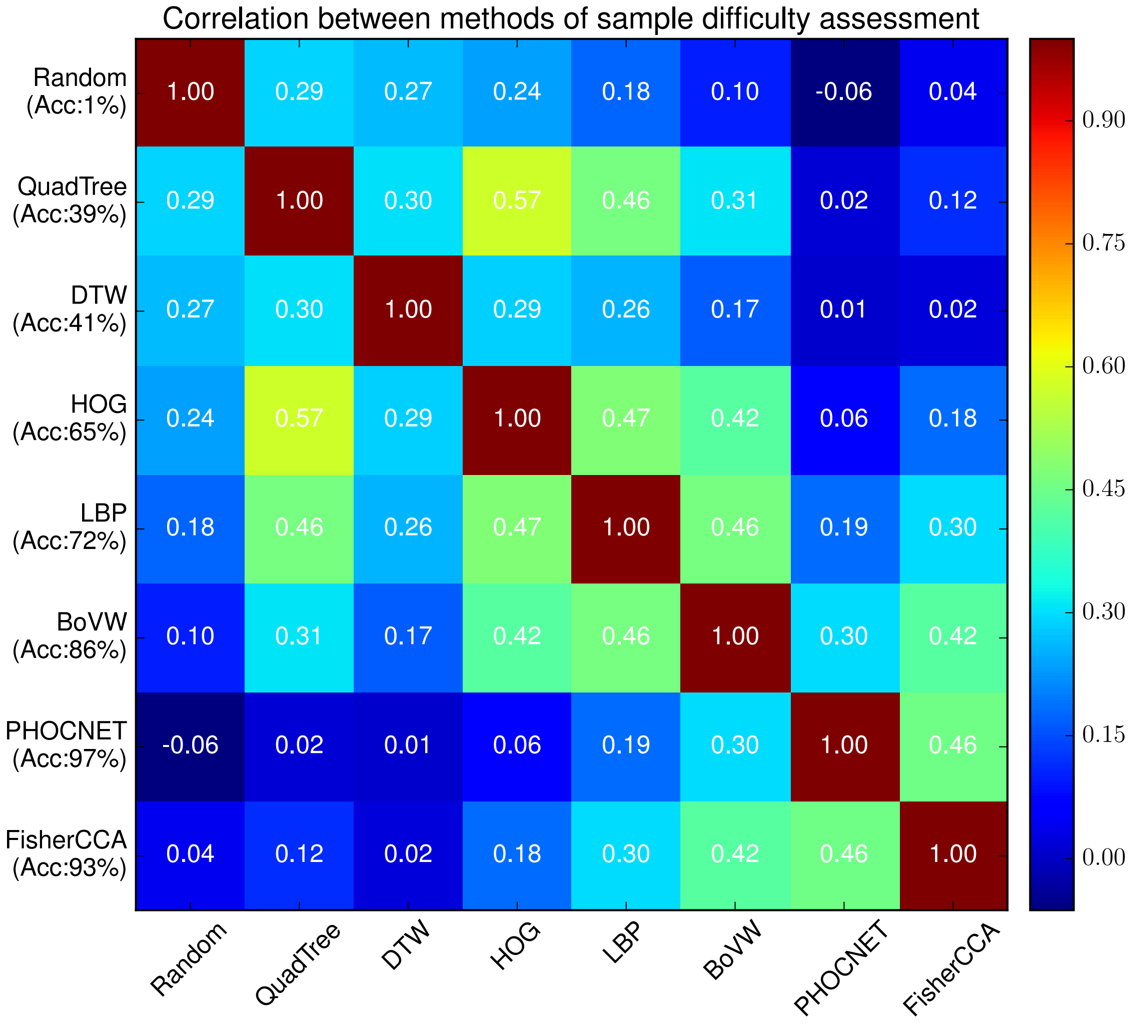


Figure 3.13: Method agreement in easy and hard samples. Correlation between methods on partitioning the samples in to easy and hard. (The figure is best viewed in color and with PDF magnification)

3.3.7.2 Method Fusion

Although early fusion with a supervised learning framework would better use the method independence, development of such a method would go beyond the scope of this assay. A naive late fusion of all methods is presented as an indication of the potential in method fusion. In (3.9) the fusion of a pair of methods M_1, M_2 can be seen.

$$\max \forall n \in [1..100] R(\frac{n * m_1 + (100 - n) * m_2}{100}) \tag{3.9}$$

where m_1 and m_2 are the distance matrices of each method and $R()$ is the mAP obtained given a distance matrix. The fusion reported is the best weighted sum of distance matrices for every pair of methods. In Fig.3.14 the results of all fusions can be seen. Each cell in the table contains the mAP of the best fusion as well as the improvement the fusion introduces over the best of the two methods. In Fig. 3.15 the optimal coefficients for each method $M1$ can be seen. The high performing methods see negligible improvement through fusion with other methods. The most significant improvement achieved by the employed late fusion is between LBP and HOG. We were quite surprise to see the fusion result between FisherCCA and PHOCNET, which could be nice to investigate further in details in future.

3.4 Conclusions and Discussion

3.4.1 Conclusions

Several conclusions can be drawn from the experiments presented above.

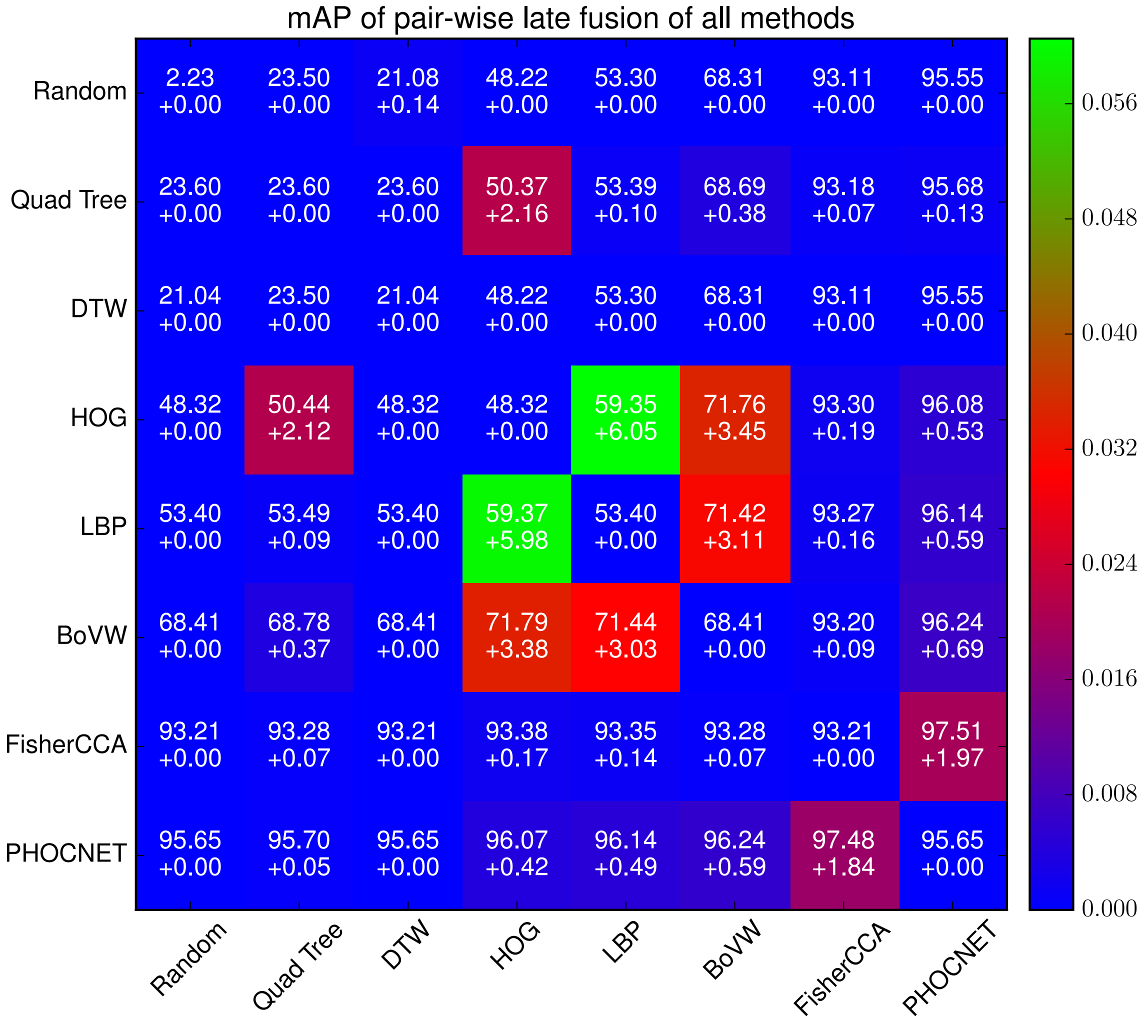
The most important property of methods is the level of learning they employ. The methods using advanced machine learning techniques PHOCNET and FischerCCA stand out in performance. PHOCNET demonstrates superior performance but FischerCCA appears to be more robust both on unseen data as well as on significant distortions on the BCN segmentation. Other than using supervised learning, both these methods also share the PHOC representation to which high performance should also be attributed. It is uncontested that if enough groundtruth and computational resources are available, one of these two methods should be used.

BoVW is the sole method analyzed employing unsupervised learning. Its performance is high and it might be the optimal solution in cases such as groundtruth for data not being available.

LBP and HOG features performed well given that they are pure feature extraction methods. More than that, their speed as well as their out-of-the-box applicability of heterogeneous data is well demonstrated in the cross-dataset experiment. At the same time LBP proved to be the fastest method and HOG a close second. The high improvement the fusion of these methods provides is also an interesting find.

DTW method compared to other methods performs poorly both in mAP as well as execution time. It is the only method that has a quadratic complexity with respect to the word image width and is intractable for any dataset of a significant size.

In what concerns performance measurement, the two metrics that demonstrated better behavior are mAP and rPrecision. These two metrics demonstrated remarkable correlation between them and although rPrecision appears less saturated, mAP was preferred because it is more broadly used. The end-to-end performance extrapolation



	Random	Quad Tree	DTW	HOG	LBP	BoVW	FisherCCA	PHOCNET
Random	2.23 +0.00	23.50 +0.00	21.08 +0.14	48.22 +0.00	53.30 +0.00	68.31 +0.00	93.11 +0.00	95.55 +0.00
Quad Tree	23.60 +0.00	23.60 +0.00	23.60 +0.00	50.37 +2.16	53.39 +0.10	68.69 +0.38	93.18 +0.07	95.68 +0.13
DTW	21.04 +0.00	23.50 +0.00	21.04 +0.00	48.22 +0.00	53.30 +0.00	68.31 +0.00	93.11 +0.00	95.55 +0.00
HOG	48.32 +0.00	50.44 +2.12	48.32 +0.00	48.32 +0.00	59.35 +6.05	71.76 +3.45	93.30 +0.19	96.08 +0.53
LBP	53.40 +0.00	53.49 +0.09	53.40 +0.00	59.37 +5.98	53.40 +0.00	71.42 +3.11	93.27 +0.16	96.14 +0.59
BoVW	68.41 +0.00	68.78 +0.37	68.41 +0.00	71.79 +3.38	71.44 +3.03	68.41 +0.00	93.20 +0.09	96.24 +0.69
FisherCCA	93.21 +0.00	93.28 +0.07	93.21 +0.00	93.38 +0.17	93.35 +0.14	93.28 +0.07	93.21 +0.00	97.51 +1.97
PHOCNET	95.65 +0.00	95.70 +0.05	95.65 +0.00	96.07 +0.42	96.14 +0.49	96.24 +0.59	97.48 +1.84	95.65 +0.00

Figure 3.14: mAp increase by weighted linear fusion of methods. The final mAP achieved is marked as a percentage along with the increment in performance due to the fusion. (The figure is best viewed in color and with PDF magnification)

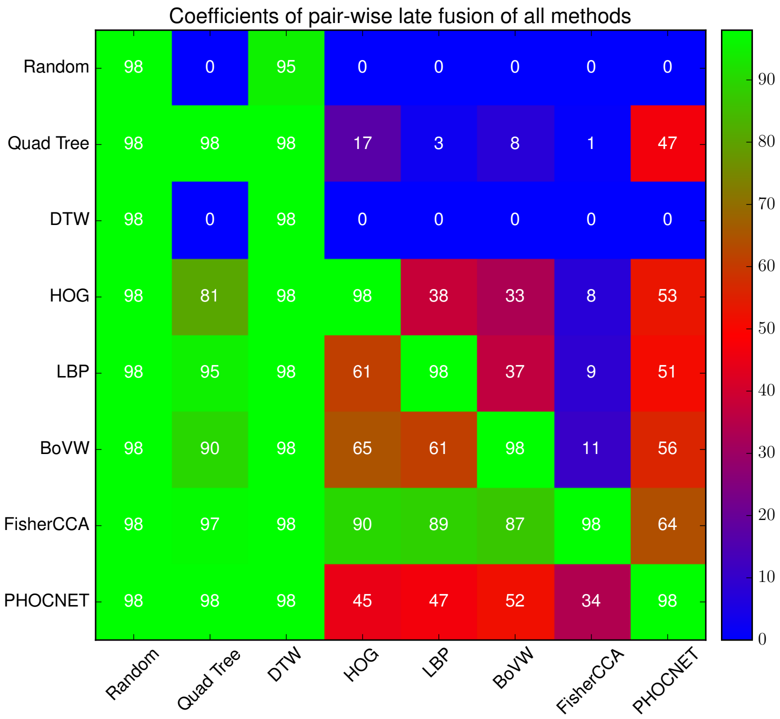


Figure 3.15: Coefficients of the optimal fusion for every pair of methods.

measurements demonstrate that the actual curves of IoU to performance need to be computed as the assumption they are correlated brings unpredictable errors. The method independence analysis provided insights into how different methods are between them. The following naive late fusion could not harvest the variability of independent methods, in opposite, very similar methods such as LBP and HOG demonstrated the greatest improvement by naive late fusion. As for the future work we will try to increase the document types as well as the number of samples.

3.4.2 Discussion

The experimental work presented in this chapter revolves around the fundamental hypothesis that IoU is a valid quantification of the quality of segmentation. The hypothesis is validated by the fact that the conditional extrapolation of end-to-end pipeline performance predicts well the empirical measurements. The validation of the IoU hypothesis, indicates that the distortion engine used, although not perfect, is realistic enough to generate good estimates.

When thinking of real world applications in the near future, representative groundtruth for a digital library of substantial size should be considered unfeasible. Methods that rely heavily on vast quantities of annotated data, are probably inapplicable. The cross-dataset experiment models this scenario better. In the cross-dataset experiment, LBP is a close second to FischerCCA mAP while being three times faster. This makes LBP as good if not better for such cases.

Top performing networks, achieve almost perfect performance and although both GW and BCN datasets could be considered "solved", our experiments suggest that the evaluation protocol using only perfect segmentation is probably too lenient.

If one looks at each of the publications in which the compared methods are described, he will observe some differences between the performances reported there and the performances reported in this study. These differences happen because of deviations in the experimental procedures followed in each work, but in some cases are greater than difference in performance methods demonstrate between them. A fundamental contribution of this work, was applying the exact same evaluation protocol on all methods. More than this, the evaluation protocol was designed even before selecting the methods to compare. The principal design goal was modularity and to maximize the number of methods that can be compared.

Chapter 4

Writer Identification

4.1 Introduction

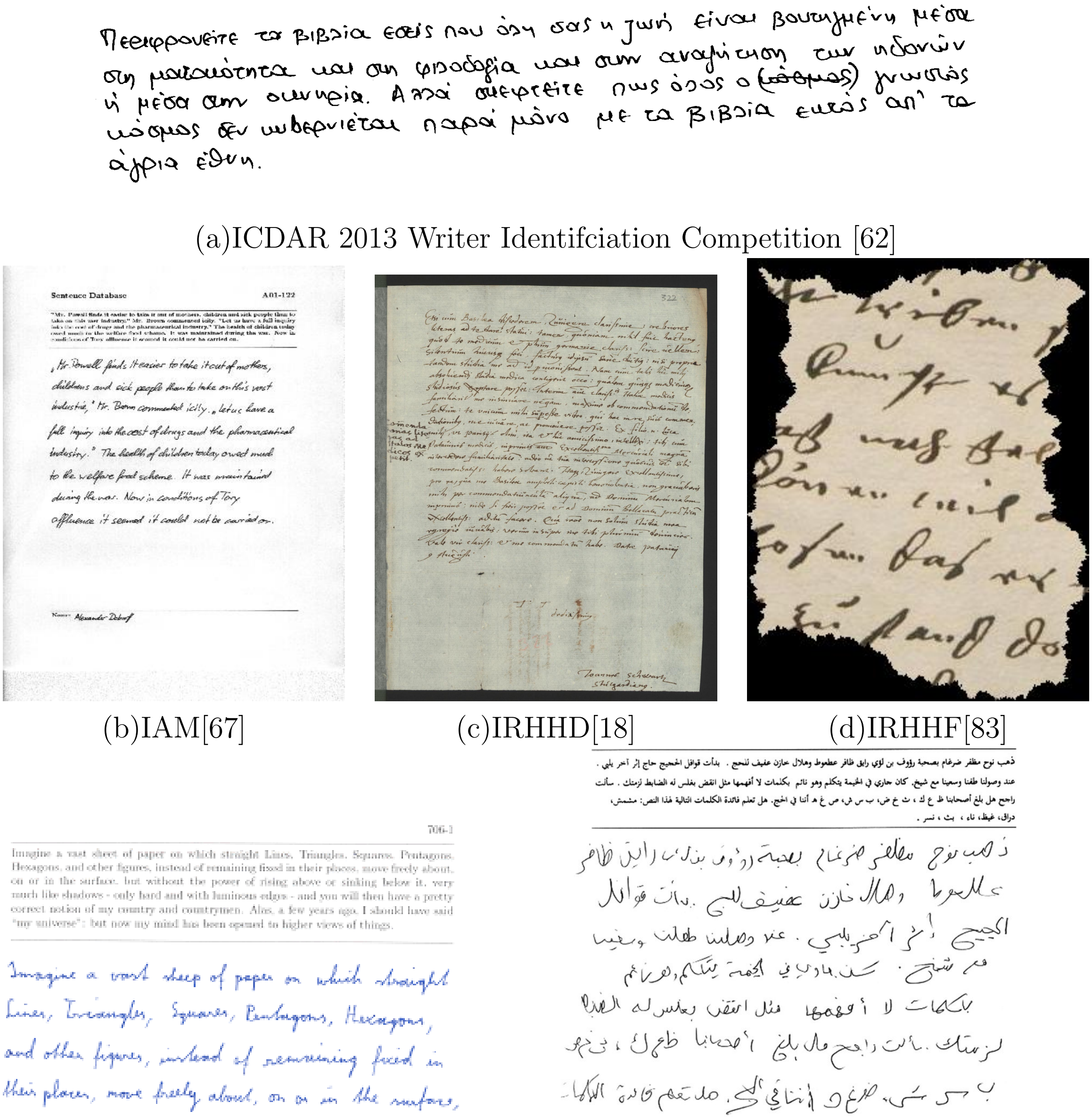
4.1.1 Problem Description

Writer identification is the problem of identifying the authorship oftext samples based on an index of examples of text written byknown authors [13]. It has a long tradition in forensics where it has been accepted by the court as evidence for more than a century. From a pattern recognition perspective, three variations of the problem of writer identity are defined: writer identification, writer verification, and writer retrieval. Writer identification is the most popular of these, and in most cases a method can be modified from solving one to solving another with little effort. From a Document Image Analysis (DIA) perspective there are other applications such as scribe identification for historical documents . Handwriting has been considered as a behavioral biometric [42, 8, 13], although experiments on disguised handwriting have proved to confuse both graphonomists, and automatic systems [64]. Despite the attention writer identification has received from DIA researchers in recent years, it remains a difficult problem due to variations in writing style and conditions, and the myriad problems arising from variations in image quality due to document degradation and other incidental factors.

4.1.2 Use cases

4.1.2.1 Historical Documents

The most important dichotomy of writer identification systems is between historical and conteporay writer identification. Historical writer identification is principally done to help researchers in the humanities to trace the origins and history of documents and assist in general paleographers. In recent years new datasets used in historical writer identification retrieval competitions, have raised the number of of identitities to several thousants. A



Περιφρονείτε τα βιβλία εσείς που όλη σας η ζωή είναι βουτηγμένη μέσα στη ματαιότητα και στη φιλοδοξία και στην αναζήτηση των ηδονών ή μέσα στην οκνηρία. Αλλά σκεφτείτε πως όλος ο (~~κόσμος~~) γνωστός κόσμος δεν κυβερνιέται παρά μόνο με τα βιβλία εκτός απ' τα άγρια έθνη.

(a)ICDAR 2013 Writer Identifciation Competition [62]

Sentence Database A01-122

"Mr. Powell finds it easier to take it out of mothers, children and sick people than to take on this war industry." Mr. Brown commented icily. "Let us have a full inquiry into the cost of drugs and the pharmaceutical industry." The health of children today owed much to the welfare food scheme. It was maintained during the war. Now in conditions of Tory affluence it seemed it could not be carried on.

„Mr. Powell finds it easier to take it out of mothers, childrens and sick people than to take on this vast industrie," Mr. Brown commented icily. „let us have a full inquiry into the cost of drugs and the pharmaceutical industry." The health of children today owed much to the welfare food scheme. It was maintained during the war. Now in conditions of Tory affluence it seemed it could not be carried on.

Name: Alexander Dobruf

(b)IAM[67]

(c)IRHHD[18]

(d)IRHHF[83]

706-1

Imagine a vast sheet of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface, but without the power of rising above or sinking below it, very much like shadows - only hard and with luminous edges - and you will then have a pretty correct notion of my country and countrymen. Alas, a few years ago, I should have said "my universe": but now my mind has been opened to higher views of things.

Imagine a vast sheep of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface,

(e)CVL[51]

ذهب نوح مظفر ضرغام بصحبة رؤوف بن لؤي رابق ظافر عطعوط وهلال حازن عفيف للحج . بدأت قوافل الحجيج حاج إثر آخر يلبي .
عند وصولنا طفنا وسعينا مع شيخ. كان حاري في الخيمة يتكلم وهو نائم بكلمات لا أفهمها مثل انقض بغلس له الضابط لزمتك . سألت
راجح هل بلغ أصحابنا ظ ع ك ، ث خ ض، ب س ش، ص غ ه أننا في الحج. هل تعلم فائدة الكلمات التالية لهذا النص: مشمش،
دراق، غيظ، ناء ، بث ، نسر .

ذهب نوح مظفر ضرغام بصحبة رؤوف بن لؤي رابق ظافر
عطعوط وهلال خازن عفيف للحج . بدأت قوافل
الحجيج إثر آخر يلبي . عند وصولنا طفنا وسعينا
مع شيخ . كان حاري في الخيمة يتكلم وهو نائم
بكلمات لا أفهمها مثل انقض بغلس له الضا
لزمتك . سألت راجح هل بلغ أصحابنا ظ ع ك ، ث خ ض
ب س ش ، ص غ ه أننا في الحج . هل تعلم فائدة الكلما

(f)KHATT[63]

Figure 4.1: Samples from Writer Identification datasets

mojor problem in historical datasets is that the background texture, the pygment color, material degradations, and varius other factors, can be suspected to be responcible for the performace measured on datasets. To put it simply, the question is: *"are the systems finding similar writers or similar pages"*?. Recent competitions [18, 83] have addressed this issue by having single document and multidocument modalities. Along with writer identification there are also related tasks such as dating as classifying the script style as in the CLaMM [20] dataset and the 2017 ICDAR competition.

4.1.2.2 Contemporary Handwriting Writer Identification

Contemporary writer identification is mostly applied to graphonomics and other biometric applications. Contemporary handwritting datasets are usually curated under controlled and homogenious conditions. In Fig. 4.1 (a), (b), (e), and (f) samples from contemporary datasets can be seen. These datasets are usually aquired in controlled conditions beeing consistent in the writing medium, the aquisition resolution, and even the text beeing writen. This consistency in the sample aquisition process, allows us to infer that whatever performance is the actual handwriting style as there no obvious other confounding factors. These datasets allow for a more profound understanding of the actual nature of handwriting and the detectabillity. In what concerns the biometric nature of handwriting, according to [24] "*Any human physiological or behavioral trait can serve as a biometric characteristic as long as it satisfies the following requirements: 1) Universality. Everyone should have it; 2) Distinctiveness. No two should be the same; 3) Permanence. It should be invariant over a given period of time; 4) Collectability*". While we can assume universality in most social contexts and writer identification systems have prooven distinctive with quite high accuracy, to our knowledge there are no datasets containing handwriting samples aquired over different points in time, and therefore there is no proof of the permanence of the handwriting style over time or eg in diferent hours of the day, under different stress levels etc. In what concerns collectabillity, in most scenarios, aquiring a sample can always be done as long as the subject concents, which acts like an ethical safeguard whith respect to the use of biometrics.

4.1.3 Image Retrieval vs. Image Classifcation

Depending on the exact aplication in mind, writer identification can be framed in many diferent ways and optimal methods might differ from case to case. The most strainghtforward aporach is to model this problem as a supervised-learning classification problem as training image claassifiers is quite simple and performs well. Unfortunately framing writer identification as a classification problem has several failings:

- Every time the database changes even slightly, the hole model has to be retrained

- The ouputs of the model are not interpretable
- Several samples per identity are required
- Discriminative classifiers don't scale well to the extremely large class numbers require

In most cases writer identification is framed like an image retrieval problem: rank a a database of images containing writing samples of known identities by a similarity metric with respect an handwriting sample query image. A retrieval method can always be perceived as a classification method if the database is labeled by a KNN classifier and therefore retrieval systems are also classification systems. In the retrieval setting, it is assumed that the data on which the system will be employed or, in the case of experiments the test-set, identities are not known in any away when the method is tuned. Datasets curated for the retrieval optionaly provide a train-set with different identities from the train-set so that methods can be trained if they need to. When dealing with very large sets of identities, whether it is historical or contemporary, it is very important that there are no duplicate identities labeled as diferent ones; this sets a limit into the process with which databases can be agregated.

4.1.4 LBP for writer identification

LBP are dense local texture descriptors that can be used to describe the local structure of images [74]. They have been successfully applied to many of the major computer vision problems, as well also been applied to specific problems in Document Image Analysis, including optical font recognition and writer identification. LBP were originally designed for graylevel images, and despite their widespread application to bilevel document images, it remains unclear how LBP should be computed on such images in order to remain discriminative and robust to noise.

In this work we introduce Sparse Radial Sampling LBP (SRS-LBP), a variant of LBP that is better suited for thetask of writer identification and text-as-oriented-texture classification in bilevel images in general. Our main contribution is the introduction of sparse radial sampling of the circular patterns used for LBP construction. This allows sampling of patterns up to very large radii for each pixel at low computational cost, and also avoiding vocabulary compression techniques such as rotation invariant or uniform patterns commonly applied to standard LBP representations. We show that using a single local descriptor, our SRS-LBP variant densely extracted and pooled over the entire image, results in a low-dimensional feature representation that yields SotA performance at a fraction of the cost of other techniques. Our representation is compact and extremely efficient to compute.

4.1.5 State of the Art Evolution

Automatic writer identification has been researched for many decades. While at some point methods were qualifying offline as oposed to online writer identification [13], in recent years, writer identification unless qualified, refers to refers to offline. Recent online methods have been published using CNNs [105] but this study focuses on offline data.

Due to the fast pace of innovation, heterogenious evaluation protocols, and differently framed use-cases, the state of the art can not be defined in an absolute sence but it is rather nuanced. Methods and their performance vary in performance, computational resources, the tradeoff between performance for the first match and the last match. Methods in the literature have been evolving and could be perceived as belonging to families of methods although this organisation.

4.1.5.1 Hinge based

In 2007 Schomaker et al. propose allographic features for writer identification their features were based on radial PDF which were encoding texture features. For an overview of previous to this work on writer identification, we refer to their excellent survey [13]. This method would extract the contours of every connected component and actually would even segment multiletter of cursive script into letter sized segements. Their best performing custom-made features were called contour-hinge which were encoding two angles for every point on the contours and therefore an aspect of local curvature as well as the local slant. Another variant of this family of methods [36] was proposed in a segmentation-free setting by sampling This family of methods has been further developed recently [43] and participated in the 2019 writer identifcation competition but I can no longer be considered SotA.

4.1.5.2 Codebook based

These methods operate by building a codebook of shapes which are clusters of graphemes or similar sized shapes, features are than extracted from these codebooks. The idea of a codebook of shapes aproximating letters was previously in the similar task of language detection [**?**]. The first of these methods was presented ny Jain and Doermann [46]. In [47] proposed pseudo letters ,jain2014 which was the SotA until 2015. In [30] Fiel compiled codebooks using SIFT features instead of pseudo words.

4.1.5.3 Texture generation

The principal aproach of these methods is to generate texture images from a sample and than perform texture analysis on them. This aproach was poroposed in [32, 33] where GSCM and gabor features were used to analyse texture images. Bertolini et al. [8] used the same aproach employing LBP in a comparative study with Local Phase Quantization and concluded that LPQ are performing better. While the specific method was applied successfully in music scores, for generic writer identification SotA performance has not been demonstrated.

4.1.6 CNN based versions

CNN can not be directly employed for writer identification as the retrieval scenario makes the use of discrimitave CNNs unusable. In [16] Christlein employes CNN features trained discriminately on the classes of the train-set and than encodes them with the use of GMM supervectors in order to retrive identities in test-set. In [17] clustering on document images is used to create surogate classes on which ResNets [41] are trained discriminatevely and than their penultimate layer is used with a VLAD encoding. In [14] ResNets are used in pipelines with exemplar SVM [65] and achieve SotA over contemporary data. In [19] mdodified generalised max pooling [70] and employed as a differentiable neural network layer; the network was than used in end-to-end training for handwrtting style classification.

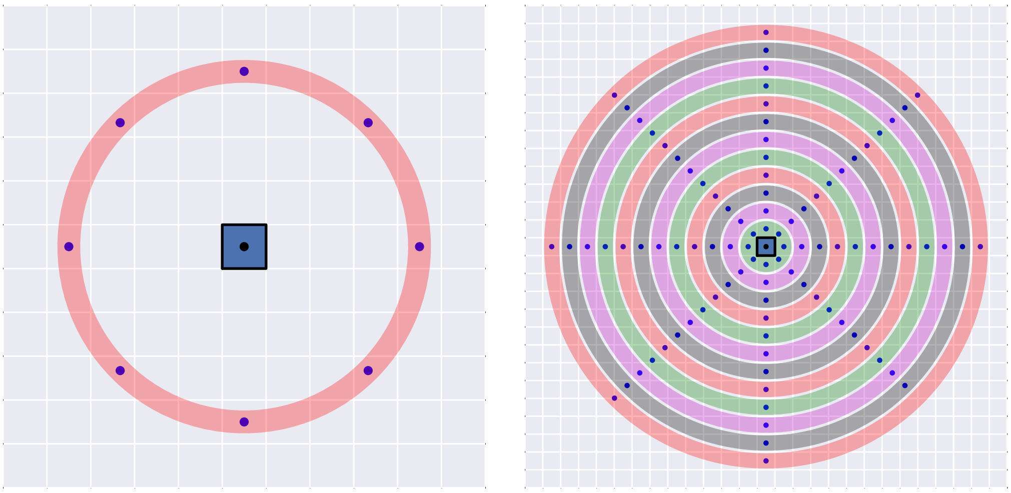
4.2 Sparse Radial Sampling LBP

4.2.1 My contribution with respect to the State-Of-The-Art

Our adaptation of the LBP consists of replacing the sign operator with a threshold statistically derived from each image, the use of sparse radial sampling at each radius, and the use of very large radii when computing LBP. Though we concatenate LBP histograms extracted at many radii, sparse radial sampling ensures that the final LBP features are compact.

We apply my SRS-LBP to writer identification using a standard LBP pipeline. SRS-LBP are computed at each location in an image, and these features are pooled over the entire page image. Our approach requires no character segmentation and is based on a single, compact feature that is extremely efficient to extract. In this sense it stands out with respect to SotA approaches based on complex character segmentation, clustering, and extraction of multiple feature descriptors [48].

LBP feature extraction consists of two principal steps: the LBP transform, and the pooling of LBP into a histogram representation of an image. The LBP transform maps each pixel to an integer code representing the relationship between the center pixel and the pixels of its neighbourhood. It encapsulates the local geometry at each pixel by encoding



(a) $\mathrm{LBP}_{4,8}$ (b) $[\mathrm{LBP}_{1,8}, ..., \mathrm{LBP}_{12,8}]$

Figure 4.2: Proposed LBP transform sampling patterns used in in conjunction.

binarized differences with pixels of its local neighbourhood:

$$\mathrm{LBP}_{P,R,t} = \sum_{p=0}^{P-1} s_t(g_p - g_c) * 2^p, \tag{4.1}$$

where g_c is the central pixel being encoded, g_p are P symmetrically and uniformly sampled points on the periphery of a circular area of radius R around g_c, and s_t is a binarization function parametrized by t. The sampling of g_p is performed with bilinear interpolation. The use of local differences in (4.3) endows LBP with a degree of illumination invariance.

In our LBP definition, s is a simple threshold:

$$s_t(x) = \begin{cases} 1 & : x \geq t \\ 0 & : x < t \end{cases}, \tag{4.2}$$

where t, which in the standard definition is considered zero, is a parameter that determines when local differences are considered ”big enough” for consideration.

$\mathrm{LBP}_{P,R,t}$ can be seen as a transform from the graylevel domain to a domain of discrete labels encoded over a vocabulary of 2^P integers.

In this section we describe our approach to LBP extraction based on sparse radial sampling. We follow the development and notation of [74].

4.2.2 The LBP transform

LBP feature extraction consists of two principal steps: the LBP transform, and the pooling of LBP into a histogram representation of an image. The LBP transform maps each pixel to an integer code representing the relationship between the center pixel and the pixels of its neighbourhood. It encapsulates the local geometry at each pixel by encoding binarized differences with pixels of its local neighbourhood:

$$\mathrm{LBP}_{P,R,t} = \sum_{p=0}^{P-1} s_t(g_p - g_c) * 2^p, \tag{4.3}$$

where g_c is the central pixel being encoded, g_p are P symmetrically and uniformly sampled points on the periphery of a circular area of radius R around g_c, and s_t is a binarization function parametrized by t. The sampling of g_p is performed with bilinear interpolation. The use of local differences in (4.3) endows LBP with a degree of illumination invariance.

In our LBP definition, s is a simple threshold:

$$s_t(x) = \begin{cases} 1 & : x \geq t \\ 0 & : x < t \end{cases}, \tag{4.4}$$

where t, which in the standard definition is considered zero, is a parameter that determines when local differences are considered "big enough" for consideration.

$\mathrm{LBP}_{P,R,t}$ can be seen as a transform from the graylevel domain to a domain of discrete labels encoded over a vocabulary of 2^P integers. In Fig. **??** sampling patterns of popular and proposed LBP can be seen where g_c is marked as a black dot and g_p are marked as blue dots.

4.2.3 Sparse sampling LBP on bilevel images

The original LBP was designed for graylevel images. Though text images are often fundamentally bilevel by nature, the bilinear interpolation used to extract neighbouring pixel values g_p renders pixels non-binary and standard LBP is (at least mathematically) applicable. However, images of a bilevel nature such as text even when they are acquired on the graylevel domain do not benefit much from illumination invariance. Another problem is that large $g_c - g_p$ differences are more rare than small ones and so treating both of them the same introduces noise.

Rather than use arbitrary or empirically derived threshold t to re-binarize differences in the computation of LBP in (4.3), we propose to apply Otsu's method to estimate optimal threshold $\hat{t}$ from the statistics of image differences themselves:

$$\hat{t} = \arg\min \omega(d_{1,t})\sigma^2(d_{1,t}) + \omega(d_{t,P})\sigma^2(d_{t,P}) \tag{4.5}$$

where $d_{1,t}$ is the set of $|g_c - g_p|$ less than threshold $\hat{t}$, $d_{t,P}$ is the set of $|g_c - g_p|$ greater than the threshold, ω is the probability of $d_{...}$ and σ^2 its variance. The use of $\hat{t}$ yields a unified solution to both of these problems. The Otsu threshold of the differences effectively separates the significant differences from insignificant ones. Note that this formulation works for bilevel and graylevel source imagery.

Sparse radial sampling is integrated into our descriptor by holding constant the number of points sampled at each radius:

$$\text{SRS-LBP}_R = \text{LBP}_{8,R,\hat{t}}. \tag{4.6}$$

Keeping $P = 8$ constant (i.e. sparse radial sampling) allows us to sample more radii while maintaining a compact code.

4.2.4 Processing pipeline

Our complete processing pipeline is comprised of the following steps:

1. **SRS-LBP transformation:** each image pixel is transformed to several SRS-LBP according to (4.3) and (4.6). This encodes the input image as several 8-bit images (one for each radius).

2. **SRS-LBP pooling:** a histogram of SRS-LBP codes is computed for each radius. We discard the zero pattern which corresponds to foreground- and background-only patterns, and then L1 normalize and concatenate all histograms. The result is a block-normalized descriptor of size 256 × $|R|$, where $|R|$ is the number of radii.

3. **PCA projection**: the block-normalized descriptor is projected onto the first N principal components computed through Principal Component Analysis (PCA).

4. **Normalization:** the Hellinger kernel is applied to the projected descriptor, followed L2 normalization. This combination has been shown to improve performance of a variety of image recognition techniques and specifically on writer identification [15].

Note that we use none of the standard vocabulary compression techniques such as rotation invariance or uniformity used in [74] to make possible the usage of larger radii. Avoiding these techniques is one of the main motivations for SRS-LBP because, in the case of textual images specifically, there is important information in the discarded patterns.

4.3 Experimental results

In this section we report on a series of writer identification experiments we performed to evaluate the potential of SRS-LBP and to compare its performance with the SotA.

4.3.1 Datasets

We use a range of publicly available benchmark datasets for our experimental evaluation.

4.3.1.1 ICDAR 2013:

The dataset from the ICDAR 2013 competition consists of 1,000 samples from 250 persons who each contributed two samples in English and two in Greek [61].

4.3.1.2 CVL:

The CVL [51] dataset consists of 1,550 samples from 310 persons who contributed four samples in English and one in German. The samples were acquired in color with different pens. Although 27 of the writers contributed two more texts in English, it is common practice to remove them from consideration.

4.3.1.3 ICHFR 2012:

The ICHFR 2012 dataset consists of 400 samples from 100 subjects contributing two samples in English and two samples in Greek [60]. We use this dataset for baseline performance analysis.

4.3.2 Evaluation protocols

Our approach uses a nearest neighbor classifier, and evaluation is based on leave-one-out cross validation. For each sample represented in feature space, we rank all remaining samples by their distance to that sample. Two important performance measures are the top-n soft criterion, which means having any image of the same class as the query sample in the first n most ranked results, and the top-n *hard* criterion which means having *only* images of the same class as the query sample in first n samples [61, 60].

Comparison with the SotA is complicated by the wide variety of evaluation protocols for writer identification used by international benchmarks and contests. Just indicatively on the three most recent competitions in writer identification, methods had to be, a similarity measurement [60], a trainable classifier [64], and a feature extraction method accompanied by a metric [61].

These protocols were designed not only with performance analysis in mind, but also to accommodate the black-box scenarios in which these contests were performed. In our evaluation we employ two evaluation protocols.

One we refer to as **metric** and is a protocol totally compatible with measurements in [60, 61] (e.g. we only consider pairs of samples, never the entire dataset as a whole).

The other we call **l1out** and is the average performance of a leave-one-out cross-validation in a trainable classifier sense which is compatible with [64]. In practice, the difference between **l1out** and **metric** is that **l1out** allows access to all the samples in the evaluation dataset while **metric** restricts access to each sample alone.

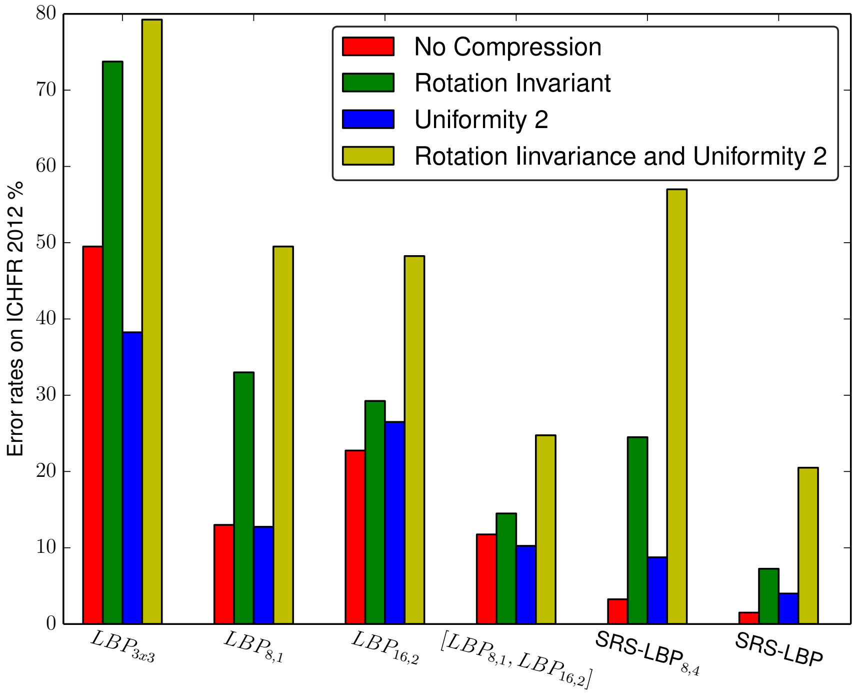


Figure 4.3: Comparison of SRS-LBP with baseline LBP.

For our approach, the difference between the two amounts to whether PCA analysis was done on the evaluation dataset (and thus learning from it) or an independent dataset. In all experiments other than comparison with the SotA, the evaluation protocol used is **l1out**. Since **metric** is a stricter protocol than **l1out**, all SotA performance numbers derived from a protocol that explicitly adheres to metric are marked with an asterisk (*).

4.3.3 Baseline performance analysis

Here we report on a number of baseline experiments we performed to quantify the performance of our approach and to estimate key parameters of our SRS-LBP. All experiments in this section were performed on the ICHFR 2012 dataset.

4.3.3.1 Comparison of SRS-LBP and standard LBP

We consider four standard LBP variants in these experiments: $\text{LBP}_{3\times3}$ [96], $\text{LBP}_{8,1}$ [74], $\text{LBP}_{16,2}$ [74], and a concatenation of LBP $[\text{LBP}_{8,1}, \text{LBP}_{16,2}]$ [74]. For SRS-LBP we consider two variations: a single radius $\text{SRS-LBP}_{8,4}$ and a multi-radius $[\text{SRS-LBP}_{8,1}, ..., \text{SRS-LBP}_{8,12}]$.

In Fig. 4.3 we report the error rates of top-1 accuracy leave-one-cross-validation for standard LBP and proposed SRS-LBP. In addition to the full LBP vocabulary, we also show results for each LBP variant with combinations of vocabulary compression commonly employed for standard LBP. From this figure, we see that SRS-LBP outperform standard LBP in all compression modalities. Note also that SRS-LBP are less sensitive to

Table 4.1: Comparison with the SOA on ICDAR 2013

Method	**Top 1**	**Top 2**	**Top 5**	**Top 10**	**Hard 2**	**Hard 3**
Tebessa-C* [27]	93.4	96.1	97.8	98.0	62.6	37.8
CS-UMD-a* [47]	95.1	97.7	98.6	99.1	19.6	7.1
Super Vector [15]	97.1	NA	NA	NA	42.8	23.8
SRS-LBP$_{8,4}$ **l1out**	97.2	98.2	98.9	99.2	52.9	29.2
SRS-LBP **metric***	96.9	98.5	99.0	99.5	54.5	32.9
SRS-LBP **l1out**	**98.5**	**99.1**	**99.5**	**99.6**	**63.8**	**38.3**

Table 4.2: State-of-the-art on ICDAR 2013 Greek

Method	**Top 1**	**Top 2**	**Top 5**	**Top 10**
Tebessa-C* [27]	93.1	97.0	99.5	99.5
Delta-n Hinge [42]	93.4	NA	NA	98.4
CS-UMD-a* [47]	95.1	97.7	98.6	99.1
Multi Feature* [48]	**99.2**	**99.6**	**99.8**	**99.8**
SRS-LBP$_{8,4}$ **l1out**	96.6	98.0	99.6	99.8
SRS-LBP **metric***	96.6	97.8	98.8	99.4
SRS-LBP **l1out**	98.4	99.2	99.4	**99.8**

compression than standard LBP. Compression is usually applied to standard LBP in order to render vocabulary sizes tractable. Due to their sparse nature, SRS-LBP are already a compact and tractable without resorting to compression.

4.3.4 Comparison with the State-Of-The-Art

In recent years the topic of writer identification has seen a lot of activity. Contests, datasets, as well as several top performing methods have been published. In this section we compare SRS-LBP with the SotA in writer identification.

Table 4.3: State-of-the-art on ICDAR 2013 English

Method	**Top 1**	**Top 2**	**Top 5**	**Top 10**
Tebessa-C* [27]	91.5	95.5	97.5	98.0
Delta-n Hinge [42]	93.4	NA	NA	97.8
CS-UMD-a* [47]	95.2	98.2	98.8	99.2
Multi Feature* [48]	**97.4**	**97.8**	**98.6**	98.8
SRS-LBP$_{8,4}$ **l1out**	95.2	96.4	98.0	98.4
SRS-LBP **metric***	95.6	96.8	98.4	**99.0**
SRS-LBP **l1out**	97.2	97.4	98.2	**99.0**

Table 4.4: State-of-the-art on CVL using only 4+1 samples per writer

Method	**Soft Top 1**	**Soft Top 2**	**Soft Top 5**	**Soft Top 10**	**Hard Top 2**	**Hard Top 3**	**Hard Top 4**
Tebessa-C* [27]	97.6	97.9	98.3	98.5	96.1	94.2	90.0
Multi-feature* [48]	**99.4**	**99.5**	**99.6**	**99.7**	98.3	94.8	82.9
Super Vector [15]	99.2	NA	NA	NA	98.1	95.8	88.7
SRS-LBP$_{8,4}$ **l1out**	99.0	99.2	99.4	99.5	97.7	95.2	86.0
SRS-LBP **metric***	98.6	98.8	98.9	99.1	97.8	94.6	85.3
SRS-LBP **l1out**	**99.4**	99.4	99.5	99.6	**98.6**	**97.0**	**90.1**

Table 4.5: State-of-the-art on IAM with $301 * 2$ samples

Method	**Top 1**	**Top 2**	**Top 5**	**Top 10**
CS-UMD-a* [47]	**96.5**	**97.2**	NA	**97.3**
Delta-n Hinge[42][1]	93.2	NA	NA	97.2
Multi Feature[48]	94.7	95.9	98.1	98.7
SRS-LBP$_{8,4}$ **l1out**	93.2	93.9	95.2	95.2
SRS-LBP **metric***	94.7	95.2	**96.0**	96.3
SRS-LBP **l1out**	94.9	95.2	95.8	96.3

4.3.4.1 Performance on ICDAR 2013

In table 4.1 we compare the performance of SRS-LBP with the SotA on the ICDAR 2013 contest dataset. In Table 4.2 and Table 4.3 we compare our performance with the SotA on the Greek and English portions of the ICDAR 2013 dataset. As of this writing, the Multi Feature method represents the SotA in writer identification [48]. This technique is based on character segmentation and clustering (which is one reason they do not report results on the mixed-language dataset) and multiple features extracted from characters. It is interesting that our approach, which is based on dense extraction of a single feature, performs comparably to this more complicated technique.

4.3.4.2 Performance on IAM

In Table 4.5 we compare the performance of SRS-LBP with the state-of-the-art on the IAM dataset. In contrast to other datasets, SRS-LBPs do not outperform [47] on IAM. A possible reason might be the fact that samples vary between writers and that there are only two samples per class and thus there is only exactly one relevant result per query.

4.3.4.3 Performance on CVL

Finally, in Table 4.4 we compare the performance on SRS-LBP with the SotA on the CVL dataset. On this dataset we have the most complete comparison with SotA approaches for both hard and soft criteria. Our approach performs equivalently to the Multi Feature technique of [48] for top-1 evaluation criterion, and we outperform all others approaches for the hard recognition evaluation criterion across all ranks which is associated with writer retrieval.

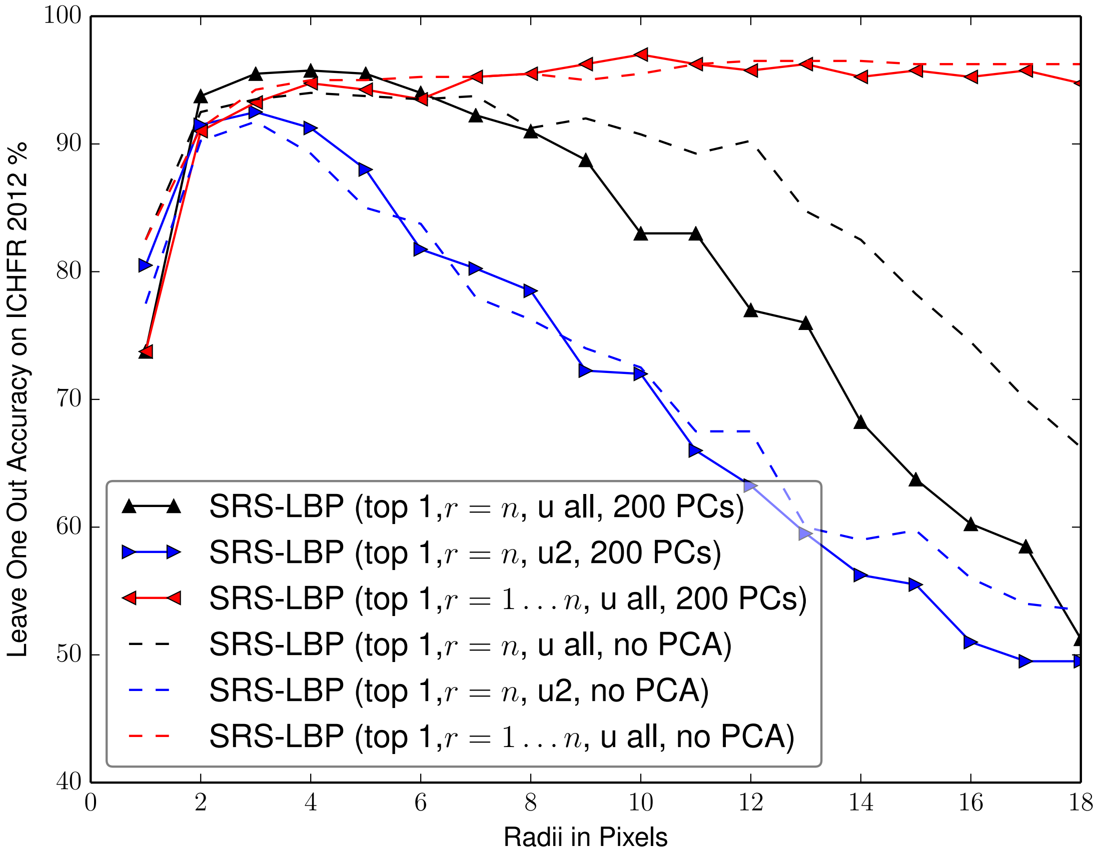


Figure 4.4: Individual and cumulative contribution of each radius to the top-1 accuracy.

4.3.5 Ablation Studies

The proposed SRS-LBP demonstrates to be a powerfull method but in order to get better insights for the performance of the method, several ablation studies were caried out.

4.3.5.1 Radii Contribution

To better understand the contribution of each radius and principal component to the robustness of our descriptor, we decomposed performance as a function of each. In Fig. 4.4 we show the accuracy of each radius independently and cumulatively (i.e. by concatenation). From this, we see that even large radii continue to contribute to improved recognition performance. Note also how, as radii grow beyond 3, the performance of the uniform-compressed features (in blue), drops quite sharply compared to the non-compressed features (in black). This hints that uniformity compression doesn't scale to large radii. Since performance continues increasing until about twelve sparsely sampled radii, we use this configuration of SRS-LBP $= [\text{LBP}_{8,1}, \dots, \text{LBP}_{8,12}]$ for all subsequent experiments.

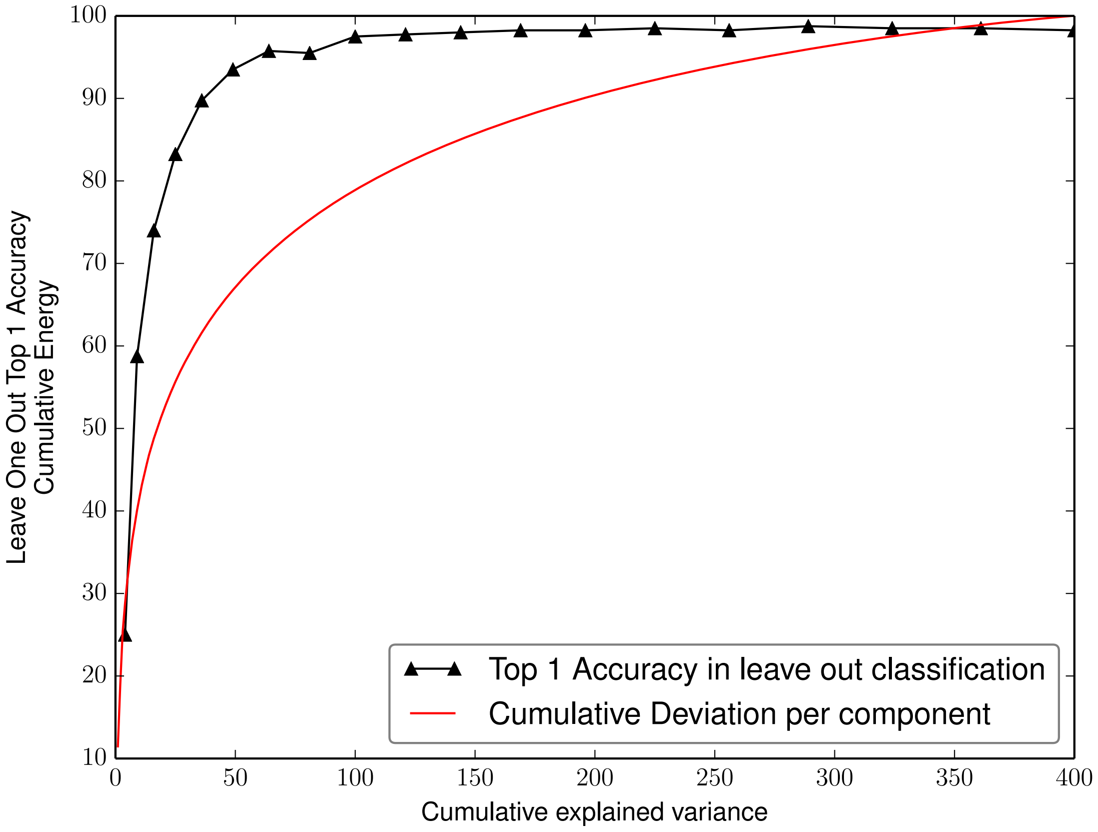


Figure 4.5: The cumulative contribution principal components have in top-1 accuracy.

4.3.5.2 Contribution of principal components

PCA in the pipeline pipeline improves recognition accuracy in all cases. In Fig. 4.5 we plot of top-1 writer identification accuracy as a function of increasing PCs. Although performance quickly begins to saturate, it improves steadily until around 200 principal components. For all subsequent experiments we use 200 principal components, resulting in a very compact image descriptor of only 200 dimensions.

Another way to get some intuitions about the effect of PCA compression on historams of LBP can be seen in Fig. 4.6. What is specificly show is for the 8 components with the highest variance (C1-C8) an image is shown where the columns represent the participation of each specific pattern (0-255) and and the rows represent the the radii. The pattern numbers are each pattern's representation as a bitstring and therefore two consecutive patterns can be quite different. Due to the nature of the PCA and because retrieval happens using PCA, the signs of the weights should not be interpreted to be associated with impact of the pattern frequency, the impact of each pattern at a specific radius can be infered by the absolute value of the component coefficients. What stands out in Fig. 4.6 are the vertical stripes which indicate that a specific pattern is treated consistently across a range of radii. It can be seen that in most cases there are two main radii clusters, small radii between 1 and 4 pixels, and large radii between 4 and 12 pixels. It can be suggested

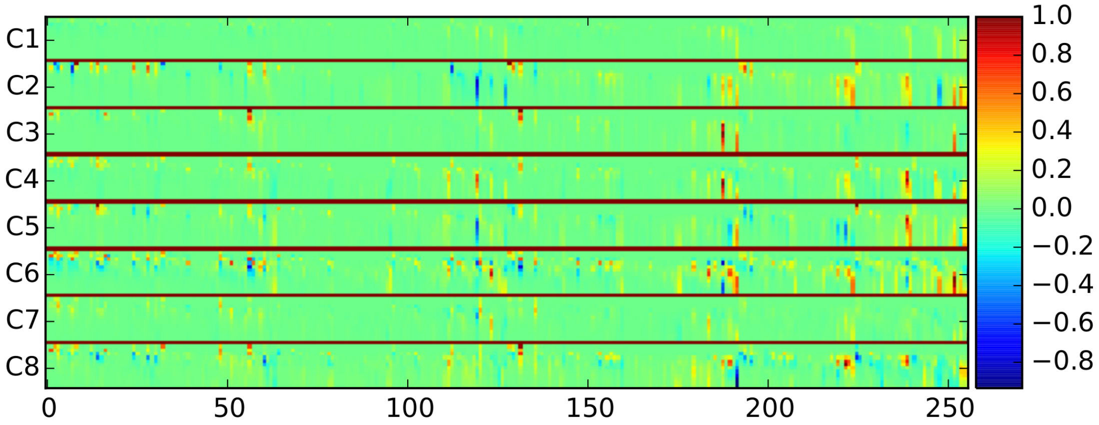


Figure 4.6: Pattern and Radii contribution for 8 most important componets in icdar2013 dataset.

that this dichotomy of the radii is associated with the stroke width.

4.3.5.3 Rotation sensitivity

Tolerance to small rotations can be important for text documents, and in the case of handwritten text the exact orientation of text is probably unknown. To measure sensitivity to rotation, the ICHFR 2012 data-set was rotated from angles $-20°$ to $20°$. For each rotation, all samples in the non-rotated dataset were used as queries against the rotated dataset. In Fig. 4.7 we show the sensitivity to rotation of the proposed SRS-LBP pipeline. An interesting observation in Fig. 4.7 is that while the single radius SRS-LBP performs nearly equivalently to the multi-radius one, it is more sensitive to rotations.

4.3.5.4 Analysis of the results

It is uncontested that the Multi-Feature method at the moment of writing this paper is the top performing method on most relevant datasets. While the proposed SRS-LBP under ***l1out*** did achieve to match to the decimal digit, its performance and in the case of the hard criteria even exceed its performance it should be pointed out that ***metric*** is probably a fairer comparison. Other than the Multi-Feature, SRS-LBP consistently performs better than state-of-the-art methods in all cases other than the IAM dataset. The authors speculate that proposed SRS-LBP features being generic texture descriptors are more sensitive to the the high variation of the stroke width. An other interesting observation is that there appears to be consistency between the ranking some methods get in the soft and hard criteria. Looking at tables 4.4 and 4.1, it can be seen that SRS-LBP and Tebessa-C are consistently the best and second best methods in the hard criteria while methods such as Multi-Features, CS-UMD-a, and Tsinghua, consistently rank better in the soft criteria. We speculate that this happens because methods that perform better

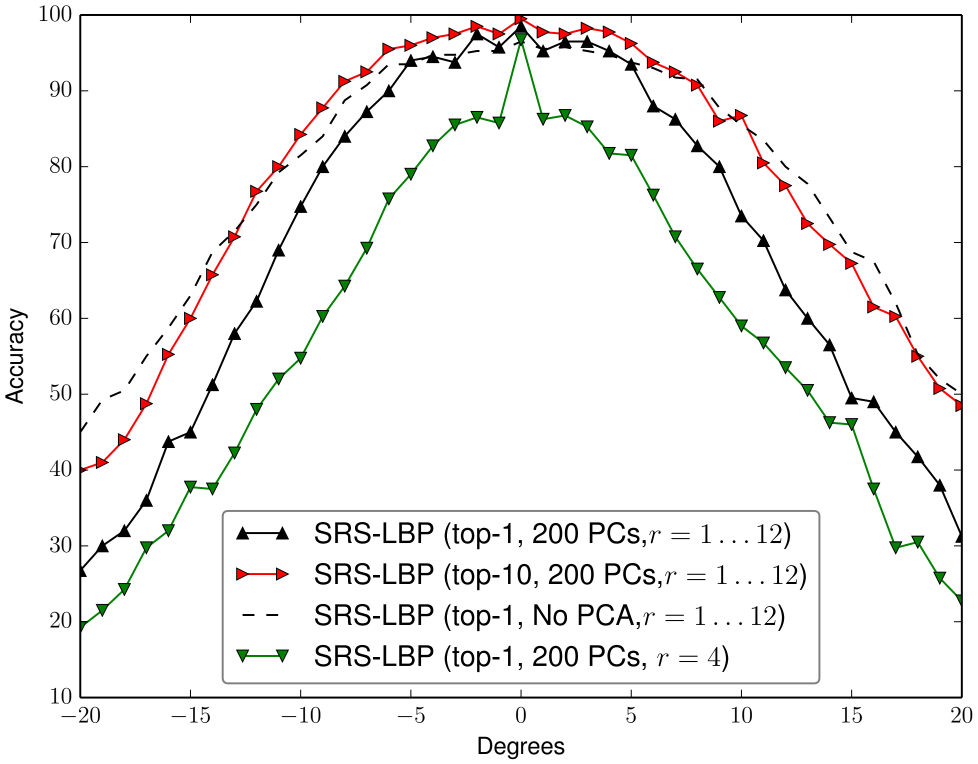


Figure 4.7: Tolerance of SRS-LBP to rotations of the image.

on the hard criteria, have denser clusters for each class.

4.4 In depth analysis of the SRS-LBP

While it has been proven experimanetally that SRS-LBP performs better than generic LBP, in text image classification tasks the reasons of the superior performance might still be ellusive. In order to get a better understanding of the SRS-LBP and why it is representation better suited for text one can look at several perspectives. In the following visualisations, each pattern is represented by a distinct color and thus the relationship between each pattern and the shape of the image can be seen. The 0 pattern *("Not grater than any neighbor")* is marked as black and the 255 pattern *("Grater than all neighbors")* is marked as white. The key features of the SRS-LBP with respect to the standard LBP are:

- No vocabulary compression via uniformity or roation invariance
- Otsu thresholding istead of positive-negative
- Sample multiple radii independently
- Employ very large radii (> 4 pixels) in adition to small-ones

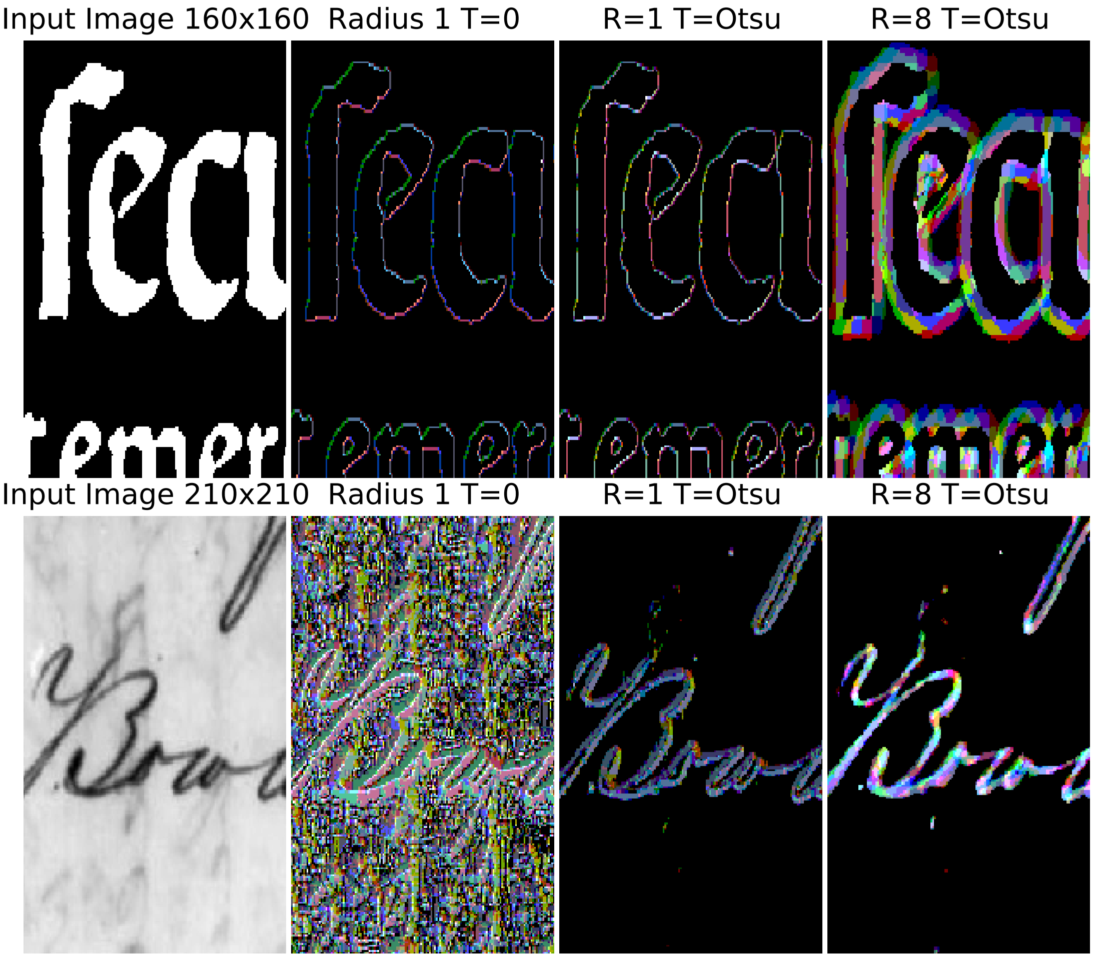


Figure 4.8: SRS-LBP patterns on both printed and handwritten text images.

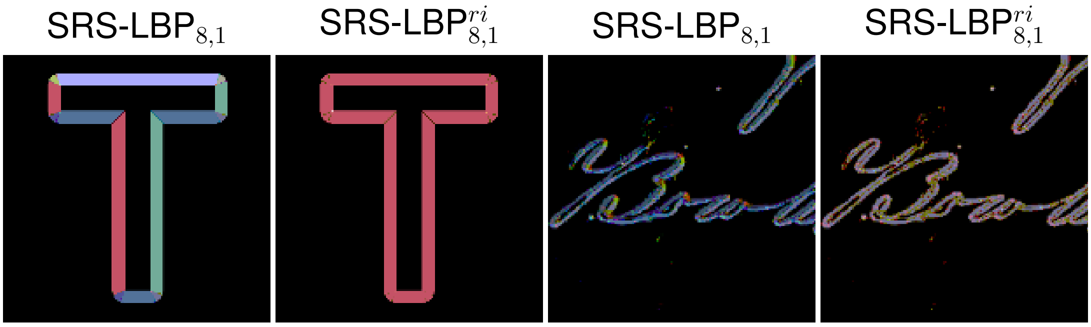


Figure 4.9: Effect of rotation invariance in SRS-LBP patterns

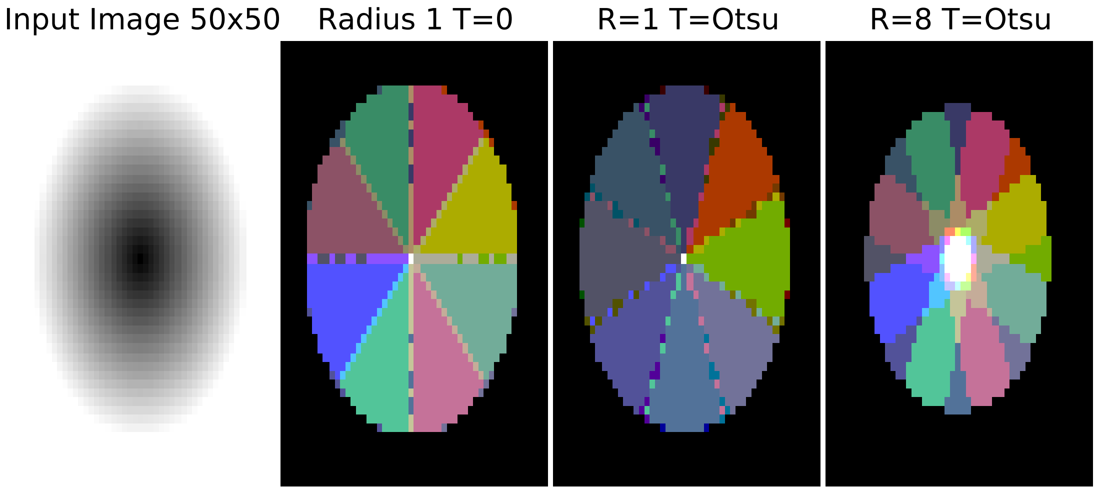


Figure 4.10: Orientation and radius affecting SRS-LBP patterns

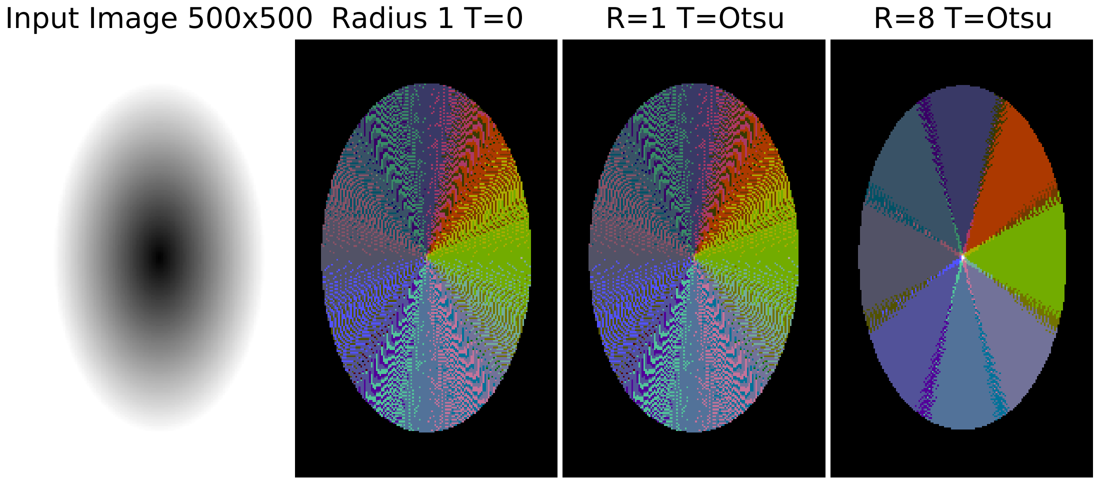


Figure 4.11: Moire patterns in very smooth gradients

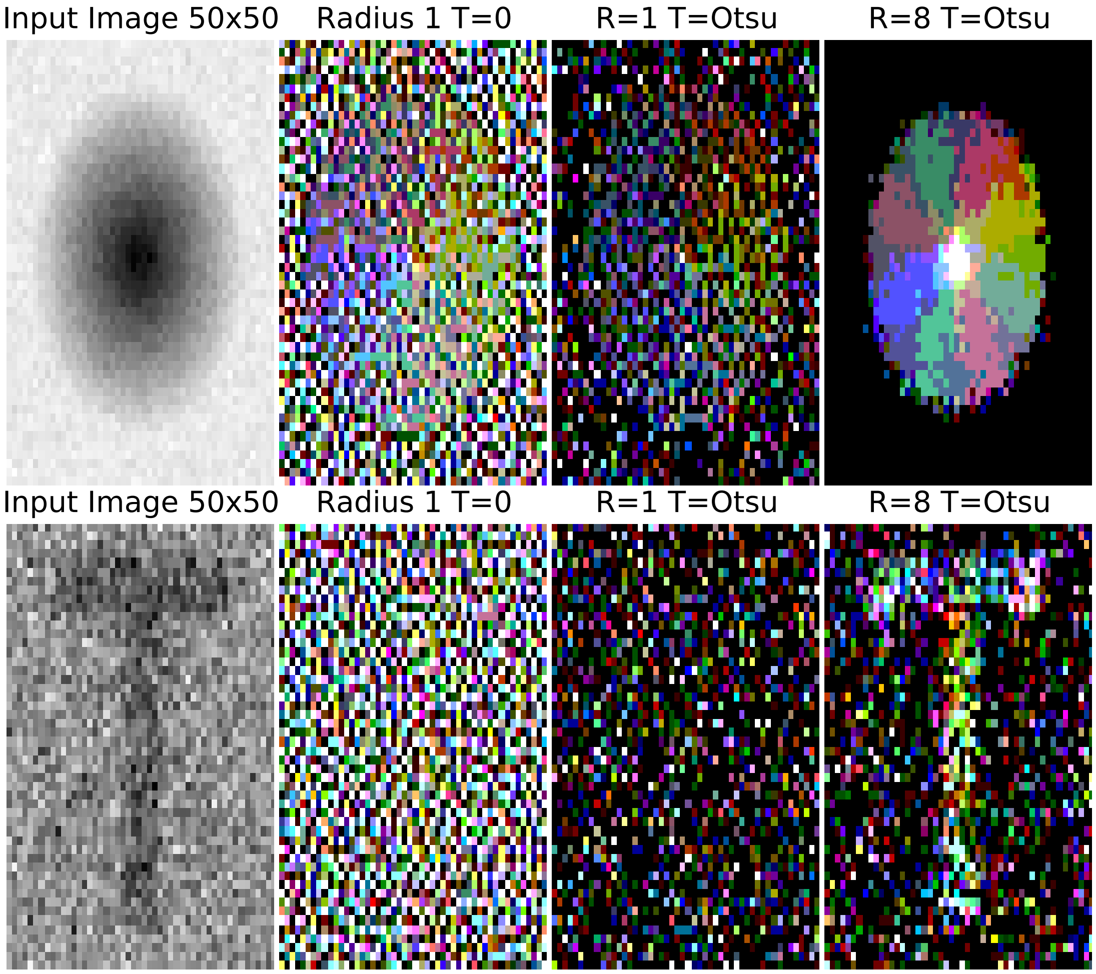


Figure 4.12: Tolerance of SRS-LBP to high frequency noise.

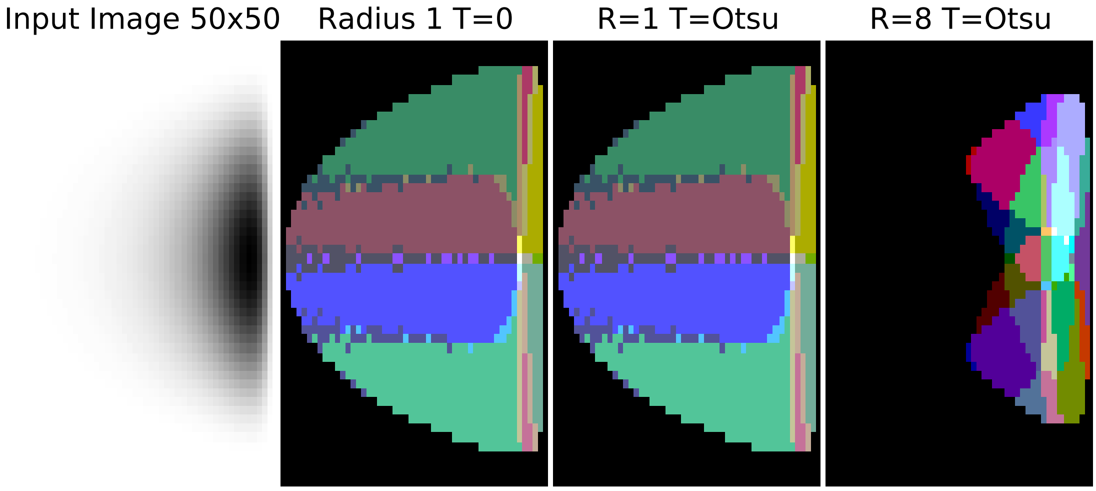


Figure 4.13: Steep vs. mild gradient representation in an image

4.4.1 No Rotation Invariance Compression

In Fig. 4.9 the effect of rotation invariance compression in to the representation can be seen. In effect the orientation of the pattern is the most informative part when the stroke-width of the text is larger than the LBP radius. The detrimental effect the compression has on the descriptive power of the represantion can be seen in its most extreme form in synthetic images but the effect is dominating real-world samples as well. In Fig. 4.10 and Fig. 4.11 the way the angle of the local image gradient produces specific patterns can be seen. It should be pointed out that almost all non-zero patterns occuring are uniform patterns.

4.4.2 The absence of patterns

The LBP descriptors are a bit counter-intutive with respect to the steepness of significant gradients. As can be seen Fig. 4.13 all gradients above the significance threshold produce a uniform pattern regardless of their magnitude. Due to the converse of the gradient theorem[104], along any direction the sum of the gradient must be zero. If along a direction there is a disparity between the slope of ascending and the slope of descending gradients, the smoother direction of the gradient will occupy more pixels and therefore will be overrepresented in gradient. This allows for histogram vlues of uniform patterns to encapsulate information similar to the HOG image descriptor[23]. Otsu thresholding of the LBP in conjuction with large radii will supress regions of very smooth gradients and cast the to the 0-pattern.

Chapter 5

Script Identification

5.1 Introduction

As document analysis systems are evolving, their multi-lingual capabilities are becoming more important. Script identification is a key element in multilingual system pipelines. Other than performance in detecting the script and the language of text in such pipelines, the position this step occupies in the pipeline dictates whether it will assist or be assisted by other steps in the pipeline.

The earlier in the pipeline script and language identification occurs the better. On several modalities of text identification prior knowledge is typically employed in the form of a language model. Even in the most generic case the alphabet from which the text consists must be known or assumed. While assuming that the language is a given, a reasonably condition in controlled conditions, automatic systems associated with automatic and unconstrained acquisition can not make such assumptions. Script detection as a problem is increasing in importance, as well as the attention it gets from researchers.

In this work we address the problem of script or language identification in several modalities such as video-text, scene-text, or handwritten text, and introduce a method consisting of hand-crafted features and a fully connected deep neural network. We demonstrate that k-NN classification over the features obtained from the first layer of the deep neural network equals or outperforms the deep network classification. The principal contributions of this work are: the introduction of a method that uses a deep neural network on top of hand-crafted features for script identification, a method to perform a purely visual identification of language, even for languages sharing the same script, and the use of the activations of the employed neural network as a learned metric in order to generate more adaptable classifiers.

5.2 Script IdentitificationSotA

Script detection has been an open problem for several decades. For the contents of this chapter, script identification refers to identifying the system of writing, the alphabet used in a sample, while language identification refers to identifying the language given a text sample. The above definition produces ambiguities on some cases, yet those two notions from a pattern recognition perspective are very different. Script identification implies focussing on detecting symbols, while language identification implies detecting some specific auxiliary symbols, such as diacritics, and an underlying language model. Several variations of the problem exist depending on aspects such as the granularity of the data samples, the number of scripts out of which the systems classify, and the modality of the textual data, i.e. whether its printed text, handwritten text, scene text etc. For a detailed overview of script identification before 2009, we refer to [38], which provides a thorough taxonomy of methods available up to that time. In 2009 Unnikrishnan and Smith [98] demonstrated that for simple cases of binarized printed text, the problem can be considered solved by a method developed for the Tessaract OCR engine. Zhu et al. [107] have used codebooks generated from printed and handwritten data in order to perform handwritten language identification. Ferrer et al. [29] used the simple $LBP_{3\times3}$ pooled horizontally to perform script identification. More recently Long Short Term Memory (LSTM) networks have been used by Ulhasan et al. [97] for separating characters in multilingual text with a granularity of characters. Mioulet et al [68] also used a bidirectional variant of LSTM networks with a cascade of script detection, OCR and language models, in order to infer the language even when two languages share the same script. The ICDAR2015 Competition on Video Script Identification (CVSI) [87] posed the problem of script identification over superimposed text in videos ; the four best participant methods were all using Convolutional Neural Networks (CNN). Shi et al. [90] have also used a deep CNN to address the problem of script detection in the wild. The CNN approach has the drawback of the need for vast computational resources as well as large amounts of annotated data. Depending on the granularity of samples, i.e. character based, word based, text-line based, and paragraph based, as well as the modality of text, whether it be scene-text, printed documents, or handwritten texts, script identification can be seen as many problems rather than one. From this perspective, the problem addressed in this paper, script identification of a word level granularity on scene-text, is a challenging one that is starting to gain momentum.

5.3 Method

The proposed method consists of a preprocessing step, followed by LBP feature extraction, and training an Artificial Neural Network (ANN) on these features. The intermediary

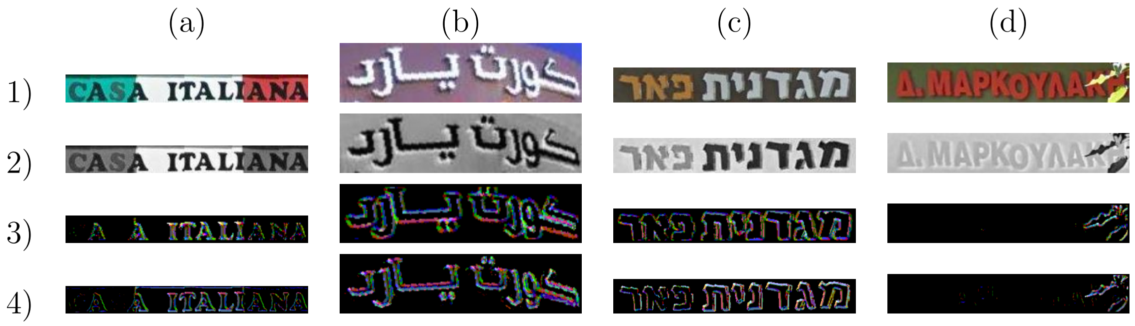


Figure 5.1: Preprocessing and SRS-LBP transform for radius of 3. Data taken from SIW-10 dataset [88]. 1) Original image, 2)Preprocessed image, 3)SRS-LBP$_{8,3}$ of input images and 4)SRS-LBP$_{8,3}$ of preprocessed images.

layers of the ANN are then used as a generative model to perform classification.

5.3.1 Preprocessing

Before passing images to the LBP transform, each image is preprocessed independently. Since the LBP transform is applied on a single channel image, instead of luminance, the principal component of all pixel colors was chosen in order to enhance the perceptual differences that are not attributed to luminance. In order to have a consistent LBP encoding between images with a light foreground on a dark background and images with a dark foreground on a light background, whenever the central band is darker than the image average, the image is flipped. The assumption is that more foreground pixels will exist in the central band between 25% and 75% of the image width. In fig. 5.1 rows 1) and 2) demonstrate some examples of the preprocessing. When global pooling and using local structure features such as the LBP or Histogram of Oriented Gradients (HOG), inverting the image has an effect equivalent to flipping it across both axes. This means that making all samples have light background on dark foreground is as important as enforcing all samples to be properly oriented. In Column (b) of Fig. 5.1 the effect the flipping has on the LBP transform can be seen.

5.3.2 Local Binary Patterns

For feature extraction the SRS-LBP variant of LBP histogram features is employed. Briefly, the SRS-LBP embeds a clustering of the center-neighbourhood differences using Otsu's [75] method; it also uses a dis-joined approach to obtain a multi-radius feature representation with linear complexity. LBP have several advantages for script identification: they exploit the bi-level nature textual images have, they are very fast to compute, and they are pooled over regions which makes them segmentation-free and an inherently global descriptor of an image region. In [29] Ferrer et al. extracted LBP features from text-lines by concatenating

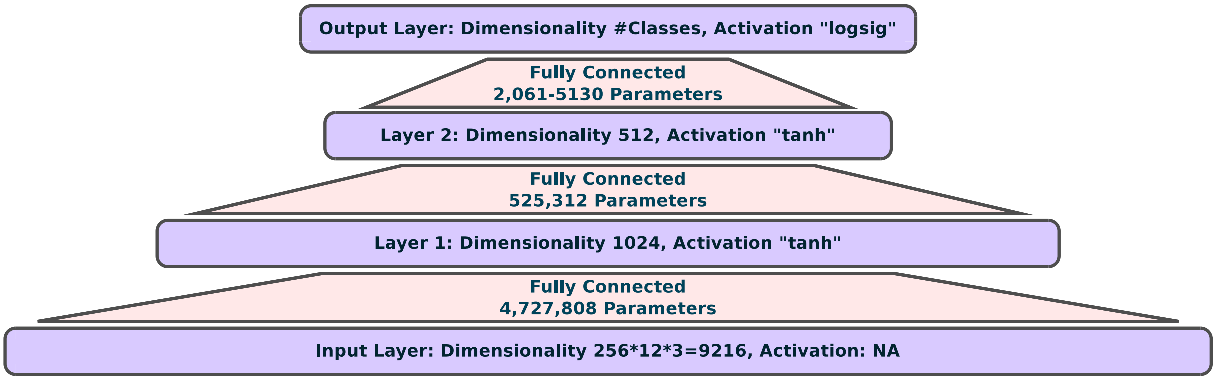

Output Layer: Dimensionality #Classes, Activation "logsig"
Fully Connected
2,061-5130 Parameters
Layer 2: Dimensionality 512, Activation "tanh"
Fully Connected
525,312 Parameters
Layer 1: Dimensionality 1024, Activation "tanh"
Fully Connected
4,727,808 Parameters
Input Layer: Dimensionality 256*12*3=9216, Activation: NA

Figure 5.2: Architecture of the Proposed Neural Network

histograms of 4 horizontal stripes. In [88] Shi et al. used a deep convolutional network that employs horizontal pooling to discard spatial information along the horizontal direction.

In the same respect, and assuming images are either cropped words or cropped lines, the SRS-LBP were extracted for 3 regions in the images: the upper half of the image, the central half of the image, and the lower half of the image. The dimensionality of the extracted features-set is the product of the histogram size, the different radii and the pooling zones: $2^8 \times 12 \times 3 = 9216$. In Fig. 5.1, in column (d) a rare example where the SRS-LBP is fooled can be seen; this happens because the SRS-LBP assumes that the most significant contrast in an image with text will be the contrast related to foreground-background transitions.

5.3.3 Classification

The remaining pipeline of the SRS-LBP is an unsupervised learning approach, aimed at totally different class and samples per class cardinalities. A deep Multi Layer Perceptron (MLP) is used as a classifier of the feature representation to a given and limited set of languages. The network consists of 3 fully connected layers plus the input layer. The first layer maps the 9,216 features to a dimensionality of 1,024, the second layer maps the data from 1,024 to 512 dimensions and the third layer maps the data to as many neurons as the number of classes. The output layer can be interpreted as the probability of the presented data belonging to each class. The activations for the layers are respectively *tanh*, *tanh*, and the logistic function. In Fig. 5.1 a visual representation of the architecture can be seen. It should be pointed out that the number of parameters of the network varies depending on the feature vectors dimensionality as well as the number of classes in each dataset used. In the case of classifying word images to 10 classes, the model has 9,968,138 parameters.

5.3.4 MLP as Metric Learning

While the discriminative deep MLP performs well, it is quite restricted by the need of computational resources for training. More than that, deep networks require datasets of substantial size and with all classes represented in a balanced way. The other alternative is to use metric learning techniques, which can have some drawbacks. Metric learning methods tend to have quadratic and even cubical complexities with respect to feature dimensionality, therefore an intermediary dimensionality reduction technique must also be used. The idea of using neural networks dedicated to metric learning is best exemplified by the Siamese network architecture [12]. While Siamese networks have all the benefits of metric learning in typical classification tasks, such as digit image classification, the results are lower than the state-of-the-art classifiers [57]. On the other hand, intermediary activations of CNN are being used as generic feature extractors which are then classified with off the shelf classifiers such as Support Vector Machines (SVM) [86]. The work presented in this paper is greatly influenced by the principal idea in [86] of using intermediary activations of deep CNN as generic features for standard classifiers in tasks other than what the original CNN was trained for. Established CNN architectures do not directly preserve the aspect ratio of samples. In the case of word samples, this means that the same letters could have a different representation if they appeared in words of different size. There is no straight forward solution to this problem, i.e. the winning method of the CVSI competition addressed this problem by performing a sliding window of a fixed aspect ratio in each word and selecting the window with highest activation [87]. The drawback in such an approach is that all information outside the maximal activation window is ignored. The authors propose the use of hand-crafted features that can address the aspect ratio problem. Specifically the authors use the SRS-LBP histograms as inputs to a deep MLP since the pooling mechanism of the LBP histograms preserves perfectly the aspect ratio. Instead of using specialised architectures CNN obtained features, the authors of this paper propose to use the activations of the early layers in the proposed MLP as learned metrics on the hand-crafted SRS-LBP features. Building on the idea of [86] the activations of the early layers in the MLP are used as input to a Nearest Neighbour classifier. Depending on the dataset, lower levels of the proposed MLP can reach in performance and even exceed its output layer. At the same time, networks used with this strategy are not limited to classes available during training. In Fig. 5.3 the error rates of all layers during training of the proposed MLP on the CVSI2015 dataset can be seen.

5.4 Experiments

In the experimental section we present experiments on script identification and language identification that demonstrate the potential of the proposed approach.

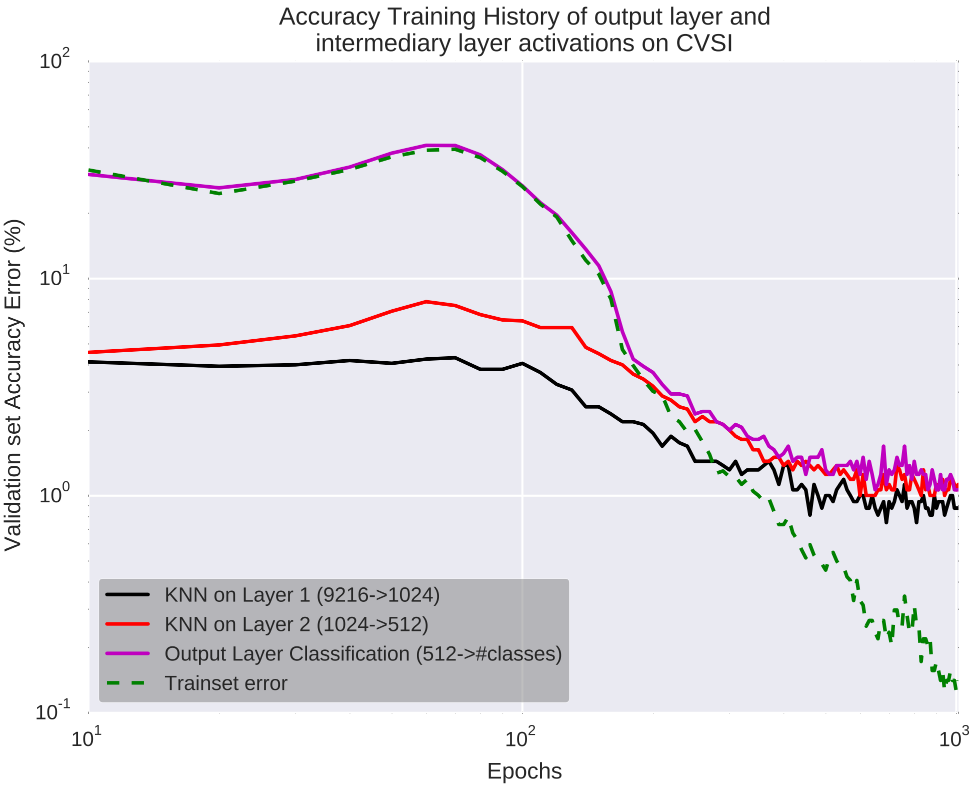


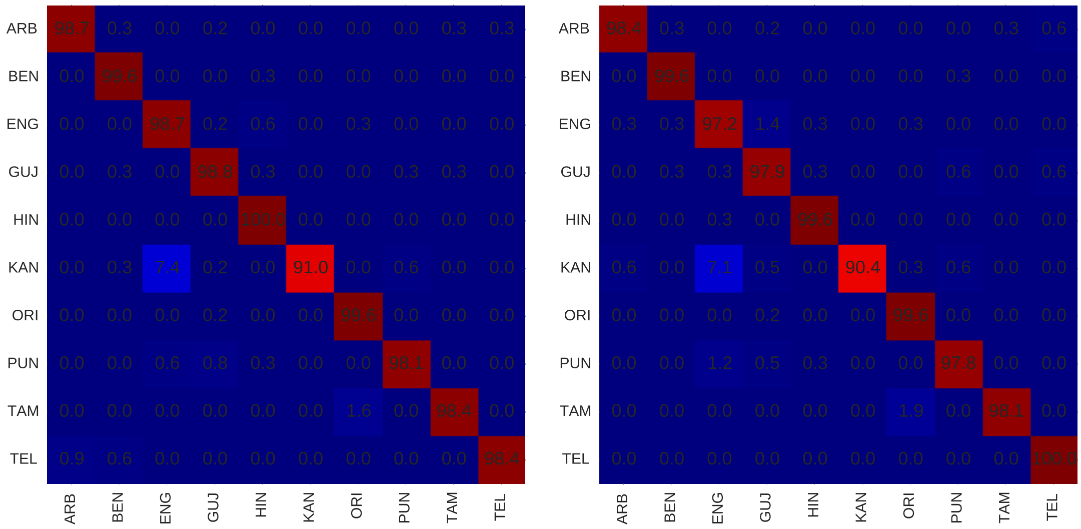
Figure 5.3: Training of the MLP.

Table 5.1: Accuracy % on the CVSI Video-text Dataset

Language	C-DAC	HUST	CVC	Google	CUK	Layer 1, 1NN	Layer 2, 1NN	Layer 3
Arabic	97.69	**100.0**	99.67	**100.0**	89.44	98.7	98.4	98.4
Bengali	91.61	95.81	92.58	99.35	68.71	**99.6**	99.3	**99.6**
English	68.33	93.55	88.86	97.95	65.69	**98.7**	98.4	97.2
Gujrathi	88.99	97.55	98.17	98.17	73.39	**98.8**	95.3	97.2
Hindi	71.47	96.31	96.01	99.08	61.66	**100.0**	99.6	99.6
Kannada	68.47	92.68	97.13	**97.77**	71.66	91.0	87.5	90.4
Oriya	88.04	98.47	98.16	98.47	79.14	**99.6**	**99.6**	**99.6**
Punjabi	90.51	97.15	96.52	**99.38**	82.55	98.1	97.8	97.8
Tamil	91.90	97.82	**99.69**	99.37	82.55	98.4	98.1	98.1
Telugu	91.33	97.83	93.80	99.69	57.89	98.4	98.1	**100.0**
Average	84.66	96.69	96.00	**98.91**	74.06	98.18	97.26	97.9

5.4.1 Video-text Script Identification

The principal experiment to demonstrate near state-of-the-art performance is by comparing to the methods participating in the CVSI 2015 Video Script Identification [87]. The dataset contains of 10 languages used commonly in India: Arabic, Bengali, English, Gujrathi, Hindi, Kannada, Oriya, Punjabi, Tamil, Telugu. The dataset consists of cropped images containing a single word each. Most words appear to come from overlayed text, but there are also images that appear to be scene-text. The dataset comes partitioned to a train-set, a test-set, a validation set, and a small sample-set. For the experiments the test-set was isolated and used only for testing, the remaining data were mixed and partitioned randomly for training during the tuning of the proposed deep MLP architecture. While the competition defines four tasks that are related to different use cases specific to India, such



(a) (b)

Figure 5.4: Confusion matrices for the CVSI dataset. Accuracy of the Nearest Neighbor for the activations of the first layer (a) and the third layer (b)

Table 5.2: Cross-domain use of deep MLP layers

MLP Train Dataset	Retrieval Dataset	k-NN on layer 1	k-NN on layer 2
SIW-10	CVSI	94.8%	76.1%
CVSI	CVSI	98.2%	97.3%
CVSI	SIW-10	66.4%	47.7%
SIW-10	SIW-10	84.5%	84.6%

as discriminating between languages occurring on the same regions, in our experiments we only address Task-4, classifying all 10 scripts, as it is the most generic task. In table 5.1 the performance per script of every participant to the competition can be seen along with the accuracy achieved by each layer. All layers of the proposed deep MLP rank on average second to the method submitted by Google. While the method of Google, the state-of-the-art, obtains 98.9% using a CNN, k-NN on the first layer of the deep MLP obtains 98.2% while layer 2 obtains 97.3% and the output layer obtains 97.9%. In Fig. 5.4 the confusion-matrices between languages for k-NN on the first layer, as well as the output layer can be seen. What stands out is the non-symmetric misclassification of 7% English samples as Kannada; all other confusions could be considered negligible. It can also be observed that layer 1 and layer 2 demonstrate some consistency.

5.4.2 Scene-text Script Identification

While the method was developed for video-text script identification, experiments on how it would perform on script detection in the wild were performed. We used the SIW dataset. Two variants of the dataset are publicly available. The SIW-10 [88] contains cropped

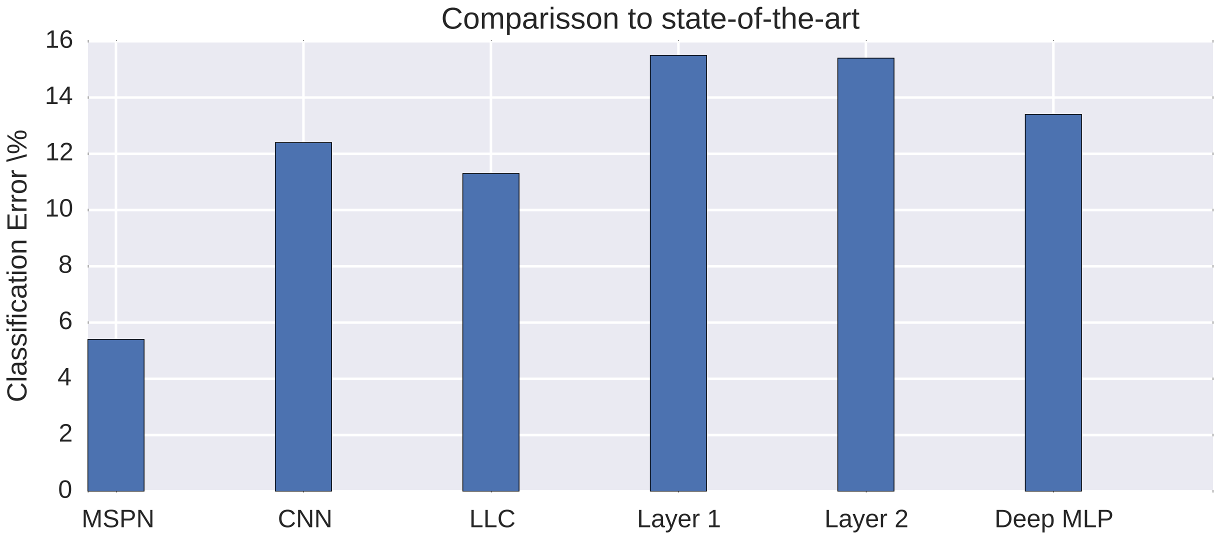


Figure 5.5: Comparison to state-of-the-art on SIW-10. Error rates of the state-of-the-art CNN approach (MSPN), CNN baseline methods (CNN and LCC), intermediary layers as metric learning fed in to a Nearest Neighbour Classifier (Layer 1,Layer 2), and the proposed (Deep MLP).

word images in Arabic, Chinese, English, Greek, Hebrew, Japanese, Korean, Russian, Thai, and Tibetan. The SIW-13 [89] adds Cambodian, Kannada, and Mongolian to the languages of SIW. SIW-10 is partitioned in a train-set of 8,045 samples and test-set of 5,000 images while SIW-13 9,791 and 6,500 respectively. Brief experimentation suggested that the partition of SIW-13 is not compatible with SIW-10, as test samples from SIW-13 appear to be in the SIW-10 train-set. At the time of writing this paper the state-of-the art performance on the particular dataset is 94.6% and is achieved by the Multi-stage Spatially-sensitive Pooling Network (MSPN) method introduced in [88]. Briefly, MSPN is a CNN developed specifically for script identification which introduces among other things a horizontal pooling layer. In Fig. 5.5 a comparison of the proposed deep MLP with state-of-the-art methods for script detection in the wild can be seen. The proposed method achieved an error rate of 13.4% in classification accuracy which is significantly worst than the state-of-the-art 5.6%. Yet, this experiment allows an analysis in to the workings of the proposed deep MLP and the benefits of using k-NN on the intermediary activations. The initial SIW-10 dataset was augmented by the three new languages of SIW-13. An MLP trained on the SIW-10 was used to perform k-NN on the augmented dataset.

In Fig. 5.6 a confusion matrix of employing the first layer of the MLP with k-NN on the SIW-10 dataset augmented by the 3 languages of SIW-13. The overall accuracy is 83.7%, while for the initial 10 scripts it is 84.5%. While the second layer performed better than the first on the dataset for which the model was trained, 84.6%, it proved to be less generic than the first layer and got 77.3% when applied on all 13 classes.

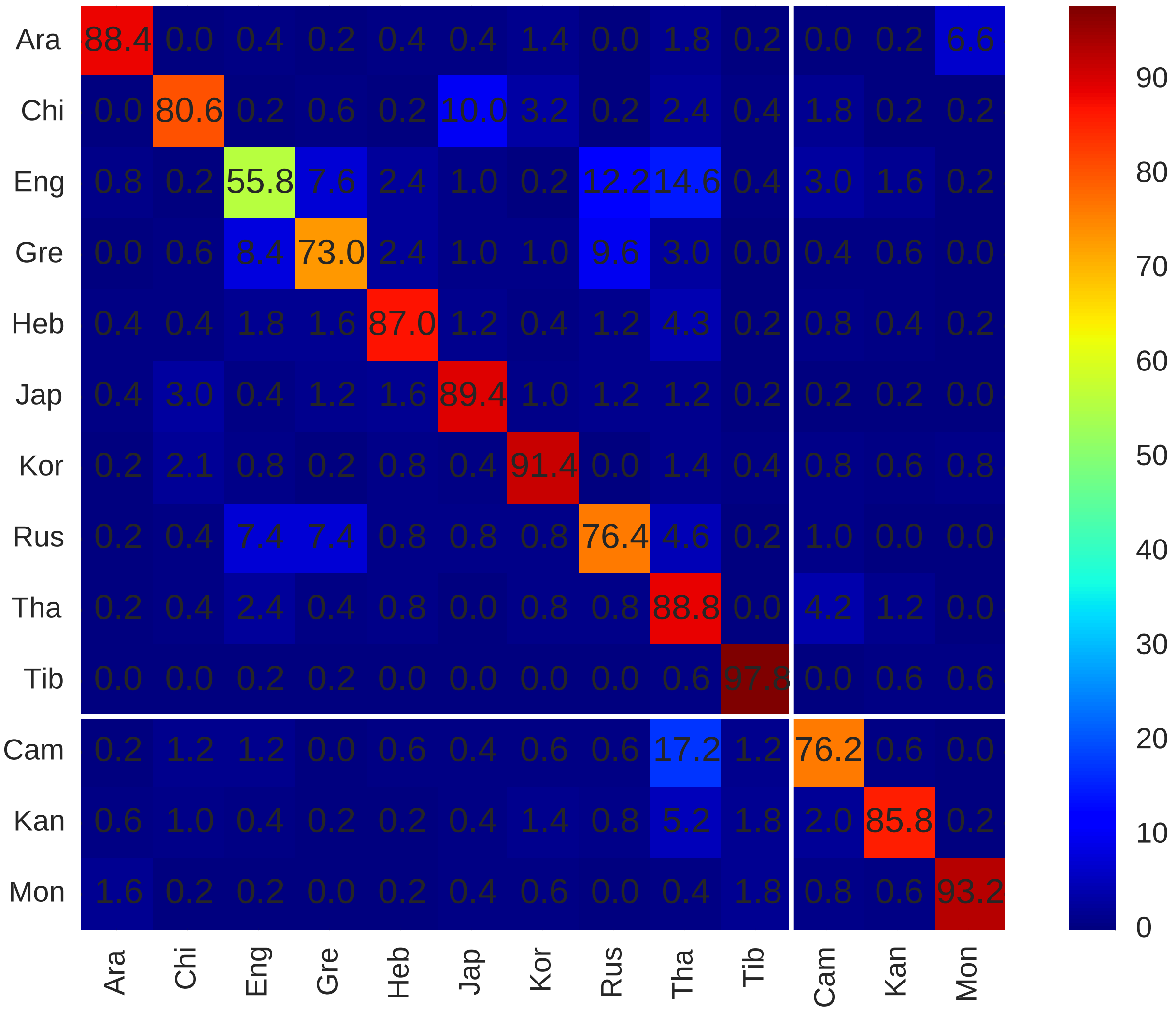


Figure 5.6: Confusion matrix of k-NN on 13 scripts form the SIW datasets using the first layer of a deep MLP trained on 10 of them.

The fact that the classes used to train the model have an average accuracy of 82.9% and the unseen classes have an average accuracy of 85.1% demonstrates the overall genericness of the first layer. As opposed to the CVSI experiments, when training on SIW data, consistently the second layer seemed to outperform the first layer after some epochs.

In Fig. 5.7 the training of the deep MLP can be seen and we can observe that the second layer converges towards the output layer while the first layer appears to be more independent. The extent to which layer 2 is domain specific while layer 1 is much more domain independent can be seen in table 5.2, where layer 1 increases error rates when changing domains by 2.2 and 2.8 times, while layer 2 increases error rates by 3.4 and 8.9 times respectively.

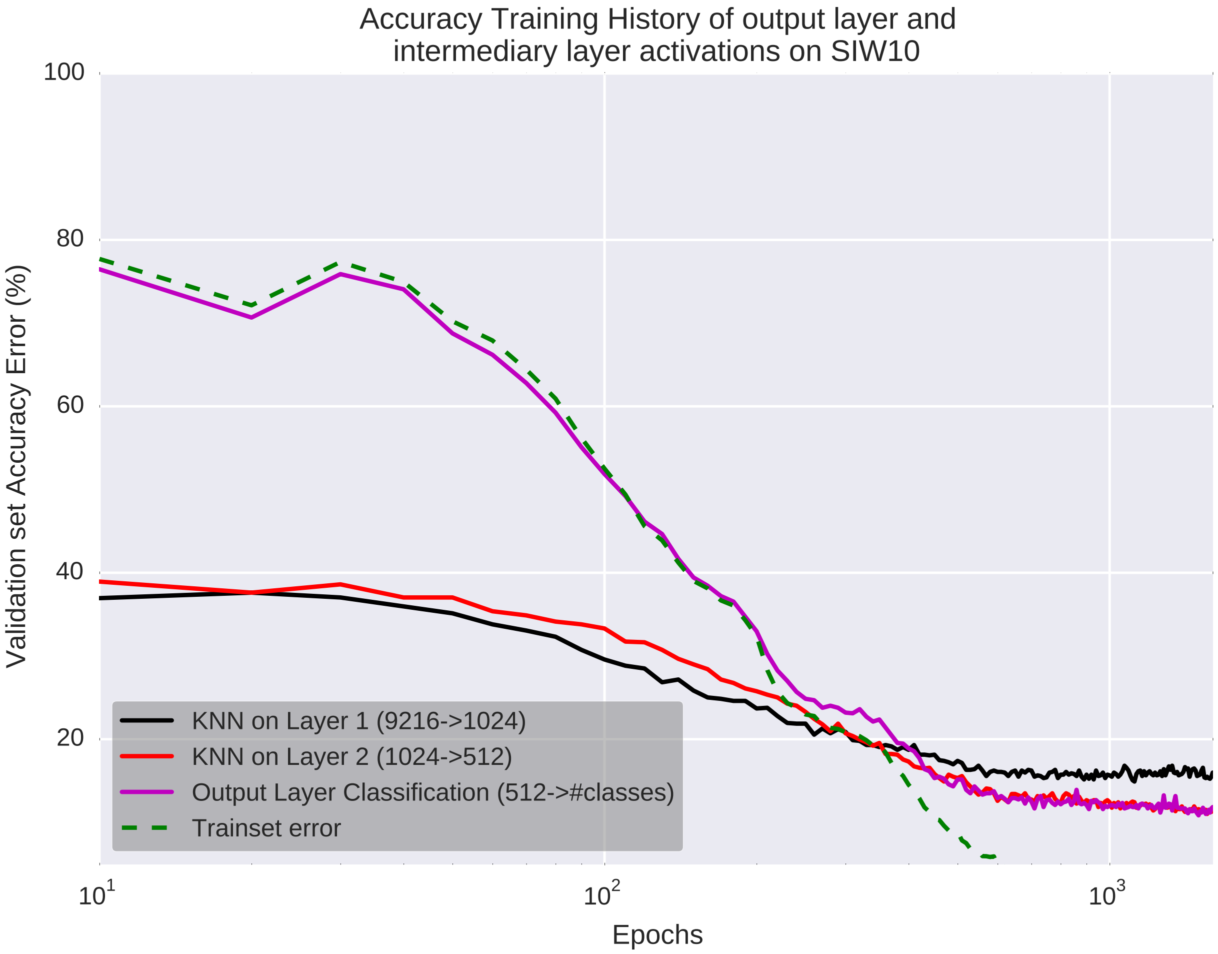
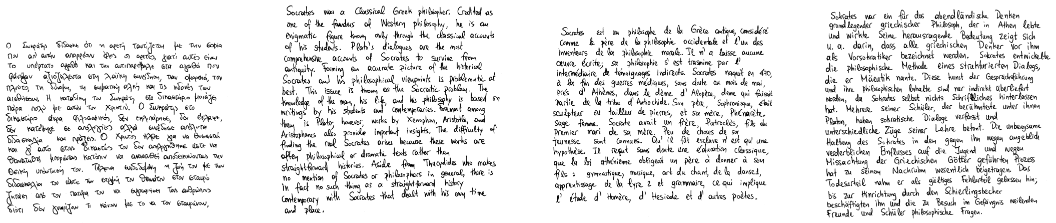


Figure 5.7: Training of the proposed deep MLP on the SIW-10 data.

Table 5.3: Visual Language Identification Accuracy

Method	Accuracy
Random Classifier	25.0%
SRS-LBP learning free pipeline	50.96%
SVM + SRS-LBP features	91.18%
Deep MLP + SRS-LBP features	**92.78%**



(a) (b) (c) (d)

Figure 5.8: Samples from handwritten language identification. Samples of the same text and writer in Greek (a), English (b), French (c), and German (d).

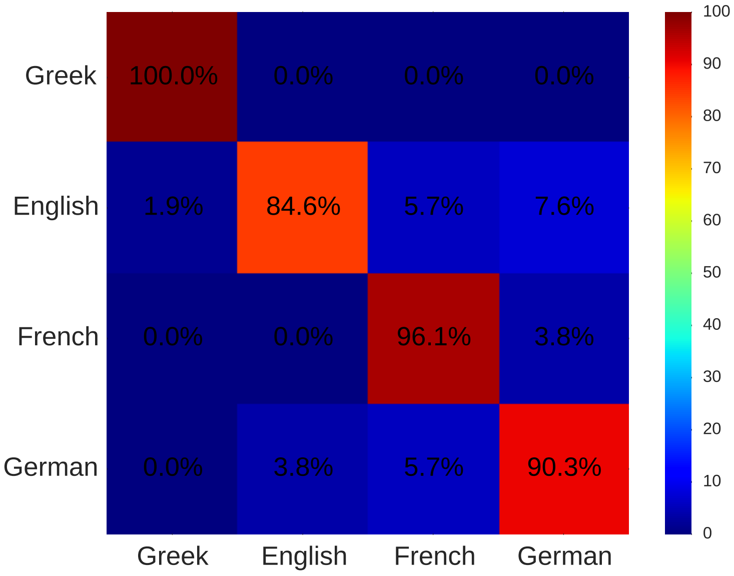


Figure 5.9: Confusion Matrix on Visual Language Detection with an ANN 26-fold cross-validation

5.4.3 Visual Identification of Handwritten Language

The boundary between script and language identification is hard to define, as can be best exemplified in Latin derived languages. Visual language identification could also be perceived as fine-grained script classification. Yet distinguishing between such scripts before identification is required if one is to use language models for identification. In the case of handwriting identification, it becomes even more important, since identification frequently relies in word-spotting, which by definition needs a lexicon. In order to address this problem the ICDAR 2011 writer identification dataset was used[59] to estimate the language classification. This dataset consists of two paragraph-long texts translated to four languages: Greek, English, French, and German. In Fig. **??** the same text written by the same writer in all four languages can be seen. Twenty six writers wrote all these samples which were then digitized and binarized. State-of-the-art methods report performances of over 95% accuracy in writer identification. As the dataset has never been used in the language identification context and visual handwritten-text language identification is a new problem to the authors knowledge, there is no state-of-the-art method. In order to make writer identification irrelevant, a 26-fold cross validation scheme was employed. All 8 samples contributed from every writer were used as testing samples, while all other samples were used for training.

In table 5.3 the performance of the proposed deep MLP along with baselines is presented. The dataset is totally balanced and has 4 classes, so an unbiased random classifier would be performing with 25%. The SRS-LBP unsupervised learning from [72] performs poorly, although significantly better than the random classifier. The same pipeline when applied on the same dataset for writer identification obtains 98.1%. This could be interpreted as an indication of how harder the Handwritten Visual Language Identification problem is compared to writer identification, at least for LBP features. The proposed method Deep MLP is exactly the same as the one described and used for CVSI, but instead of three pooling zones only global pooling is employed as the image has more than one text-line. Deep MLP achieves top performance 92.78%, although an SVM applied on the same features performs nearly as well. In Fig. 5.9 the confusion matrix of visual language classification can be seen. As one would expect, Greek is separated from the other three perfectly while English, French, and German have some confusions. It should be pointed out the dataset was acquired in Greece and all subjects would have Greek as their primary language and this might be helping distinguish it from the other three languages.

Chapter 6

Conclusion

In the course of this thesis we hypothecised that texture analysis can lead to an understanding of textual images. The methods presented demonstrate the versatillity of the aproach. The coherence of the methods employed in the braod scope of analysis hints that this aproach encapsulates an underlying common aspect of text.

Bibliography

[1] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006.

[2] Timo Ahonen, Jiří Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Scandinavian conference on image analysis*, pages 61–70. Springer, 2009.

[3] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Efficient exemplar word spotting. In *Bmvc*, page 3, 2012.

[4] J. Almazn, A. Gordo, A. Forns, and E. Valveny. Handwritten word spotting with corrected attributes. In *2013 IEEE International Conference on Computer Vision*, pages 1017–1024, Dec 2013. doi: 10.1109/ICCV.2013.130.

[5] Marios Anthimopoulos, Basilios Gatos, and Ioannis Pratikakis. A hybrid system for text detection in video frames. In *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*, pages 286–292. IEEE, 2008.

[6] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379, 2010.

[7] A Balasubramanian, Million Meshesha, and CV Jawahar. Retrieval from document image collections. In *Document Analysis Systems*, volume 3872, pages 1–12. Springer, 2006.

[8] Diego Bertolini, Luiz S Oliveira, E Justino, and Robert Sabourin. Texture-based descriptors for writer identification and verification. *Expert Systems with Applications*, 40(6):2069–2080, 2013.

[9] Anurag Bhardwaj, Damien Jose, and Venu Govindaraju. Script independent word spotting in multilingual documents. In *Proceedings of the 2nd workshop on Cross Lingual Information Access (CLIA) Addressing the Information Need of Multilingual Societies*, 2008.

[10] Francesco Bianconi and Antonio Fernández. On the occurrence probability of local binary patterns: a theoretical study. *Journal of Mathematical Imaging and Vision*, 40(3):259–268, 2011.

[11] Robert Bringhurst. *The elements of typographic style*. CRC Studio, 1996.

[12] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[13] Marius Bulacu and Lambert Schomaker. Text-independent writer identification and verification using textural and allographic features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):701–717, 2007.

[14] Vincent Christlein and Andreas Maier. Encoding cnn activations for writer recognition. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 169–174. IEEE, 2018.

[15] Vincent Christlein, David Bernecker, Florian Hoenig, and Elli Angelopoulou. Writer identification and verification using gmm supervectors. In *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2014.

[16] Vincent Christlein, David Bernecker, Andreas Maier, and Elli Angelopoulou. Offline writer identification using convolutional neural network activation features. In *German Conference on Pattern Recognition*, pages 540–552. Springer, 2015.

[17] Vincent Christlein, Martin Gropp, Stefan Fiel, and Andreas Maier. Unsupervised feature learning for writer identification and writer retrieval. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 991–997. IEEE, 2017.

[18] Vincent Christlein, Anguelos Nicolaou, Mathias Seuret, Dominique Stutzmann, and Andreas Maier. Icdar 2019 competition on image retrieval for historical handwritten documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1505–1509. IEEE, 2019.

[19] Vincent Christlein, Lukas Spranger, Mathias Seuret, Anguelos Nicolaou, Pavel Král, and Andreas Maier. Deep generalized max pooling. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1090–1096. IEEE, 2019.

[20] Florence Cloppet, Veronique Eglin, Marlene Helias-Baron, Cuong Kieu, Nicole Vincent, and Dominique Stutzmann. Icdar2017 competition on the classification of medieval handwritings in latin script. In *2017 14th IAPR International Conference*

on Document Analysis and Recognition (ICDAR), volume 1, pages 1371–1376. IEEE, 2017.

[21] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[22] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[23] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[24] Kresimir Delac and Mislav Grgic. A survey of biometric recognition methods. In *Proceedings. Elmar-2004. 46th International Symposium on Electronics in Marine*, pages 184–193. IEEE, 2004.

[25] Sounak Dey, Anguelos Nicolaou, Josep Llados, and Umapada Pal. Local binary pattern for word spotting in handwritten historical document. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 574–583. Springer, Cham, 2016.

[26] Persi Diaconis and Ronald L Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 262–268, 1977.

[27] Chawki Djeddi, Labiba Souici-Meslati, and Abdellatif Ennaji. Writer recognition on arabic handwritten documents. In *Image and Signal Processing*, pages 493–501. Springer, 2012.

[28] David Fernández-Mota, Jon Almazán, Núria Cirera, Alicia Fornés, and Josep Lladós. Bh2m: The barcelona historical, handwritten marriages database. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 256–261. IEEE, 2014.

[29] Miguel A Ferrer, Aythami Morales, and Umapada Pal. Lbp based line-wise script identification. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 369–373. IEEE, 2013.

[30] Stefan Fiel and Robert Sablatnig. Writer retrieval and writer identification using local features. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 145–149. IEEE, 2012.

[31] Andreas Fischer, Andreas Keller, Volkmar Frinken, and Horst Bunke. Lexicon-free handwritten word spotting using character hmms. *Pattern Recognition Letters*, 33 (7):934–942, 2012.

[32] Alicia Fornés, Josep Lladós, Gemma Sánchez, and Horst Bunke. Writer identification in old handwritten music scores. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, pages 347–353. IEEE, 2008.

[33] Alicia Fornés, Josep Lladós, Gemma Sánchez, Xavier Otazu, and Horst Bunke. A combination of features for symbol-independent writer identification in old music scores. *International Journal on Document Analysis and Recognition (IJDAR)*, 13 (4):243–259, 2010.

[34] Volkmar Frinken, Andreas Fischer, R Manmatha, and Horst Bunke. A novel word spotting method based on recurrent neural networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):211–224, 2012.

[35] A. Garz, A. Fischer, R. Sablatnig, and H. Bunke. Binarization-free text line segmentation for historical documents based on interest point clustering. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 95–99, March 2012. doi: 10.1109/DAS.2012.23.

[36] Angelika Garz, Marcel Würsch, and Rolf Ingold. Training-and segmentation-free intuitive writer identification with task-adapted interest points. In *17th Biennial Conference of the International Graphonomics Society*, 2015.

[37] Basilios Gatos and Ioannis Pratikakis. Segmentation-free word spotting in historical printed documents. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 271–275. IEEE, 2009.

[38] Debashis Ghosh, Tulika Dube, and Adamane Shivaprasad. Script recognitiona review. *IEEE Transactions on pattern analysis and machine intelligence*, 32(12):2142–2161, 2010.

[39] Suman Ghosh and Ernest Valveny. R-phoc: Segmentation-free word spotting using cnn. *arXiv preprint arXiv:1707.01294*, 2017.

[40] Suman Ghosh and Ernest Valveny. Text box proposals for handwritten word spotting from documents. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 1–18, 2018.

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

[42] Sheng He and Lambert Schomaker. Delta-n hinge: rotation-invariant features for writer identification. In *International Conference on Pattern Recognition*, 2014.

[43] Sheng He and Lambert Schomaker. Beyond ocr: Multi-faceted understanding of handwritten document characteristics. *Pattern Recognition*, 63:321–333, 2017.

[44] Charles Higounet. *Η ΓΡΑΦΗ*, volume Η ΓΡΑΦΗ. Daedalus Editions, Athens, Greece, 2006. ISBN 9602270187.

[45] Nicholas R Howe, Toni M Rath, and R Manmatha. Boosted decision trees for word recognition in handwritten document retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 377–383, 2005.

[46] Rajiv Jain and David Doermann. Offline writer identification using k-adjacent segments. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 769–773. IEEE, 2011.

[47] Rajiv Jain and David Doermann. Writer identification using an alphabet of contour gradient descriptors. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 550–554. IEEE, 2013.

[48] Rajiv Jain and David Doermann. Combining local features for offline writer identification. In *2014 14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2014.

[49] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[50] Gyeonghwan Kim and Venu Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(4):366–379, 1997.

[51] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 560–564. IEEE, 2013.

[52] Thomas Konidaris, Basilios Gatos, Kostas Ntzios, Ioannis Pratikakis, Sergios Theodoridis, and Stavros J Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(2-4):167–177, 2007.

[53] George Lakoff. *Cognitive models and prototype theory.* MIT Press, Cambridge, MA, 1999.

[54] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan Yuille, and Christof Koch. Adaboost for text detection in natural scene. In *2011 International Conference on Document Analysis and Recognition*, pages 429–434. IEEE, 2011.

[55] Yiqing Liang, Michael C Fairhurst, and Richard M Guest. A synthesised word approach to word retrieval in handwritten documents. *Pattern Recognition*, 45(12): 4225–4236, 2012.

[56] Laurence Likforman-Sulem, Abderrazak Zahour, and Bruno Taconet. Text line segmentation of historical documents: a survey. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(2-4):123–138, 2007.

[57] Chen Liu. *Probabilistic Siamese Networks for Learning Representations.* PhD thesis, University of Toronto, 2013.

[58] Georgios Louloudis, Basilios Gatos, Ioannis Pratikakis, and Constantin Halatsis. Text line and word segmentation of handwritten documents. *Pattern Recognition*, 42(12):3169–3183, 2009.

[59] Georgios Louloudis, Nikolaos Stamatopoulos, and Basilios Gatos. Icdar 2011 writer identification contest. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1475–1479. IEEE, 2011.

[60] Georgios Louloudis, Basilios Gatos, and Nikolaos Stamatopoulos. Icfhr 2012 competition on writer identification challenge 1: Latin/greek documents. In *ICFHR*, pages 829–834. Citeseer, 2012.

[61] Georgios Louloudis, Basilios Gatos, Nikolaos Stamatopoulos, and A Papandreou. Icdar 2013 competition on writer identification. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1397–1401. IEEE, 2013.

[62] Georgios Louloudis, Basilios Gatos, Nikolaos Stamatopoulos, and A Papandreou. Icdar 2013 competition on writer identification. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1397–1401. IEEE, 2013.

[63] Sabri A Mahmoud, Irfan Ahmad, Mohammad Alshayeb, Wasfi G Al-Khatib, Mohammad Tanvir Parvez, Gernot A Fink, Volker Märgner, and Haikal El Abed. Khatt: Arabic offline handwritten text database. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 449–454. IEEE, 2012.

[64] Muhammad Imran Malik, Marcus Liwicki, Linda Alewijnse, Wataru Ohyama, Michael Blumenstein, and Bryan Found. Icdar 2013 competitions on signature verification and writer identification for on-and offline skilled forgeries (sigwicomp 2013). In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1477–1483. IEEE, 2013.

[65] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svms for object detection and beyond. In *2011 International conference on computer vision*, pages 89–96. IEEE, 2011.

[66] R Manmatha and Jamie L Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1212–1225, 2005.

[67] U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.

[68] Luc Mioulet, Utpal Garain, Clément Chatelain, Philippine Barlas, and Thierry Paquet. Language identification from handwritten documents. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 676–680. IEEE, 2015.

[69] Yadong Mu, Shuicheng Yan, Yi Liu, Thomas Huang, and Bingfeng Zhou. Discriminative local binary patterns for human detection in personal album. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[70] Naila Murray and Florent Perronnin. Generalized max pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2473–2480, 2014.

[71] Anguelos Nicolaou, Fouad Slimane, Volker Maergner, and Marcus Liwicki. Local binary patterns for arabic optical font recognition. In *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*, pages 76–80. IEEE, 2014.

[72] Anguelos Nicolaou, Andrew D Bagdanov, Marcus Liwicki, and Dimosthenis Karatzas. Sparse radial sampling lbp for writer identification. In *2015 13th International*

Conference on Document Analysis and Recognition (ICDAR), pages 716–720. IEEE, 2015.

[73] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29 (1):51–59, 1996.

[74] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[75] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[76] Toni M Rath and Raghavan Manmatha. Features for word spotting in historical manuscripts. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 218–222. IEEE, 2003.

[77] Toni M Rath and Raghavan Manmatha. Word image matching using dynamic time warping. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–521. IEEE, 2003.

[78] Jose Rodriguez-Serrano, Florent Perronnin, et al. A model-based sequence similarity with application to handwritten word spotting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2108–2120, 2012.

[79] Leonard Rothacker, Sebastian Sudholt, Eugen Rusakov, Matthias Kasperidus, and Gernot A Fink. Word hypotheses for segmentation-free word spotting in historic document images. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 1174–1179. IEEE, 2017.

[80] Jamie L Rothfeder, Shaolei Feng, and Toni M Rath. Using corner feature correspondences to rank word images by similarity. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 3, pages 30–30. IEEE, 2003.

[81] Marcal Rusinol, David Aldavert, Ricardo Toledo, and Josep Lladós. Browsing heterogeneous document collections by a segmentation-free word spotting method. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 63–67. IEEE, 2011.

[82] Marçal Rusiñol, David Aldavert, Ricardo Toledo, and Josep Lladós. Efficient segmentation-free keyword spotting in historical document collections. *Pattern recognition*, 48(2):545–555, 2015.

[83] Mathias Seuret, Anguelos Nicolaou, Dominique Stutzmann, Andreas Maier, and Vincent Christlein. Pixel level handwritten and printed content discrimination in scanned documents. In *International Conference on Frontiers Handwriting recognition ICFHR, 2020*. IEEE, 2020.

[84] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.

[85] L Shapiro and G Stockman. Computer vision prentice hall. *Inc., New Jersey*, 2001.

[86] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[87] Nabin Sharma, Ranju Mandal, Rabi Sharma, Umapada Pal, and Michael Blumenstein. Icdar2015 competition on video script identification (cvsi 2015). In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1196–1200. IEEE, 2015.

[88] Baoguang Shi, Cong Yao, Chengquan Zhang, Xiaowei Guo, Feiyue Huang, and Xiang Bai. Automatic script identification in the wild. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 531–535. IEEE, 2015.

[89] Baoguang Shi, Xiang Bai, and Cong Yao. Script identification in the wild via discriminative convolutional neural network. *Pattern Recognition*, 52:448–458, 2016.

[90] Baoguang Shi, Xiang Bai, and Cong Yao. Script identification in the wild via discriminative convolutional neural network. *Pattern Recognition*, 52:448–458, 2016.

[91] Panagiotis Sidiropoulos, Stefanos Vrochidis, and Ioannis Kompatsiaris. Content-based binary image retrieval using the adaptive hierarchical density histogram. *Pattern Recognition*, 44(4):739–750, 2011.

[92] Sargur Srihari, Harish Srinivasan, Pavithra Babu, and Chetan Bhole. Spotting words in handwritten arabic documents. In *Electronic Imaging 2006*, pages 606702–606702. International Society for Optics and Photonics, 2006.

[93] Nikolaos Stamatopoulos, Basilis Gatos, Georgios Louloudis, Umapada Pal, and Alireza Alaei. Icdar 2013 handwriting segmentation contest. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1402–1406. IEEE, 2013.

[94] Sebastian Sudholt and Gernot A Fink. Phocnet: A deep convolutional neural network for word spotting in handwritten documents. *arXiv preprint arXiv:1604.00187*, 2016.

[95] Kengo Terasawa and Yuzuru Tanaka. Slit style hog feature for document image word spotting. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 116–120. IEEE, 2009.

[96] Mäenpää Topi, Ojala Timo, Pietikäinen Matti, and Soriano Maricor. Robust texture classification by subsets of local binary patterns. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 935–938. IEEE, 2000.

[97] Adnan Ul-Hasan, Muhammad Zeshan Afzal, Faisal Shafait, Marcus Liwicki, and Thomas M Breuel. A sequence learning approach for multiple script identification. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1046–1050. IEEE, 2015.

[98] Ranjith Unnikrishnan and Ray Smith. Combined script and page orientation estimation using the tesseract ocr engine. In *Proceedings of the international workshop on multilingual OCR*, pages 1–7, 2009.

[99] Georgios Vamvakas, Basilis Gatos, and Stavros J Perantonis. Handwritten character recognition through two-stage foreground sub-sampling. *Pattern Recognition*, 43(8): 2807–2816, 2010.

[100] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[101] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1457–1464. IEEE, 2011.

[102] Hao Wei, Kai Chen, Anguelos Nicolaou, Marcus Liwicki, and Rolf Ingold. Investigation of feature selection for historical document layout analysis. In *Image Processing Theory, Tools and Applications (IPTA), 2014 4th International Conference on*, pages 1–6. IEEE, 2014.

[103] Eric W. Weisstein.

[104] Richard E Williamson and Hale F Trotter. Multivariable mathematics: Linear algebra, calculus. *Differential Equations*, 2(4), 2004.

[105] Weixin Yang, Lianwen Jin, and Manfei Liu. Deepwriterid: An end-to-end online text-independent writer identification system. *IEEE Intelligent Systems*, 31(2):45–53, 2016.

[106] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):915–928, 2007.

[107] Guangyu Zhu, Xiaodong Yu, Yi Li, and David Doermann. Language identification for handwritten document images using a shape codebook. *pattern recognition*, 42 (12):3184–3191, 2009.