Bachelorarbeit im Fach Informatik

Adapting the RWTH-OCR Handwriting Recognition System to French Handwriting

Rheinisch-Westfälische Technische Hochschule Aachen
Lehrstuhl für Informatik 6
Prof. Dr.-Ing. H. Ney

vorgelegt von:
Patrick Röder
Matrikelnummer 275695
E-Mail patrick.roeder@rwth-aachen.de

Gutachter:
Prof. Dr.-Ing. H. Ney
Prof. Dr. rer. nat. B. Leibe

Betreuer:
Dipl.-Inform. P. Dreuw

Dezember 2009

Erklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, im Dezember 2009

Patrick Röder

Abstract

This bachelor thesis investigates the use of the RWTH-OCR system on the French handwriting database RIMES. The RWTH-OCR system is based on the RWTH-ASR speech recognition system. The field of offline handwriting recognition is an open topic in research and in the past the RWTH-OCR system has been adapted to several languages as English or Arabic handwriting. The RWTH-OCR is a hidden Markov model based recognition system. The system deals with writing style variations by using only a few preprocessing steps and simple appearance based features. In addition further methods are applied to improve the results, such as model length estimation and discriminative training. Finally, the results achieved by the RWTH-OCR system are compared to the results of the official RIMES 2 evaluation campaign.

Acknowledgment

I would like to thank Prof. Dr.Ing. Hermann Ney for the possibility to work at the Chair of Computer Science VI of the RWTH Aachen University in the field of image recognition.

I would also like to thank Prof. Dr. Bastian Leibe, who kindly accepted to co-supervise the work.

Special thanks to Philippe Dreuw who supervised this work and had always a helping idea, when the work got stuck. In addition, I thank the complete Image Group for some helpful discussions and support, especially Christian Oberdörfer, who went through a similar stressful time and Stephan Jonas for some helpful hints. Thanks also to Jens Forster for teaching me the principals of scientific writing and some helpful conversations.

Also I want to thank my parents who gave me the possibility to do these studies.

Last but not least, I thank my sister and Christoph Pallasch for proofreading.

Contents

Chapter 1

Introduction

This work deals with the recognition of handwritten French words using the RWTH-OCR handwriting recognition system. The task of offline handwriting recognition is still an open topic in research. In general the error rates that are achieved by systems for offline handwriting recognition are too high to be used in commercial products and it is an aim of today's research to decrease the error rate in handwriting recognition. One use for handwriting recognition systems is the automatic digitalization of historical documents. Older documents are mostly handwritten and thus cannot be read by standard OCR systems that work on machine printed texts.

One problem of handwritten documents is the high variation of the letters depending on the writing style. There can be many ligatures in a handwritten text and it is difficult for a system to model all this variation. Another problem that occurs especially in historical documents is that the pages might be damaged. This causes artifacts that can lead to false recognition results.

The RWTH-OCR system was already used on several other handwriting databases and other languages like English or Arabic handwriting. In this work the HMM based RWTH-OCR system is adapted to French handwriting. The RWTH-OCR system is based on the RWTH-ASR system for automatic speech recognition which works on large vocabularies. The system works with simple appearance based features, and the adaptation is evaluated on the RIMES database. This database consists of handwritten French words and was recently used in several evaluation campaigns.

In chapter 2 the current state of the art of handwriting recognition is presented. Especially methods used for the recognition tasks on the RIMES database are described. Chapter 3 gives an overview about how the RWTH-OCR system is used for the French handwriting recognition. Also the preprocessing of the data is depicted. Then, chapter 4 looks closer to the RIMES database that is used for the experiments in this thesis. The experiments are presented in chapter 5. The results of the experiments are also compared to the results of two systems that took part in the official evaluation campaign of the RIMES 2 database. Finally, chapter 6 gives a summary and conclusion of this work.

Chapter 2

State of the Art

In handwriting recognition (HWR) mostly Hidden Markov Models (HMMs) are used. In newer approaches they are modified or combined with other structures like neural networks (NN). Most approaches, as for example the HMM approach, require a preprocessing of the data and a feature extraction. In this section some of the methods used in the ICDAR 2009 competition are presented.

2.1 Preprocessing

Methods like HMMs require a preprocessing on the input data in order to filter out noises like damages of the scanned documents or cursive writing style. The filtering of cursive handwriting is achieved by deslanting. To correct this the angle of the inclination needs to be found. There are several methods that can be applied. [Yanikoglu & Sandon 98] use the Sobel edge detector. [Sun & Si 97] introduce two methods. In one approach, parallelograms are fitted to components of a text and then the angle of the axis is corrected. The other approach makes use of histograms of gradients in an image. A peak in the histogram gives an angle for the shear transformation, by which the slanting is corrected.

Another step of the preprocessing is the binarization of the data, such that each pixel belongs either to the foreground or to the background. This can for example be achieved by defining an intensity threshold. An alternative to binarization is gray value normalization, where not all pixels are mapped to black or white. Instead, two thresholds are defined, one for black an one for white. Pixels in between these two thresholds do not change. In this work gray value normalization is used.

In the ICDAR 2009 competition on the RIMES database [Grosicki & Abed 09] other possible preprocessings are described. To normalize the writing styles one system performs a noise reduction as well as skew and slant corrections. Another system uses binarization, slant and rotation correction and determines writing lines to partition the sliding window.

2.2 HMM Based Approaches

As a very common approach to HWR, HMMs are used. The advantage of HMMs is that they are able to compensate variations in writing direction. HMMs as used in the RWTH-OCR system interpret a text part as a sequence of states. To each state a probability distribution is assigned. The transitions can be described by a stochastic finite state automaton.

Another approach that uses HMMs is the multi-stream segmentation free HMM by [Kessentini & Paquet^{+} 08]. This method combines two low level feature streams. One stream contains density features that are extracted from two sliding windows with different size, the other stream consists of contour features. Compared to results using only one type of features, the multi-stream approach clearly outperforms the other setups.

A different approach is given by [Choisy 07] with non-symmetric half-plane HMM. This method uses Markov Random Fields (MRF) for each column of pixels. The MRF information is synthesized using an HMM along the whole image.

2.3 Discriminative Approaches

One kind of a discriminative approach in HWR is the use of neural networks. One approach in this field are the Energy Based Models (EBMs) introduced by [LeCun & Chopra^{+} 07]. An EBM computes the energy between the input variables and the possible outputs. The best fitting output is found by minimizing the energy.

Another approach that uses neural networks, is the method of Multidimensional Recurrent Neural Network (MDRNN) [Graves & Schmidhuber 08]. This approach needs no preprocessing on the input data and is very flexible to use.

The system uses neural networks with recurrent connections of spatial and temporal dimensions, which occur in the data. To transform the two dimensional features into one dimension, the system consists of a hierarchy of MDRNN layers. This allows to model complex features in stages. The same system can be used for different tasks of handwriting recognition, like Arabic or English handwriting without any adaptation.

The MDRNN system won several competitions on handwriting recognition at the ICDAR 2009 conference.

Another discriminative approach is the Modified Maximum Mutual Information (M-MMI) by [Dreuw & Heigold^{+} 09] that is also used in this work. The approach is based on the Maximum Mutual Information (MMI) criterion and extended by an additional parameter that controls the smoothness of the criterion.

Chapter 3

System Overview

This section gives an overview of the system as shown in figure 3.1.

3.1 Preprocessing

The first step of the system is the preprocessing of the data. The preprocessing in this work consists of three main parts and is described in [Juan & Toselli+ 01].

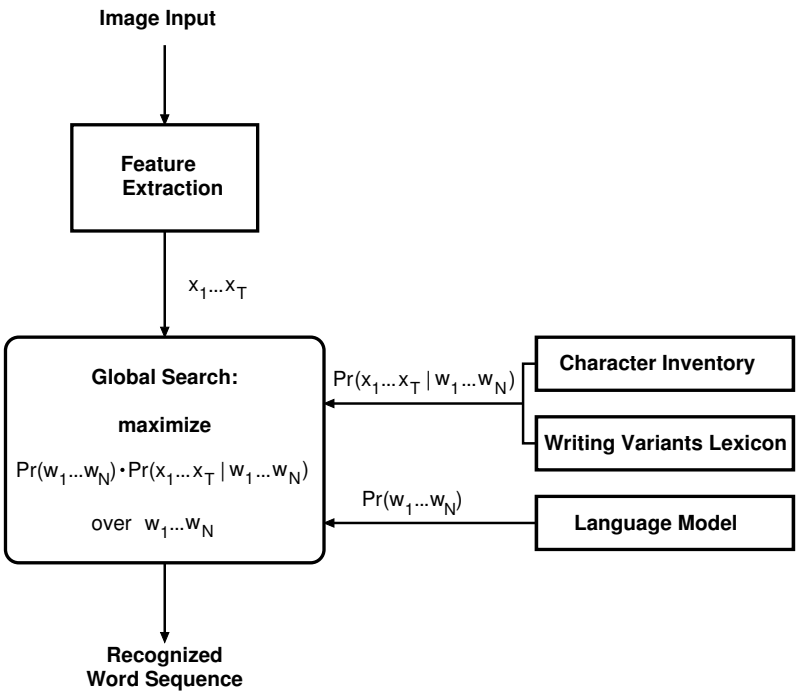


Figure 3.1. Architecture of the RWTH-OCR handwriting recognition system

1. Grey value normalization: First the gray values of the images of the corpus are remapped to achieve a better contrast. Therefore 70 percent of the pixels in an image are mapped to white while five percent are mapped to black. In the next step a median filter is used. Here, the value of a pixel is replaced by the median of the value of its neighborhood pixels.

2. Deslanting: The deslanting step corrects cursive handwriting, such that all letters are upright. To calculate the slant angle, a Sobel edge detector is used. After the computation of the slant angle, the words are adjusted by this angle in the opposite direction.

3. Size normalization: In the last main step the ascenders and descenders of the letters are removed. This is done in order to decrease their influence to the HMM, because there can be a huge variation that can not be compensated. First, the main body part of the word is computed. Since there are more ascenders than descenders, the ascenders are scaled down to 30 percent of the main body part height, while descenders are scaled down to 15 percent of the body part height.

Figure 3.2 shows with two example images of the RIMES database how the three steps of the preprocessing work. As a last step of the preprocessing, all images are scaled to a height of 50 pixels.

After applying the preprocessing steps on the images, on some images errors occur during the alignment in the training. The system is not able to align all characters in the image. This problem occurs mostly in images with very short words, consisting mostly of one or two characters. Because of this fact, 32 images have to be removed from the database.

3.2 Feature Extraction

Several types of features are used to analyze how the system performs on the RIMES database. In this section, the extraction of simple intensity based features and gradient based features using a Sobel operator are presented. In both cases the height of the pictures is scaled down to 16 pixels for the feature extraction.

3.2.1 Intensity

The intensity motion features use the original pixel values of the vertical slices of width one and are extended by the difference of the current slice to the previous slice. For the feature extraction all pictures are scaled down to 16 pixels height. In general, an image

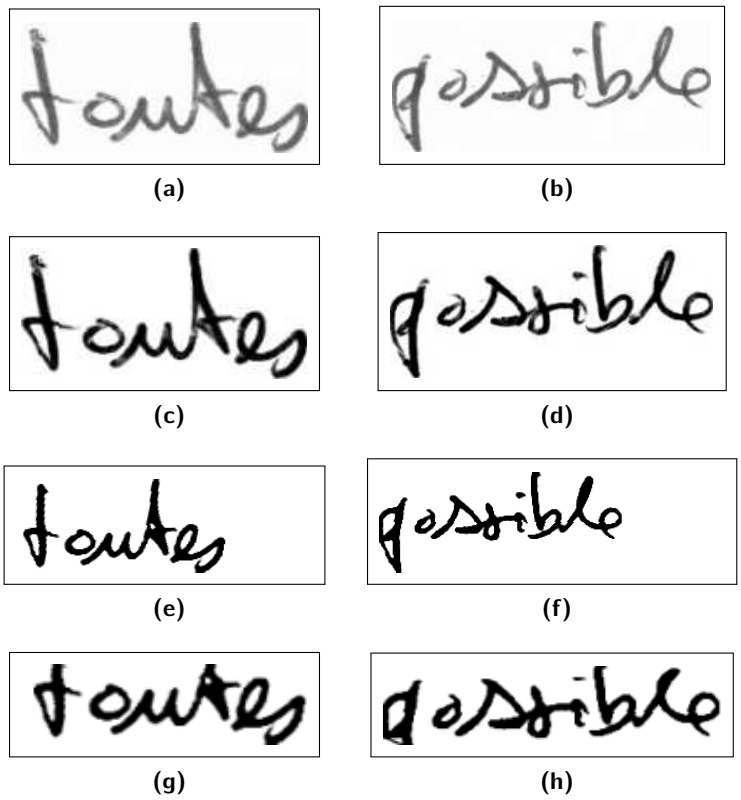


Figure 3.2. Example of the preprocessing on two example images of the RIMES database

$I(i,j)$ is represented by T feature vectors with height H as follows:

$$T \times H \rightarrow x_1^T := x_1, \cdots, x_T \quad with \quad x_t = \begin{bmatrix} I(t,1) \\ \vdots \\ I(t,H) \end{bmatrix} \tag{3.1}$$

The intensity-motion feature is represented by

$$x'_t = \begin{bmatrix} x_t \\ x_t - x_{t-1} \end{bmatrix} \tag{3.2}$$

3.2.2 Gradient Based Features

For this type of features Sobel filters are used. These are two types of filters that detect horizontal and vertical edges. In addition to the values of the horizontal and the vertical

Sobel filter, also the absolute value of the filtered image is used to derive the features. The reason for this is to add values which are linear independent regarding the original values. Formally the feature vector x_t of the Sobel filtered image S looks as the following:

$$x_t = \begin{bmatrix} S(t,1) \\ \vdots \\ S(t,H) \end{bmatrix} \tag{3.3}$$

3.2.3 Sliding Window

In order to include spatial and temporal context information to the current feature vector, a sliding window is used. The sliding window is shifted from left to right over the image. The feature vector of a sliding window of size $2\delta + 1$ has the form

$$x'_t = \begin{bmatrix} x_{t+\delta} \\ \vdots \\ x_t \\ \vdots \\ x_{t-\delta} \end{bmatrix} \tag{3.4}$$

The ideal size of the sliding window is the size of one character.

3.2.4 PCA Reduction

Since the number of feature dimensions increases with a higher size of the sliding window, the dimensions are reduced, using principal component analysis (PCA). The best results in several experiments are achieved using a PCA dimension reduction to 30 components.

3.3 Training

In this section the process of training the system for the recognition of French handwriting database is presented. We show what the basic ideas are and how they are improved.
As an estimation of the probability $p(x_1^T|w)$ of observing the feature sequence x_1^T given a word w the sum over all possible state sequences s_1^T has to be calculated as described in [Jonas 09].

$$p(x_1^T|w) = \sum_{[s_1^T]} p(x_1^T, s_1^T|w) \tag{3.5}$$

with

$$p(x_1^T, s_1^T|w) = \prod_{t=1}^{T} p(x_t, s_t|x_1^{t-1}, s_1^{t-1}, w) \tag{3.6}$$

Given these two equations, the probability of x_1^T given w_1^T can be computed. The first order Markov assumption assumes that the probability of x_t depends only on the state s_t that depends only on the previous state s_{t-1}. With a Viterbi approximation as described in [Ney 90], the term can be approximated as follows (for details see [Rybach 06]):

$$p(x_1^T|w_1^N) \approx \max_{s_1^T}\{\prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N)\} \tag{3.7}$$

The term $p(x_t|s_t, w_1^N)$ is called emission probability which calculates the probability of observing feature x_t in state s_t. The term $p(s_t|s_{t-1}, w_1^N)$ gives the transition probability of moving from state s_{t-1} to s_t.

The transition probabilities are controlled with the time distortion penalties which are explained in more detail in the following.

3.3.1 Character Modeling

The used HMM assigns three states to every character in a word. There are three possible moves from a state. The system allows to loop a state. Also the system can go to the next state, or skip the next state and go to the over next state. To influence the possibilities of the transitions, time distortion penalties (TDPs) are used. They add penalties to certain kinds of transitions, which make it less likely that the system uses these transitions. The first idea of the character modeling is to model only the characters that were seen during the training, without modeling whitespace, since the database contains only single words. Looking at the visualization of the character alignment reveals that due to the preprocessing, especially in the beginning of the word snippets, some whitespace is inserted. This fact causes the thought that it might be useful to add a whitespace to the lexicon, such that the system could recognize whitespace in front or behind the words. Figure 3.3 shows an example of a word where adding a whitespace at the beginning of the word leads to a better alignment. The alignment pictures show the preprocessed word snippets with the separation of characters as thin vertical lines. Above the picture, the recognized character is shown. A whitespace is described by 'si' for 'silence'. The bigger green and red vertical lines show the initially estimated borders of the word. The line below the word shows to which parts of the characters the states 0-1-2 of the hidden Markov model are mapped. This mapping is emphasized by a different shade of the background color. The upper picture shows that all white space features are aligned to the model of the character 'm', leading to a noisy model. If this occurs many times during the training, one state of a character is wasted to model the white space in front of the character.

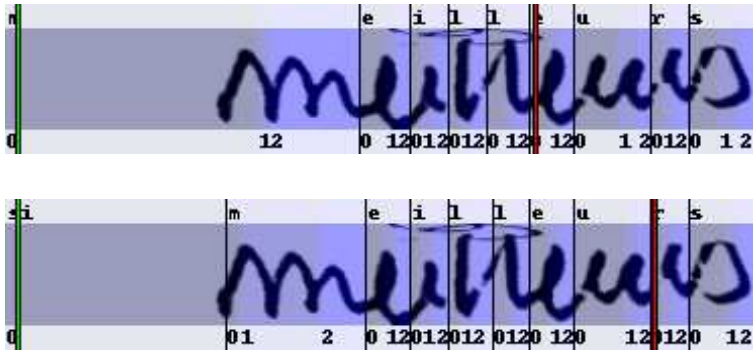


Figure 3.3. Example for the improvement of the alignment when whitespace is added to the lexicon. The upper picture shows the alignment without whitespace, the lower with whitespace (si).

3.3.2 Lexica

French handwriting in general is very similar to English or German handwriting. The basis is the Latin alphabet with its 26 characters that can be written in upper and lower case. In addition to that the French language has also accents, that occur mainly in combination with vowels. In French handwriting also ligatures can occur which are similar to the ligatures seen in German or English handwriting.

The standard lexicon contains all characters that can appear in a language. For the French handwritten database this results in a lexicon consisting of 82 characters that appear during the training procedure. These are the standard Latin characters in lower and upper case, the digits 0-9, as well as a number of special characters and characters that appear in the French language, especially accentuated letters like 'é'.

Another approach to the modeling of characters is to use a model length estimation (MLE). The MLE is used to split a character. This permits to assign more than three states of the HMM to broad handwritten characters like 'm', while small characters like 'i' are represented by only three states. This allows to change the spatial resolution of broad characters. In the lexicon characters are represented with an average length of five pixels per character state. This leads to an increase of the number of mixtures to be trained. The lexicon of the MLE system has 114 entries. There are five characters, which are modeled with nine states and 21 characters that are split into six states. Table 3.1 shows an overview, which characters are modeled with how many states.

Table 3.1. All entries of the characters in the lexicon. Shown are all characters and with how many states they are modeled in the MLE lexicon. Not shown is the whitespace which is always modeled by one state.

character	#states	character	#states	character	#states
a	3	A	6	0	3
b	3	B	6	1	3
c	3	C	3	2	3
d	6	D	9	3	3
e	3	E	6	4	3
f	3	F	6	5	3
g	3	G	6	6	3
h	3	H	6	7	3
i	3	I	3	8	3
j	3	J	6	9	3
k	3	K	3	’	3
l	3	L	3	°	3
m	9	M	9	-	9
n	3	N	6	%	3
o	3	O	3	/	3
p	3	P	3	^{2}	3
q	6	Q	6	à	3
r	3	R	6	ë	3
s	3	S	6	ï	3
t	3	T	3	ù	3
u	6	U	6	É	3
v	6	V	6	é	3
w	6	W	9	î	3
x	3	X	3	ô	3
y	6	Y	6	û	3
z	3	Z	3	ê	3
è	3	ç	3	â	3

3.3.3 Training Criteria

To train the system two approaches are used. First, a maximum likelihood (ML) training criterion is used. Second, we try a discriminative approach using the modified maximum mutual information (M-MMI) training criterion introduced by [Heigold & Deselaers$^+$ 08] and applied successfully to Arabic handwriting by [Dreuw & Heigold$^+$ 09].

Table 3.2. Statistics of the test set lexica.

	3-2	MLE
#lemma	1,641	1,641
#mixtures	244	340
#densities	15,840	18,486

3.4 Recognition

The recognition is performed in a single pass. The system uses the trained models described in the previous section. Since this work is limited to single word recognition, no language model is used.

During the recognition closed lexica are used. The lexica contain only the entries of the words that appear in the set on which the recognition is performed. Thus the lexicon size of the development set is 1,588 and the size of the test set lexicon is 1,636 words. Table 3.2 shows the statistics of the test lexica. In the next chapter, the database is presented in more detail.

Chapter 4

The Database

The first competition on the RIMES (Reconnaissance et Indexiation de données Manuscrites et de fac similÉS) database was in 2007. Since then there have been two more competitions in 2008 [Grosicki & Carre^{+} 09] and 2009 [Grosicki & Abed 09]. Each of the competitions consists of multiple subtasks. In the first competition the subtasks were isolated character recognition, structuring of letters, writers identification and recognition of logos of companies in the head of letters. All in all about 1,300 volunteers wrote circa 5,600 mails that were processed. This results in more than 12,600 partly or entirely handwritten pages with more than 250,000 words. Table 4.1 gives a summary of the statistics of the database.

The data was collected by giving an assignment to volunteers. The volunteers had to write a fax, a letter or to fill out a form to send it to companies in order to enter or withdraw from a contract or to ask questions or to write complaints. But instead of sending the letters, they were digitized and processed to get the data for the different recognition tasks.

During the second competition the logo recognition task was replaced by the recognition of single words. The other three tasks remained. The third competition took place in the context of the ICDAR 2009 conference. Also, for every new evaluation phase the provided data was extended to offer more data for training. The database of the RIMES 2 evaluation contains about 50,000 snippets of handwritten French words. The experiments in this bachelor thesis are done on the data of the second competition in the task of single word recognition.

Figure 4.1 shows some examples of word snippets from the RIMES database.

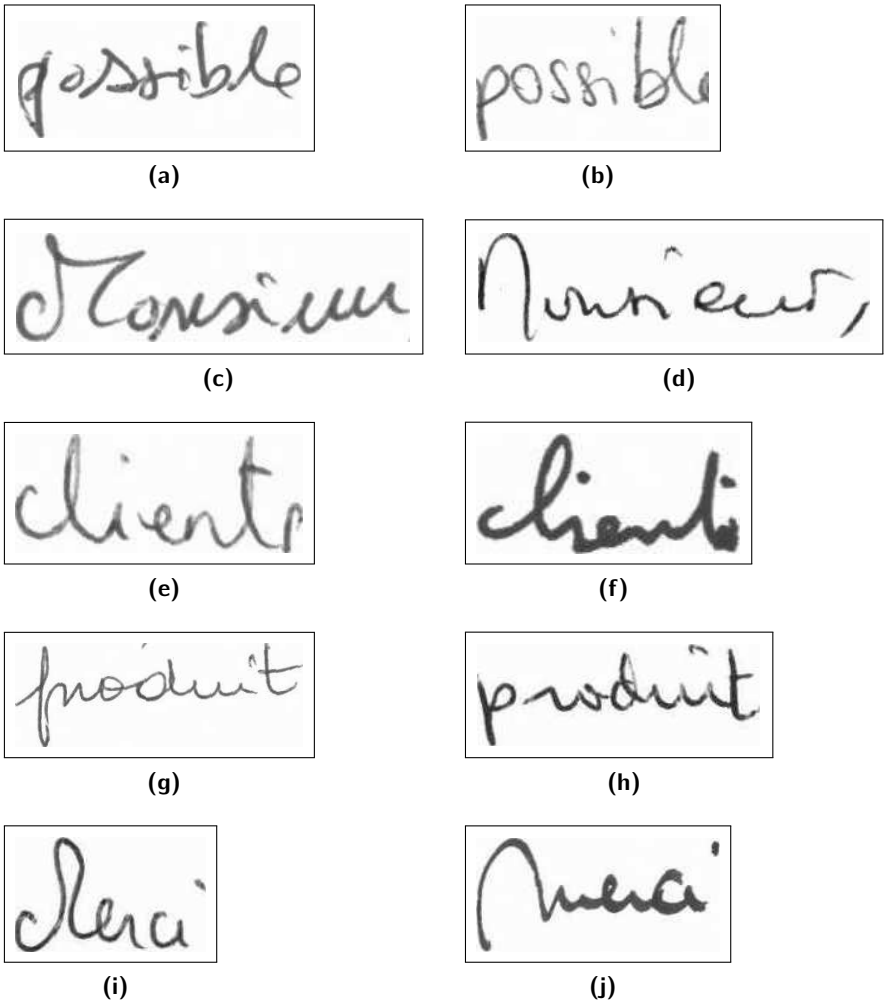


Figure 4.1. Some example images of the RIMES database with different writing styles.

Table 4.1. Statistics table of the RIMES database

#writers	1,300
#mails	5,600
#pages	12,600
#words	250,000

Chapter 5

Experimental Results

In this chapter the results of the experiments on the RIMES database are presented.

The data is divided in three sets, a training set, a validation set and a test set. The training set contains 36,444 words, the validation 7,786 and the test set 7,542 words. For the evaluation on the test set there are two separate tasks. One is to recognize a word given the complete lexicon of the test set. In the other task there is a lexicon of one hundred words from which the right one has to be found. In this thesis the focus is on the recognition task on the complete test lexicon.

The experiments of the RIMES 2 database are mainly performed on the pictures of the training set for training and the validation set for recognition. To compare the results to other groups, which participated in the evaluation campaign, some recognitions are additionally done on the set.

All experiments on the RIMES database are done with an HMM with three states and two repetitions.

Error Measurement

The error rates presented in the results part measure the word error rate (WER) and the character error rate (CER) the system achieved. The error rates are computed by summing up the number of insertions, deletions and substitutions of words respectively characters and dividing the sum by the total number of words respectively characters that are seen during the recognition process. This can be expressed by the equation:

$$ER = \frac{\#insertions + \#deletions + \#substitutions}{\#totalcount} \tag{5.1}$$

Since the evaluation campaign compares the WER, the main focus in this thesis is set on the word errors.

5.1 Raw Intensity without PCA

In order to obtain a baseline result of the RWTH-OCR system, an experiment on the raw image data is performed. In this experiment motion features of height 16 are extracted

Table 5.1. Results with various windows size for motion features without PCA. The best results are bold.

window size	WER	CER
1	85.48	26.38
3	82.64	24.86
5	80.02	24.36
7	77.34	24.02
9	75.25	**24.02**
11	**73.44**	24.16

from the unpreprocessed image data. For the experiment on the raw data an HMM with three states and one repetition is used, because with two repetitions the system is not able to align the characters of all words.

The results on window size nine, which achieves the best results in the later parts, achieves high error rates. The WER is 92.64%, while the CER is 42.60%. This shows that preprocessing is necessary.

5.2 Preprocessing

The results of section 5.1 show the necessity of preprocessing, as described in section 3.1 . In these experiments we use an HMM with three states and two repetitions.

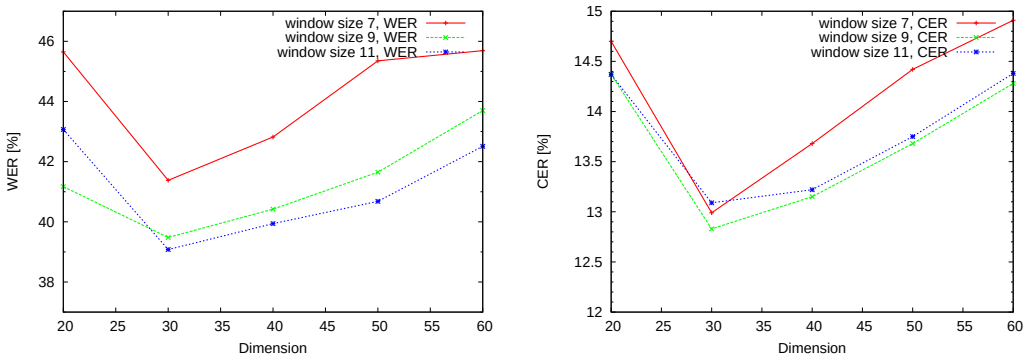
We perform an experiment without PCA estimation with a fixed number of dimensions and a variable window size between. The results in table 5.1 show that a higher windows size leads to better results.

Compared to the results on the raw image data, the error rates on the preprocessed image data improve. Especially the CER is reduced from 42% to 24%. But still the WER is about 75%, which is still high. Therefore in the next step, a principal component analysis (PCA) is added to the training process, to reduce the data dimensions which grow with larger window size.

5.3 Raw Intensity with PCA

The experiments on the intensity slices motion features with PCA matrix estimation are done to find the best fitting size of the sliding window and the dimensions.

Figure 5.1 shows the error rates of several window sizes with various dimensions. The smallest WER is achieved with a window size of 11 and 30 dimensional data, while the



(a) **(b)**

Figure 5.1. Results of the ML training with different window sizes and dimensions.

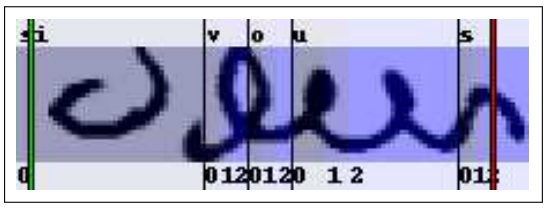
best CER is achieved with window size 9 and 30 dimensions. The result with window size 11 might be caused by overfitting, since the context that is taken into account is larger than the average size of one character. Therefore, we decide to continue the experiments primarily with the window size 9, 30 dimensions setup.

5.4 Time Distortion Penalties (TDPs)

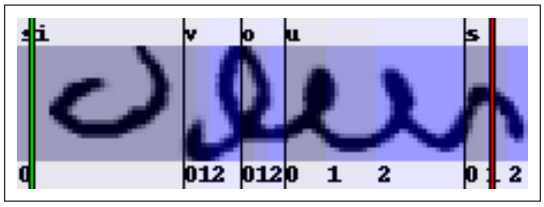
During the training process the time distortion penalties (TDPs) are adjusted by visual inspection of the alignment, based on the different values for the TDPs. The main focus in this part are on the penalties for skip and silence exit. The visualization of the training sessions with different values for these parameters shows that a skip of 3.0 and a silence exit penalty of 100.0 achieve the best results. This setup is applied to the best known configuration with window size 9 and 30 dimension. The tuning of the TDPs results in a WER of 37.09% and a CER of 12.45%. Figure 5.2 shows the improvement of a higher silence exit penalty.

5.5 Sobel

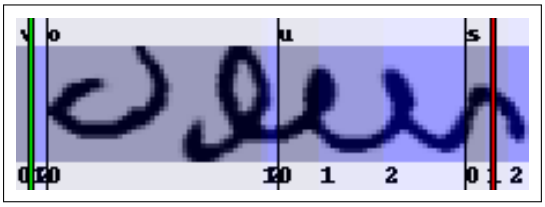
This experiment is done on the best setup for the motion experiment with PCA estimation. On this setup we do experiments without and with PCA estimation. In addition a second experiment with PCA estimation with window size 9 and 40 dimensions is performed. Table 5.2 shows the results of the first experiment and compares them to the best result of the motion features with PCA estimation. The results of the Sobel features with 39.93% WER are more than two percent higher than the best results of the



(a)



(b)



(c)

Figure 5.2. Example of a better alignment with a higher silence exit penalty. The image can be modeled without whitespace. If the silence exit penalty is too low as in (a) and (b) the system aligns a whitespace at the beginning of the word which leads to an incorrect alignment. Although in (c) the whitespace is removed, the alignment is still not correct.

motion features. Also the character error rate is almost one percent higher. This shows that Sobel features do not work as good as intensity features on the French handwriting database. This is in accordance to the results on English handwriting achieved by [Jonas 09].

5.6 Feature Summary

The summary results in table 5.3 show that the intensity features on the preprocessed images with PCA estimation in training achieve the best results. Looking at the experiments on intensity features, the preprocessing leads to a clearly lower error rate which is further improved by adding a PCA reduction. On the Sobel features, the preprocessing only leads to a smaller improvement of the error rates, while the PCA reduction in this

Table 5.2. Results of Sobel features compared to the results achieved on motion features. All experiments are done with PCA reduction.

setup	WER	CER
9-30 intensity	39.48	12.83
+ TDP opt.	**37.09**	**12.45**
9-30 Sobel + TDP opt.	39.93	13.18

Table 5.3. Summary Feature Table

Features	WER	CER
intensity	92.68	42.60
+ UPV preprocessing	75.25	24.02
+ PCA	**37.09**	**12.45**
Sobel	92.28	42.60
+ UPV preprocessing	88.26	26.44
+ PCA	39.93	13.18

case also improves the error rates a lot. Comparing the best results of intensity and Sobel features, the intensity features perform almost three percent better on the WER and almost one percent better on the CER. Similar results are reported by [Jonas 09] on the English handwriting database IAM, where Sobel features are outperformed by intensity features by about one percent in the WER. Therefore, in the next step we further try to improve the error rate on the intensity features by using a model length estimation.

5.7 Model Length Estimation

A look at the alignment of the characters shows that it might be more reasonable to model characters with a different number of states. In that way it should be possible to map the states of the HMM better to characters of different length.

Although better results are expected, the results do not support this. The MLE approach leads to a WER of 43.12% and a CER of 15.69%. In contrast to the results reported in [Dreuw & Heigold^{+} 09] on the French handwriting database no improvement could be achieved.

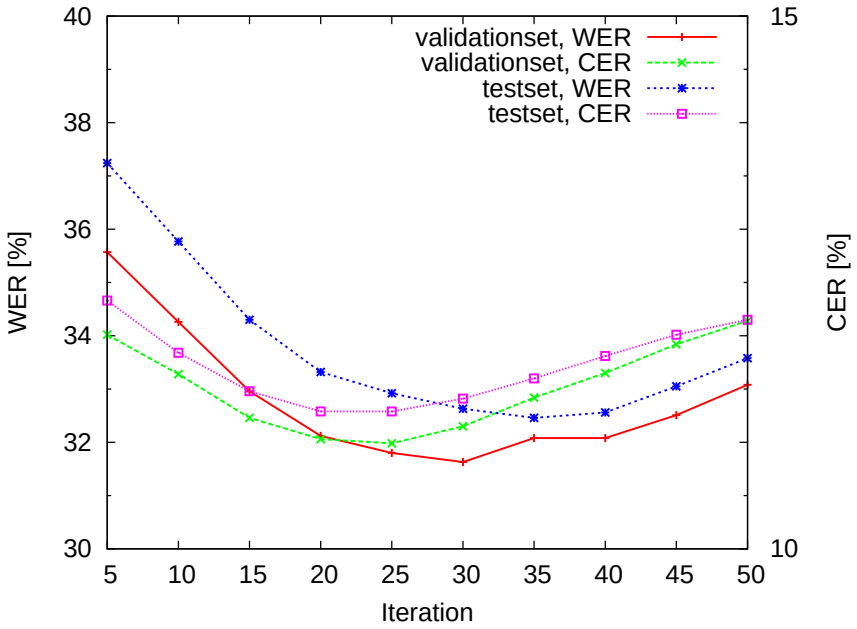


Figure 5.3. Plot of the WER and CER of the discriminative training.

5.8 Discriminative Training

As it shows that the best results so far are achieved with the slices motion with PCA setup with window size 9 and 30 dimensions, in the next step a discriminative training on this setup is performed as described in [Dreuw & Heigold+ 09]. Figure 5.3 shows that the WER of the discriminative training decreases with increasing number of Rprop iterations, while the minimum CER is reached after 25 iterations which leads to the currently best known performance of the system. Regarding the WER, optimization leads to the best results with 30 iterations. Thus, we do experiments with 30 iterations on the test set and compare the results to other groups.

5.9 Results of Other Groups

In the RIMES 2 evaluation campaign two groups attended the task of word recognition. These groups are LITIS and Itesoft. To compare the results of this work, we follow the same evaluation protocol as described in [Grosicki & Carre+ 09]. Therefore, the best result of the validation session is used, which is the discriminative training. Performing a recognition on the test set with this setup leads to the results shown in figure 5.3.

Table 5.4. Modeling + Training Summary Table

Setup	val		test	
	WER	CER	WER	CER
intensity + UPV prepr + PCA	37.09	12.45	38.64	12.77
+ MLE	43.12	15.69	44.98	16.30
+ M-MMI	**31.63**	**11.15**	**32.63**	**11.41**
LITIS [Grosicki & Carre+ 09]			27.47	
ITESOFT [Grosicki & Carre+ 09]			36.01	

Table 5.5. Comparison of the results of RWTH-OCR with other groups. The best results are bold.

System	WER
LITIS	**27.47**
RWTH-OCR	32.63
Itesoft	36.01

Table 5.4 summarizes the results that were achieved on the intensity features. The table shows how the error rate developed under the use of different modelings and different training methods. The improvement achieved on the test set is similar to the improvement achieved on the validation set. While the MLE experiment leads to higher error rates compared to the experiment on intensity features with preprocessing and PCA reduction, the discriminative training improves the error rates significantly.

Compared to the results of the RWTH-OCR system, also the results of other groups are shown. Since those results are achieved on the test set of the RIMES database, they are compared to our results on the same set.

We only test on the complete lexicon, such that the comparison limits to the task of recognition on the complete test lexicon. Table 5.5 shows the results of RWTH-OCR compared to the results of LITIS and Itesoft. LITIS achieves about 5 percent better results than RWTH-OCR, while RWTH-OCR is about 3 percent better than Itesoft.

Chapter 6

Conclusion

In this work, the RWTH-OCR system was adapted to French handwriting, based on the database of the RIMES 2 evaluation campaign.

At the beginning, experiments on the raw image data without preprocessing and without PCA estimation caused very high error rates, for intensity features as well as for Sobel features. Adding the UPV preprocessing and PCA estimation clearly improved the error rates. Thereby the results on the intensity features were better than the results on the Sobel features. All these results are in accordance to the results achieved by [Jonas 09] on English and Arabic handwriting.

In a further step a model length estimation (MLE) was applied to adapt the lexicon. Unexpectedly, this led to no further improvement of the error rates. In [Jonas 09] the MLE approach achieved the best error rates.

In contrast to the MLE experiment, a clear improvement was achieved by using a discriminative training with an M-MMI approach. With this approach the best results in this work were achieved. Also [Jonas 09] improved the error rates using discriminative training, though these error rates were higher than the error rates of the MLE experiment.

The results were compared to the results of other groups during this evaluation campaign. During the time of this work, it was not possible to outperform both of the competing system of the evaluation campaign. The RWTH-OCR lies in the middle of the range of the error rates of the LITIS and the Itesoft systems.

List of Figures

List of Tables

Bibliography

[Choisy 07] C. Choisy: Dynamic Handwritten Keyword Spotting Based on the NSHP-HMM. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pp. 242–246, Washington, DC, USA, 2007. IEEE Computer Society.

[Dreuw & Heigold$^+$ 09] P. Dreuw, G. Heigold, H. Ney: Confidence-Based Discriminative Training for Model Adaptation in Offline Arabic Handwriting Recognition. In *ICDAR*, pp. 596–600, 2009.

[Graves & Schmidhuber 08] A. Graves, J. Schmidhuber: Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. In *NIPS*, pp. 545–552, 2008.

[Grosicki & Abed 09] E. Grosicki, H.E. Abed: ICDAR 2009 Handwriting Recognition Competition. *Document Analysis and Recognition, International Conference on*, Vol. 0, pp. 1398–1402, 2009.

[Grosicki & Carre$^+$ 09] E. Grosicki, M. Carre, J.M. Brodin, E. Geoffrois: Results of the RIMES Evaluation Campaign for Handwritten Mail Processing. In *ICDAR '09: Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pp. 941–945, Washington, DC, USA, 2009. IEEE Computer Society.

[Heigold & Deselaers$^+$ 08] G. Heigold, T. Deselaers, R. Schlüter, H. Ney: Modified MMI/MPE: a direct evaluation of the margin in speech recognition. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pp. 384–391, New York, NY, USA, 2008. ACM.

[Jonas 09] S. Jonas: Improved Modeling in Handwriting Recognition. Master's thesis, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany, Jun 2009.

[Juan & Toselli$^+$ 01] A. Juan, A.H. Toselli, J. Domnech, J. González, I. Salvador, E. Vidal, F. Casacuberta: Integrated Handwriting Recognition and Interpretation via Finite-State Models. *Int. J. Patt. Recognition and Artifial Intelligence*, Vol. 2004, 2001.

[Kessentini & Paquet$^+$ 08] Y. Kessentini, T. Paquet, A. Benhamadou: A Multi-Stream HMM-based Approach for Off-line Multi-Script Handwritten Word Recognition. In *ICFHR*, 2008.

[LeCun & Chopra+ 07] Y. LeCun, S. Chopra, M. Ranzato, F.J. Huang: Energy-Based Models in Document Recognition and Computer Vision. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pp. 337–341, Washington, DC, USA, 2007. IEEE Computer Society.

[Ney 90] H. Ney: Acoustic Modeling of Phoneme Units for Continuous Speech Recognition. In *5th European Signal Processing Conference, Signal Processing V: Theories and Applications*, pp. 65–72. Elsevier Science Publishers, Dec 1990.

[Rybach 06] D. Rybach: Appearance-Based Features for Automatic Continuous Sign Language Recognition. Master's thesis, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany, Jun 2006.

[Sun & Si 97] C. Sun, D. Si: Skew and Slant Correction for Document Images Using Gradient. In *In Proc. of the 4th International Conference on Document Analysis and Recognition*, pp. 142–146, 1997.

[Yanikoglu & Sandon 98] B. Yanikoglu, P.A. Sandon: Segmentation of Off-Line Cursive Handwriting Using Linear Programming. *Pattern Recognition*, Vol. 31, pp. 1825–1833, 1998.