Local gradient histogram features for word spotting in unconstrained handwritten documents

José A. Rodríguez[*+] , *Florent Perronnin*[+]
[+] Xerox Research Centre Europe
6, Chemin de Maupertuis, 38240 Meylan (France)
[*] Computer Vision Center (Universitat Autonoma de Barcelona)
Edifici O, Campus Bellaterra, 08193 Bellaterra (Spain)
jrodriguez@cvc.uab.es, Florent.Perronnin@xrce.xerox.com

Abstract

In this article we propose a local descriptor for an unconstrained handwritten word spotting task. The presented features are inspired by the SIFT keypoint descriptor, widely employed in computer vision and object recognition, but underexploited in the handwriting recognition field. In our approach, a sliding window moves from left to right over a word image. At each position, the window is subdivided into cells, and in each cell a histogram of orientations is accumulated. Experiments using two different word spotting systems - hidden Markov models and dynamic time warping - demonstrate a very significant improvement when using the proposed features with respect to the state-of-the-art ones.

Keywords: word spotting, local gradient histogram features, hidden Markov models, dynamic time warping

1. Introduction

Handwritten word spotting (HWS) is the pattern classification task which consists in detecting keywords in handwritten document images [8]. As is the case for handwritten word recognition, one of the main difficulties is the high intra-writer and inter-writer variability. Therefore, an important decision at the representation phase is the choice of a suitable word descriptor.

In the literature on handwriting recognition, two main types of word descriptors can be identified. On the one hand, *holistic* approaches [7, 19] extract a single feature vector per image. These approaches are limited in performance but are sufficient for certain tasks such as digit recognition, character recognition or small-vocabulary word recognition.

On the other hand, *local* or *sequential* approaches provide a more accurate description of word images, describing them as 1-D sequences of feature vectors. One possibility is to segment the word horizontally into subcharacter units called graphemes [3, 5], and to extract features from each grapheme to obtain a left-to-right sequence. Recently, a more common approach is to let a sliding window move from left to right over the image and extract features from the pixels in the window [9, 17, 13, 16]. There exist suitable algorithms for efficient treatment of such sequences, like dynamic time warping (DTW) [15] or hidden Markov models (HMM) [11]. Local descriptors are used in this work since the performances reported in the literature are superior for this case.

The scenario of the present work is an unconstrained handwritten word spotting task. Therefore, the goal is to detect keywords in realistic, unrestricted conditions. This includes a variety of writer styles, document layouts, spontaneous writing, artifacts or spelling mistakes. This is to be contrasted with some previous handwritten word spotting works that do not always consider these intricacies. For instance, word spotting is mostly applied to the indexing of historical documents frequently produced by a single or a few writers [8, 16, 2, 1], a situation where the variability is significantly reduced.

The main contribution of this work is a new sequential feature set that obtains performance well beyond the state-of-the-art in an unconstrained word spotting task. This feature set is inspired by the SIFT keypoint descriptor [6], which is basically a histogram of oriented gradients at localized portions of an image. Our application of this idea for describing words consists in generating a *sequence* of such descriptors by moving a sliding window over the word image. Although the SIFT features are not common in document analysis, it is worth mentioning some works employing related concepts. For instance, in holistic approaches similar features have been proposed for describing isolated characters [18] or as a subset of a word descriptor (GSC features) [19], although these are binary and applied globally to the image. The most similar features are presented in [4] for a word recognition -not spotting- task. But there are some differences like their window subdivision, which lacks of horizontal splits and is irregular, whereas we show that incorporating horizontal splits and using a regular structure leads to improved results. In any case, note that it is not our aim to apply the whole SIFT approach but only to inspire from the keypoint descriptor part.

A secondary contribution of the present article is to

provide an experimental comparison of local word descriptors for word spotting. To give a perspective somehow independent of the classifier employed, we provide results both for DTW and HMM. This is not the first time (see [12]) that a feature comparison is done for DTW. However, [12] focuses on simple features on a fairly constrained task and using only DTW. We consider both DTW and HMM and recent features in key works (such as [17]) for an unconstrained task.

The rest of the article is organized as follows. In Section 2 we provide some context by offering a brief description of our word spotting system. In Section 3, the state-of-the-art features that have been tested are presented. This is followed by the introduction of the proposed features, in Section 4. Experimental results are reported and discussed in Section 5 and, finally, conclusions are drawn in Section 6.

2. HWS system overview

This study is part of a larger research project whose goal is to detect keywords in an incoming flow of scanned letters. One potential application of this is the routing of mails based on the presence of certain keywords. Of course, the outcome of this work can be applied to other problems of similar nature, such as indexing historical documents [8], document categorization and, more generally, metadata extraction from document images.

Although a detailed description of the complete system pipeline is out of the scope of this work, we provide a brief overview of the process:

(1) A segmentation algorithm extracts sub-images that potentially represent words, employing state-of-the-art techniques based on projection profiles and clustering of gap distances.

(2) A simple classifier using holistic features is applied to perform a first rejection pass (*fast rejection*), which prunes out about 90% of the segmented words while falsely rejecting only 5% of the keywords.

(3) The non-pruned word images are normalized with respect to slant, skew and text height, using standard techniques.

(4) For each normalized image, a sequence of feature vectors is computed by moving a window from left to right over the image and by extracting a feature vector at each position.

(5) A score is assigned to each feature vector sequence. The corresponding image will be flagged as the keyword if the score exceeds a predefined threshold.

The present study is focused on point (4) and, more precisely, it deals with the choice of a robust feature set. Several state-of-the-art features are evaluated, but due to their limited performance a new feature set is proposed and compared to them. In order to make the results relatively independent of the scoring method of step (5), all tests will be carried out using two scoring mechanisms: HMM and DTW.

3. State-of-the-art features

In this section we describe the state-of-the art features employed as baseline in our experiments.

3.1. Column features

One of the most influencing works in offline handwriting recognition using HMMs is probably the one by Marti & Bunke [9]. In their work, the features are taken columnwise. From the set of foreground pixels in each image column, 9 geometrical features are computed, namely: the total number of foreground pixels, the mean, second order moment, minimum and maximum of their positions, the differences between the maximum and minimum values with respect to the previous column, the number of black-white transitions, and the number of foreground pixels between upper- and baseline.

3.2. Pixel count features

In another well-known work by Vinciarelli et al. [17], a sliding window moves from left to right over the word. Contrarily to the column features, the width of the sliding window comprises several columns. At each position, the height of the window is adjusted to the area actually containing pixels, and then it is split into a 4×4 cell grid. The pixel counts in each of these cells are concatenated to form a 16-dimensional feature vector. To avoid boundary problems at the very first or very last positions of the sliding window, we assume the area outside the image consists of zero-valued pixels.

3.3. Gaussian filter features

The most popular approach for word spotting, the one using DTW, was introduced by Rath & Manmatha [13]. Therefore, we also implemented a set of features by these authors which are described in [12] and involve Gaussian filters. First, (a) a Gaussian filter, (b) an horizontal gradient filter and (c) a vertical gradient filter are computed. Then, a feature vector is built for each column by concatenating the values of (a), (b) and (c) corresponding to that column. In the cited work, the images have a height of 15 pixels, so an image is described using a sequence of 45-dimensional feature vectors.

The reason why the best performing features of the study [12] have not been preferred is due to the fact that they happen to be a subset of the Marti & Bunke features. We carried out preliminary experiments and observed that the performance of this subset is poorer than the performance of the full set. Furthermore, since the chosen features contain gradient information, they can be interesting to compare to ours.

4. Local gradient histogram features

The main contribution of this work is the application of a new local feature type that has some points in common with Lowe's SIFT keypoint descriptor [6], showing that it leads to improved performance in a word spotting task.

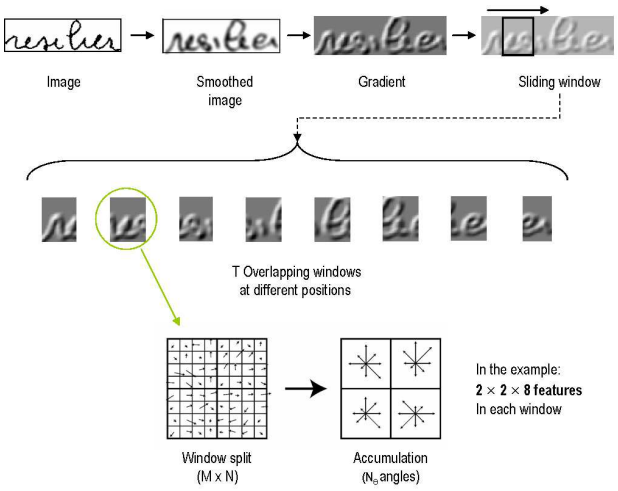

Figure 1. Feature extraction process. Note that the particular settings are exemplary and do not necessarily represent the optimal choice in the final system.

We refer to our sequential descriptor as "local gradient histogram features".

The particular details of the feature extraction process follow in the next paragraphs. As a complement to the explanation, Fig. 1 provides an overview of the process. It should be noted that in some parts of this figure the gradient has been represented by its magnitude, but it should always be thought of as a vector field.

4.1. Sliding window

Given an image $I(x, y)$ of height H and width W, we center at each column x a window of height H and width w. At each window position, a feature vector is computed that only depends on the pixels inside the window. Thus, a sequence of W feature vectors is obtained. One advantage of this sliding window approach is that it preserves the left-to-right nature of the writing.

4.2. Division of the window into cells

At each position, the sliding window is subdivided into rectangular cells. Different methods can be employed for this purpose. We propose three possibilities, illustrated in Fig. 2, namely:

(i) to split the window regularly into $M \times N$ cells of identical dimensions, as in [6]

(ii) to perform the $M \times N$ cell division only on the window area actually containing pixels, as proposed in [17], and

(iii) to do independent splits in the three zones determined by the upperline and baseline, thus resulting in a grid of $(A + B + C) \times N$ cells, where $A - 1$, $B - 1$ and $C - 1$ are the number of splits in each zone.

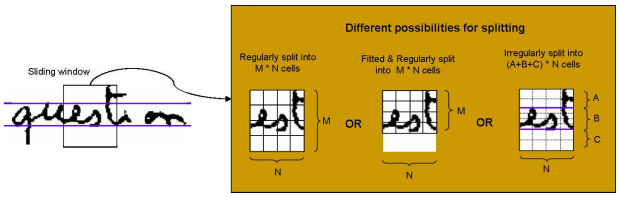

Figure 2. Possible grids for feature extraction

4.3. Gradient histogram computation

In each of the cells, local gradient histogram features are extracted. Denote $L(x, y)$ the result of convolving the image $I(x, y)$ with a smoothing filter, employed for denoising purposes. First, the horizontal and vertical gradient components G_x and G_y are determined as

$$G_x = L(x + 1, y) - L(x - 1, y) \tag{1}$$

and

$$G_y = L(x, y + 1) - L(x, y - 1). \tag{2}$$

Alternatively to the smoothing plus the gradient computation, a Gaussian derivative filter can be employed. In any case, the gradient magnitude m and direction θ are then obtained for each pixel with coordinates (x, y) as

$$m(x, y) = \sqrt{G_x^2 + G_y^2} \tag{3}$$

and

$$\theta(x, y) = \measuredangle(G_y, G_x), \tag{4}$$

where $\measuredangle$ is a function that returns the direction of the vector (G_x, G_y) in the range $[-\pi, \pi]$ by taking into account $\tan^{-1}(G_y/G_x)$ and the signs of G_x and G_y. It corresponds to the implementation of `atan2` in the C programming language.

Then the gradient angles are quantized into a number T of regularly spaced orientations, and the magnitudes for identical orientations are accumulated into a histogram. In other words, for each pixel with coordinates (x, y) we determine which of the T orientations is the closest to $\theta(x, y)$ and sum $m(x, y)$ to the corresponding bin.

Assigning gradients to the closest orientations may result in aliasing noise. To reduce its impact, the gradient magnitude of a pixel can be shared between the two closest bins, as determined by a linear interpolation in the angle domain. In particular, let α and $\frac{2\pi}{T} - \alpha$ denote the angle to the two closest bins for a particular pixel (see Fig. 3). Then the contribution of this pixel to the two bins is respectively:

$$m(x, y)\left[1 - \frac{T\alpha}{2\pi}\right], \text{ and } m(x, y)\frac{T\alpha}{2\pi}. \tag{5}$$

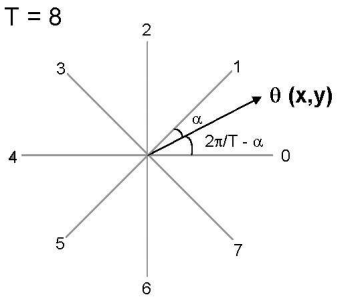

Figure 3. Angular bins for T=8 and angle differences of $\theta(x, y)$ to the two closest bins

Figure 4. Positive labelled samples for the word class *résiliation*

As in the case of the pixel count features, it is assumed that outside the image the pixel values are 0 to avoid boundary effects.

4.4. Frame normalization

The feature vector at one window position, sometimes called *frame*, is the concatenation of the gradient histograms computed in each cell. Experimentally, a performance gain is obtained when scaling each frame so that their components sum to 1. This improvement is likely due to the fact that this scaling performs a local contrast normalization.

This behaviour is not particular of our features. An improvement is also observed for the pixel count features (Sec. 3.2) when the frames are normalized, and therefore it will also be taken into account for that case. In this case, scaling the frames introduces an invariance with respect to stroke thickness.

4.5. Summary

To summarize, if in each window there are $M \times N$ cells (regular split) and each cell is represented by a histogram of T bins, each position of the sliding window will be characterized by a feature vector of $M \times N \times T$ components. The word is characterized by a sequence of W such vectors.

5. Experiments

In this section the experimental conditions and results of the feature comparison are presented.

5.1. Experimental conditions

The experiments are carried out on a dataset containing 630 real scanned letters (written in French) submitted to the customer department of a company. As mentioned in the introductory section, these data are unconstrained and therefore the letters contain different writing styles, artifacts, spelling mistakes and other types of noise. The word hypotheses are segmented from the page images (see Section 2) and the sub-images corresponding to certain keywords are manually labelled. Only the 10 most frequent word classes (e.g. "Monsieur", "Madame", "résilier", "contrat", etc.) are used in the experiments. From 208 to 750 positive examples are available for each keyword. In Fig. 4 some examples of the data labelled as *résiliation* (translated as "cancellation") are displayed. Prior to classification, samples undergo the mentioned fast rejection; therefore, all the following results refer to the set of non-pruned samples.

For all the non-pruned samples, all the described features are computed and the resulting classification performance is measured. Two classification methods are tested: HMM and DTW.

In HMM tests the employed similarity score is the log-likelihood outputted by the model. Training and testing is carried out using 5-fold cross validation. The dataset is initially split into 5 folds (ensuring that the same writer is not mixed among them). Models are trained using data from 4 folds and tested on the remaining fold. This is repeated 5 times. Word HMMs (traditional left-to-right HMMs without skip-state jumps) are trained using 10 states per character.

In DTW tests, 5 random images are used as queries. For an input image, the negative distance to the closest query is taken as a similarity score. The experiment is repeated 5 times and the results are averaged to reduce the effects of randomness. Of course, the same image sets are used for every tested feature set.

In both cases the performance of each word detector is evaluated by inspecting the DET curves [10]. These are tradeoff curves plotting false acceptance (FA) versus false rejection (FR) rates. Let θ_n be the threshold for word n, meaning that all samples with scores $S > \theta_n$ are retrieved. Then, FR(θ_n) is defined as the percentage of samples with label w_n with $S < \theta_n$. Similarly, FA(θ_n) is defined as the number of samples not labelled with w_n with $S > \theta_n$. For summarizing a curve with a single value, we employ the common average precision (AP) figure. For some results we report the mean of the AP across all words, which will be called mean AP or, shortly, mAP.

Table 1. Mean average precision (mAP) for the different grids using HMM

Grid type	mAP
Regular, unfitted grid $((1+4+1)\times 4)$	0.321
Regular, fitted grid (4×4)	**0.717**
Irregular grid (4×4)	0.655

Table 2. Mean average precision (mAP) for the different features using HMM

Features	mAP
Proposed local gradient features	**0.717**
Marti & Bunke [9]	0.329
Vinciarelli et al. [17]	0.336
Rath & Manmatha (Sec. 4.2 of [12])	0.135

5.2. Results and discussion

Determining the optimal grid type

Before comparing the proposed features to the state-of-the-art ones, we determine the best split type from the ones described in Section 4.2. The mAP values for the HMM using the different types of split are presented in Table 1. In the table, we also indicate the best settings obtained for each individual case. In all cases, the optimal number of orientation bins is found to be $T = 8$.

As shown in Table 1, the fitted regular grid is on the average significantly better than the unfitted grid and the irregular grid. Therefore, in all subsequent experiments, the proposed features use the fitted regular grid.

The superiority of the fitted grid is reasonable to the extent that the window focuses on the area with most information content. We expected a higher performance of the irregular grid. However, it depends on the accurate determination of the so-called baseline and upperline, which are less well defined in such a noisy scenario.

Experiments with HMM

The performances of the different feature types are tested in a HMM-based system. The mean of the average precision (mAP) across all the words for the different feature sets are shown in Table 2. The corresponding DET curves for the word *résiliation* are shown in Fig. 5.

It can be appreciated that the proposed features give much better performance than the state-of-the-art features. In particular, at FR=40% the FA rate is reduced by a factor from 3 to 10 compared to the next best features. In the particular case of the pixel count features, a 4×4 regular fitted grid is found to be the best setting, which confirms the choice in the original reference [17].

Two remarks should be done to the presented results. First, since the best found setting is a 4×4 grid with 8 angles, the features are 128-dimensional. Thus the important increase in performance that we obtain is at the expense of a much higher computational cost when compared to the Marti & Bunke or Vinciarelli features, that are 9 and 16-dimensional, respectively.

Second, in view of the low mAP values for the state-of-the-art features, it should be reminded that i) these results are for the set of non-pruned samples, a small and difficult subset of the input samples, and (ii) the task we are considering, spotting in unconstrained conditions, is already more difficult.

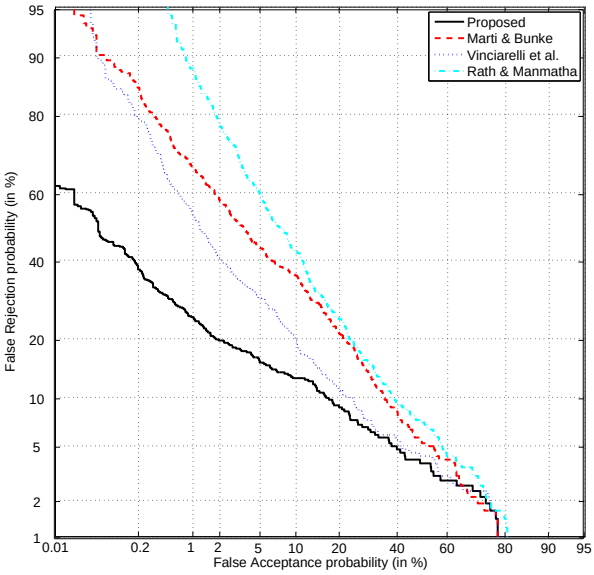

Figure 5. DET curves comparing the different features for the word *résiliation* using HMM

Table 3. Mean average precision (mAP) computed for the different features using DTW

Features	mAP
Proposed local gradient features	**0.254**
Marti & Bunke [9]	0.108
Vinciarelli et al. [17]	0.117
Rath & Manmatha (Sec. 4.2 of [12])	0.092

Experiments with DTW

We have also carried out experiments with DTW since it is very common in word spotting. Again, the mAP values are shown in Table 3. In Fig. 6 we show the particular DET curves for the word *résiliation*. As it can be appreciated, we obtain the same ordering of results as in the HMM case.

The same remarks as in the previous section apply, especially to understand the reasons for the small mAP values. A visual inspection of the best ranked samples shows that only the first couple of top samples are usually correct. Therefore, DTW can be used on these data just to retrieve the most similar samples but clearly not for robust spotting.

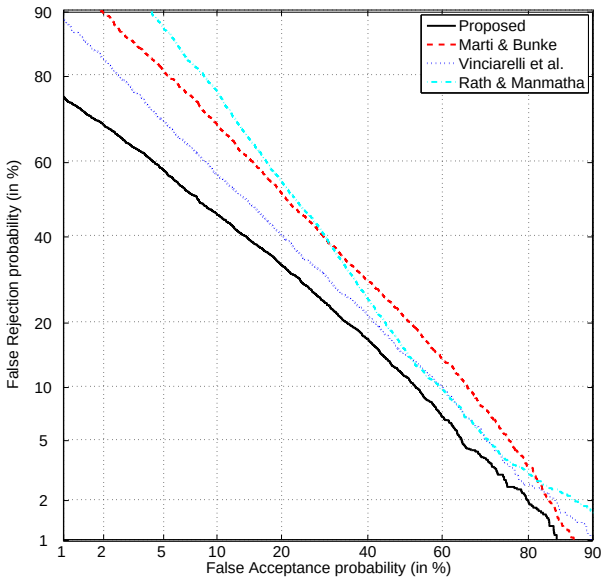

Figure 6. DET curves comparing the different features for the word *résiliation* using DTW

6. Conclusions and future work

In this paper, we propose a local feature set that obtains superior performance in a word spotting task when compared with other local state-or-the-art features. The authors hope the work also contributes as a benchmark of feature sets for word spotting. It should be made clear that the results have been obtained in unconstrained conditions, in contrast to many existing word spotting works where the source is a single writer.

One limitation of the introduced features is that they involve 128-dimensional feature vectors in our optimal setting. Such a high dimensionality has an negative impact on the computational cost. In our case, this expense is justified by the important increase in performance. Early experiments with PCA for dimensionality reduction resulted in significant decrease of performance. A future research may include the exploration of other techniques for dimensionality reduction such as non-negative matrix factorization, potentially suitable for our case since the proposed features are non-negative. Furthermore, an adequate combination of some of the presented features could lead to a performance improvement while reducing the computational cost.

A final remark is that we score word images the log-likelihood of a HMM. A posterior work has shown that this confidence measure can be improved by considering score normalization [14]. However, the conclusions of the feature comparison remain the same after score normalization.

Acknowledgements

The work of José A. Rodriguez is partially supported by the Spanish projects TIN2006-15694-C02-02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018). The authors would like to thank J. Lladós and G. Sánchez (CVC) for their support.

References

[1] T. Adamek, N. E. Connor and A. F. Smeaton, "Word matching using single closed contours for indexing handwritten historical documents", *Int. Journal on Document Analysis and Recognition*, 9(2):153–165, 2007.

[2] J. Chan, C. Ziftci and D. Forsyth, "Searching Off-line Arabic Documents", *CVPR*, 2006, pp 1455–1462.

[3] M. Y. Chen, A. Kundu and J. Zhou, "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network", *IEEE Transactions on PAMI*, 16(5):481–496, 1994.

[4] D. Guillevic and C. Y. Suen, "HMM word recognition engine", *Proc. of the 4th Int. Conf. on Document Analysis and Recognition*, 1997, pp 544–547.

[5] S. Knerr, E. Augustin, O. Baret and D. Price, "Hidden Markov Model Based Word Recognition and Its Application to Legal Amount Reading on French Checks.", *Computer Vision and Image Understanding*, 70(3):404–419, 1998.

[6] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *Int.J.Comput.Vision*, 60(2):91–110, 2004.

[7] S. Madhvanath and V. Govindaraju, "The Role of Holistic Paradigms in Handwritten Word Recognition", *IEEE Transactions on PAMI*, 23(2):149–164, 2001.

[8] R. Manmatha, C. Han and E. M. Riseman, "Word Spotting: A New Approach to Indexing Handwriting", *CVPR*, 1996, pp 631.

[9] U.-V. Marti and H. Bunke, "Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system", *Int. J. of Pattern Recognition and Artifi cial Intelligence*, 15:65–90, 2001.

[10] A. Martin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, "The DET curve in assessment of detection task performance", *Proc. of EuroSpeech'97*, 1997, pp 1895–1898.

[11] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. of the IEEE*, 77:257–286, 1989.

[12] T. M. Rath and R. Manmatha, "Features for Word Spotting in Historical Manuscripts", *Proc. of the 7th Int. Conf. on Document Analysis and Recognition*, 2003, pp 218.

[13] T. M. Rath and R. Manmatha, "Word Image Matching Using Dynamic Time Warping.", *CVPR*, 2003, pp 521–527.

[14] J. A. Rodríguez and F. Perronnin, "Score normalization for HMM-based handwritten word spotting using a universal background model", ICFHR'08.

[15] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition", *IEEE Transactions on Acoustics, Speech and Signal processing*, pp 159–165, 1978.

[16] K. Terasawa, T. Nagasaki and T. Kawashima, "Eigenspace Method for Text Retrieval in Historical Document Images", *Proc. of the 8th Int. Conf. on Document Analysis and Recognition*, 2005, pp 436–441.

[17] A. Vinciarelli, S. Bengio and H. Bunke, "Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models.", *IEEE Transactions on PAMI*, 26(6):709–720, 2004.

[18] M. Yasuda and H. Fujisawa, "An improvement of correlation method for character recognition", *Systems, Computers, Controls*, 10(2):29–38, 1979.

[19] B. Zhang, S. N. Srihari and C. Huang, "Word image retrieval using binary features", *Proc. of the SPIE Conf. on Document Recognition and Retrieval XI*, 2004.