Analysis of Preprocessing Techniques for Latin Handwriting Recognition

Hendrik Pesch, Mahdi Hamdani, Jens Forster and Hermann Ney
Human Language Technology and Pattern Recognition Group
Chair of Computer Science 6
RWTH Aachen University, D-52056 Aachen, Germany
{pesch,hamdani,forster,ney}@i6.informatik.rwth-aachen.de

***Abstract*—In this work we analyze the contribution of preprocessing steps for Latin handwriting recognition. A preprocessing pipeline based on geometric heuristics and image statistics is used. This pipeline is applied to French and English handwriting recognition in an HMM based framework. Results show that preprocessing improves recognition performance for the two tasks. The Maximum Likelihood (ML)-trained HMM system reaches a competitive WER of 16.7% and outperforms many sophisticated systems for the French handwriting recognition task. The results for English handwriting are comparable to other ML-trained HMM recognizers. Using MLP preprocessing a WER of 35.3% is achieved.**

***Keywords*-Handwriting Recognition; Hidden Markov Models, Preprocessing**

I. Introduction

The recognition of unconstrained handwritten text is a challenging pattern recognition problem. While the recognition of machine printed text can be considered solved for Latin languages this is not the case for handwritten text. One of several challenges in the recognition of unconstrained handwritten text is the high variability in the text appearance caused by effects such as image noise from scanning or errors from a potential layout analysis but more important intrinsic properties such as differences in writing styles.

High inter-class variability poses a challenge to statistical text recognition systems forcing the development of robust features, robust classifiers or data preprocessing steps. Since it is difficult to design robust features for a high number of different writing styles and it is even more difficult to gain performance in the classification process that has already been lost in the features, data preprocessing and normalization are one of the most convenient methods to improve recognition performance [1].

Many preprocessing pipelines follow the steps proposed in [2]. Those steps include image cleaning, slant correction, slope correction and size normalization. However, the methods to perform those steps differ from system to system. Preprocessing in [2] is based on binary connected component analysis and is for example used in the SIEMENS HMM Recognizer [3]. In [4], it is proposed to estimate slant angles by computing the peak of a slant histogram. The histogram itself is computed by binning the magnitudes of the image gradient. In [5], it is observed that the slant correction angle is correlated to the variance of the vertical projection of an image. [5] also report that their slant correction algorithm performs better than gradient based methods on a database of handwritten Spanish words.

A recent approach by [6] uses Multilayer Perceptrons (MLPs) for preprocessing. Different MLPs are trained on manually preprocessed images to estimate local slant angles, to clean an image or to assign pixels to writing baselines. In contrast to the preprocessing mentioned before, the approach by [6] is model-based instead of heuristic-based. The major drawback of this approach is that a training database has to be prepared manually for each task.

Our contribution is to analyze heuristic-based methods for contrast normalization, noise removal, slant and skew correction as well as characters size normalization. These steps are analyzed individually and also as part of a preprocessing pipeline in the context of a state of the art statistical optical character recognition system using the RIMES and IAM databases. Furthermore, we compare the heuristic-based preprocessing pipeline to the MLP preprocessing by [6] on the IAM database.

This paper is organized as follows. Section II gives an overview of the heuristic-based preprocessing pipeline used and details individual steps. The used unconstrained handwriting recognition system is described in Section III. Experimental results as well as the RIMES and IAM databases are presented in Section IV. The paper is concluded in Section V.

II. Preprocessing

In the following we assume that all images are gray-scale images such as shown in Figure 1a.

The first concern with images of handwritten Latin text is the high variability in contrast between images of the same class but also within a single image itself due to scanning artifacts or different writing styles. Furthermore, some of the subsequent preprocessing steps depend on binary images (e.g. the size normalization). When contrast normalization is applied, we can use a simple fixed threshold to perform the binarization. Therefore, the preprocessing steps are only analyzed on contrast normalized images. The normalization works by mapping 70% of the lightest pixels to white and 5% of the darkest pixels to black. The gray-values of the other 25% are linearly scaled in between. Those values yielded good results in previous work [7]. An example of an image of contrast normalization is shown in Figure 1b.

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.179

The second concern is image noise such as salt-and-pepper noise. To reduce noise in the text images we apply a median filter in a window of 3×3 pixels centered on the current pixel to all images. An example is shown in Figure 1c. Although a visual inspection of the image before and after median filtering shows hardly any difference experimental results on the RIMES database show that preprocessing steps such as slant correction benefit from the noise removal.

Another aspect of handwritten text is the characteristic slant of the text. Every writer has a more or less characteristic slant in his handwriting leading to strong differences in the visual appearance of text between different writers. Consider for example the slant shown in Figure 1a and Figure 3. We use the slant correction method proposed in [5] and estimate the correction angle $\hat{\alpha}$ by

$$\hat{\alpha} = \underset{\alpha\in[-45,45]}{\operatorname{argmax}} f(I,\alpha) \quad .$$

The function $f(I,\alpha)$ computes the variance of the vertical projection of an image I that is sheared by angle α.

Another characteristic of different writing styles is the height of ascenders and descenders in relation to the main body of a written word. To reduce variability we perform a size normalization following the method proposed in [8]. The method works by scaling the ascender area and the descender area to a user defined percentage of the height of the main body.The ascender area is the region above the upper baseline and the descender area is the region below the lower baseline. The baselines are estimated by line fitting through the upper or lower contour of a word respectively.

Figure 1 shows the results of the whole preprocessing cascade applied to an example image taken from the RIMES database. While the contrast normalization of Figure 1b visually smooths the gray values of the image the effect of the median filter is hardly visible to the naked eye. After slant correction, shown in Figure 1d, the letters are almost upright which is especially visible at the now upright strokes of the characters "l". Finally, size normalization, see Figure 1e, reduces the uninformative blank space above the word.

Although the result of size normalization in Figure 1e is visually not appealing, the comparison of size normalized images in Figure 2 shows that size normalization helps in reducing inter-class variability.

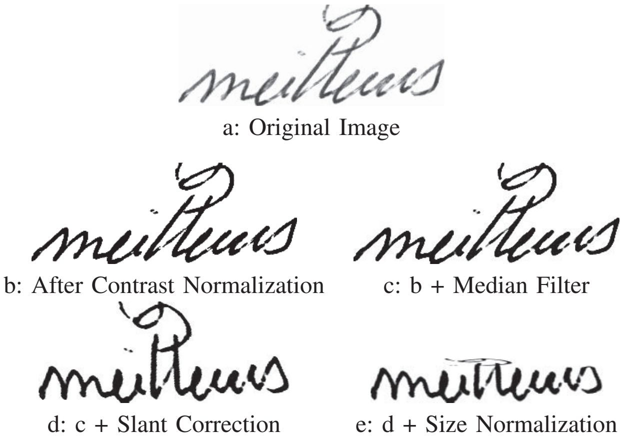
III. System Overview

The task of handwriting recognition is formulated as the search for the word sequence $w_1^N = w_1, \ldots, w_N$ for which the sequence of features $x_1^T = x_1, \ldots, x_T$ best fits trained statistical models. We maximize the posterior probability $p(w_1^N|x_1^T)$ over all possible word sequence w_1^N while the number of words N is unknown. Using Bayes decision rule, we formulate the decision process as a mapping of the feature sequence x_1^T to the optimal word sequence via decision function $\hat{w}_1^N(x_1^T)$

$$x_1^T \rightarrow \hat{w}_1^N(x_1^T) = \arg\max_{w_1^N}\{p(w_1^N)\cdot p(x_1^T|w_1^N)\}$$

where $p(w_1^N)$ is a m-gram language model (LM) and $p(x_1^T|w_1^N)$ is the visual model modeled using Hidden Markov Models whose emission probabilities are modeled by Gaussian Mixture Models (GMM) in turn. The parameters of the HMMs are learned using the Expectation-Maximization algorithm with maximum likelihood criterion and the covariance matrices are globally pooled and diagonal. HMMs are formulated on character level and are concatenated to super HMMs on word-level. We use the RWTH-OCR software package [9] for all experiments and left-to-right HMMs allowing loop and forward transitions as well as skips to the next but one state. Furthermore, HMMs are composed of segments with repetitions e.g. the first two states of an HMM have the same emission model, the third and forth state share the same emission model and so on.

Raw pixel values extracted from a window sliding in writing direction over an image are extracted as basic features and reduced to 30 dimensions by Principal Component Analysis. Please note that we focus on the impact of preprocessing on the recognition performance and not on optimizing the features themselves or the overall system parameters as a whole.


a: Original Image

b: After Contrast Normalization

c: b + Median Filter

d: c + Slant Correction

e: d + Size Normalization

Fig. 1: Example of the preprocessing pipeline on an image of the RIMES database

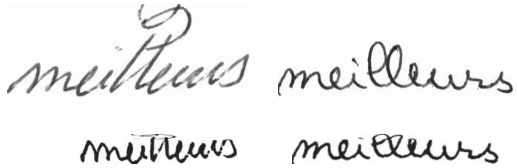


Fig. 2: Two class samples after size normalization from the RIMES database. Top row: original images, bottom row: images after size normalization

IV. Experimental Results

Two databases are used for the experiments in this work: The RIMES database is used for the evaluation of the steps in the proposed preprocessing cascade only and the IAM database is used to compare the heuristic-based preprocessing to preprocessing based on MLPs. Example images for both databases are shown in Figure 3.

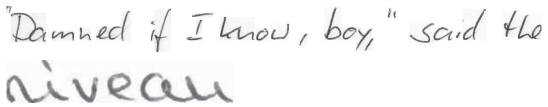


Fig. 3: Example images from the IAM database (top) and the RIMES database (bottom)

TABLE I: RIMES: Results for combination of different preprocessing steps

Preprocessing	WER [%]	CER [%]
Original (baseline system)	58.4	45.1
+ Contrast Normalization	53.2	37.9
- + Slant Correction	49.2	34.1
- - + Size Normalization	57.2	37.8
Original		
+ Contrast Normalization		
- + Size Normalization	63.0	44.8
Original		
+ Contrast Normalization		
- + Median Filter	55.4	39.4
- - + Slant Correction	**47.4**	33.4
- - - + Size Normalization	50.3	**32.5**
Original		
+ Contrast Normalization		
- + Median Filter		
- - + Size Normalization	62.6	44.9

The RIMES database [10] consists of handwritten letters in French, written by more than 1300 different writers. We will focus on the task of isolated word recognition that was also a part of the ICDAR 2009 Handwriting Recognition Competition [11]. The training set consists of 51738 words and the test set of 7464 words. The vocabulary size is 5334 words and is closed over the training and test set.

The IAM database [12] consists of handwritten transcriptions of texts from the Lanchester-Oslo-Bergen (LOB) corpus, cropped to text lines. The texts have been written by 657 different writers. The training set contains 6161 text lines, the validation set contains 920 text lines and the test set contains 2781 text lines. In contrast to the RIMES database the text line recognition is an open vocabulary task.

The performance of a system is measured using the Character Error Rate (CER) and the Word Error Rate (WER):

$$ER = \frac{\#\text{substitutions} + \#\text{insertions} + \#\text{deletions}}{\#\text{referencecount}}$$

For the CER the number of substitutions, insertions and deletions are computed on character level, for the WER on word level.

A. Analysis of preprocessing cascade on RIMES database

For the analysis of the preprocessing cascade we use the same simple setup for every step. The input images are scaled to a common height of 16 pixels, the sliding window has a width of 9 pixels. The HMM for a single character consists of six states, which share three GMMs. No language model is used. The results can be found in Table I.

As we can see, the full preprocessing cascade reduces the WER by 8.1% absolute. The contrast normalization makes a contribution of almost 5% absolute. The Median Filter makes the error rates worse in the short term. But leaving it out makes the slant correction and the size normalization perform worse. This indicates that the methods for slant correction and size normalization are sensitive to noise.

TABLE II: RIMES: Comparison of Slant Correction and Size Normalization, using scaling height 48

Preprocessing	WER [%]	CER [%]
Original	44.8	32.7
+ Contrast Normalization		
- + Median Filter		
- - + Slant Correction	37.0	24.6
- - - + Size Normalization	**31.0**	**15.3**

The gain in performance given by the slant correction is remarkable. Without the Median Filter the WER of the baseline is improved by 9.2% absolute. Applying the Median Filter further improves this to 11% absolute and gives the best WER for this setup.

The size normalization, however, increases the WER in all cases. Using only contrast normalization and size normalization results in a WER that is even worse than the baseline. Additionally applying the Median Filter improves the WER only slightly. At first glance this might indicate that size normalization is useless and even harmful. But when size normalization is applied on top of the full preprocessing cascade, the CER decreases even though the WER increases. This is due to a different error distribution and we can hope that with additional tuning the size normalization actually improves our recognition results. Since size normalization changes the average height of the images and thus the aspect ratio and the number of observations per image, the scaling height is a natural candidate for tuning.

Several scaling heights were analyzed keeping the aspect ratio fixed. The average width of the character increases by increasing the height. To compensate for this, the number of HMM states and the size of the sliding window are adjusted accordingly. We found that a scaling height of 48 pixels works best. A recognition experiment with this parameter change is performed using again the original images, the preprocessed images without size normalization and for the fully preprocessed images. The window size is set to 18 pixels and 22 HMM states with 11 different GMMs are used. Table II lists the results.

By increasing the scaling height, the errors are reduced on both the original data and on the two analyzed steps in the preprocessing cascade. The inclusion of size normalization leads to the best error rates now. This shows that size normalization is also a useful step when coupled with a good choice for the scaling height.

Next we verify that the performance gain achieved by preprocessing carries over to fully tuned systems. For this we evaluate several parameter changes in the system using the original data and in the system with the fully preprocessed data.

The tuning parameters include the number of states in the HMMs as well as the number of densities in the GMMs. A more detailed description of the tuning is given in [13].

The performance of the tuned system is further enhanced

TABLE III: RIMES: Comparison of the best systems on original and preprocessed data

Setup	WER [%]	CER [%]
Original	32.4	22.4
+Language Model	29.1	22.2
Preprocessing	22.9	11.2
+Language model	**16.7**	**8.3**

TABLE IV: RIMES: Comparison with results of the ICDAR 2009 handwriting competition [11]

System	WER [%]
RWTH	16.7
TUM	9.0
UPV	16.8
ParisTech(1)	23.7
IRISA	25.3
SIEMENS	26.8
ParisTech(2)	32.0
LITIS	33.4
ParisTech(3)	41.3
LSIS	47.6
ITESOFT	49.6

by introducing a unigram language model. The unigram is trained on the words of the training set with a perplexity of 45.2. To give non-zero probability to words that are not seen during training Kneser-Ney smoothing is used. A comparison of the systems is given in Table III. We can see that the best preprocessing system still outperforms the best system on the original data.

In Table IV we compare the results of the best system to the performance of other state-of-the-art systems, which participated in the ICDAR 2009 Handwriting Recognition Competition [11].

We can see that only one of the participating systems performs better than our system. This is remarkable since our system is relatively simple. It only uses preprocessing, tuned Gaussian HMMs and a unigram language model and performs already better than more sophisticated methods like HMM/ANN hybrids (UPV) or system combinations (SIEMENS).

B. Comparison with MLP preprocessing on IAM database

In the following a comparison is made between the preprocessing pipeline that was already used on the RIMES database and the MLP preprocessing that was recently introduced in [6] and explained in more detail in the introduction section. Unfortunately we were only given the MLP preprocessed images of the IAM database and not the software, thus we can do this comparison only on the IAM database. Our original preprocessing cascade was only slightly changed. To compensate for some artifacts in the IAM database, the contrast normalization parameters are changed. 90% of the lightest pixels are now mapped to white, 7% of the darkest pixels are mapped to black. Furthermore we introduce a slope correction step, to make the proposed preprocessing pipeline more similar to the MLP preprocessing. The slope correction corrects the angle of the lower baseline to the lower image margin. The baseline is estimated using the method described in Section 2.

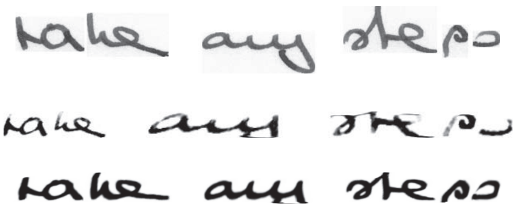


Fig. 4: Comparison of preprocessing approaches. Top: Original; Middle: heuristic-based preprocessing; Bottom: MLP preprocessing

TABLE V: IAM: Comparison of the proposed preprocessing with MLP preprocessing

	Validation		Test	
Preprocessing	WER	CER	WER	CER
This work	35.0	16.9	46.6	26.6
MLP	**27.9**	**11.3**	**35.3**	**17.0**

A visual comparison of the two approaches of preprocessing is made in Figure 4. This comparison already indicates that the MLP preprocessing is superior. The local decisions that are made by the image cleaning MLP map for example the pixels of the vertical stroke of the character ”p” to black. The global contrast normalization however maps this to a light gray, which is not consistent with the rest of the character. This has a negative influence on the subsequent preprocessing steps.

We perform a recognition experiment to confirm the hypothesis that the MLP preprocessing is superior. The setup is again the same for both kinds of input images. The scaling height is set to 16 pixels and the window size is 7 pixels. The HMMs for each character have 10 states, which share 5 different GMMs. A trigram language model is used during recognition. The LM is trained on three different text corpora, the LOB corpus, the Brown corpus and the Wellington corpus with a perplexity of 233.749 for the validation set and 258.7 for the test set. Kneser-Ney smoothing is again used. The lexicon contains the 50k most often occurring words of these corpora. The OOV-rate using this lexicon is 3.77% for the validation set and 3.45% for the test set. The results for this setup are listed in Table V.

As suspected, the MLP preprocessing is indeed better. The difference in WER is 7.1% absolute on the validation set and 11.3% on the test set. Table VI shows a comparison of the two systems with results reported in the literature. All of the listed state-of-the-art results perform better, but it should be noted that only preprocessing and no further tuning was done on the IAM database. Also all of the listed HMM systems use some kind of preprocessing, which indicates that preprocessing is helpful even if a sophisticated modelling approach is used. It is also remarkable that our system outperforms the ML baseline in [6] using the same kind of preprocessing.

V. Conclusion

We have seen that preprocessing is an easy way to gain recognition performance. All of the steps in our preprocessing cascade proved to be useful: contrast normalization and

TABLE VI: IAM: Comparison with state-of-the-art results; all numbers are WER[%]

System	Validation	Test
This work	35.0	46.6
MLP preprocessing	27.9	35.3
MLP features + HMM [14]	22.7	28.8
MLP preprocessing + HMM [6]	32.8	38.8
MLP preprocessing + HMM/ANN [6]	19.0	22.4
HMM ensemble [15]	26.8	32.8
Recurrent Neural Net [16]	-	25.9

slant correction ameliorate the performance without further parameter tuning. The Median Filter helps to reduce noise that influences subsequent preprocessing steps. Size normalization reduces the error further, but has to be coupled with a good choice for the scaling height.

We have also seen that MLP preprocessing outperforms the heuristic approach in preprocessing. However, this comes at the cost of manually preprocessing images as training data for the MLPs (including manual deslanting, manual baseline selection and manual image cleaning). It is inefficient to do this for every new database and it is not yet clear if MLPs that are trained on one database perform good on others. Future research could thus deal with further improvement of heuristic preprocessing methods. It is also probable that more powerful classifiers profit from the preprocessing. Applying discriminative training should thus result in better and more competitive error rates.

Acknowledgment: We would like to thank the Department of Information System and Computation at the "Universitat Politècnica de València" for both providing preprocessing tools and the MLP preprocessed images of the IAM database.

References

[1] A. L. Koerich, R. Sabourin, and C. Y. Suen, "Large vocabulary off-line handwriting recognition: A survey," *Pattern Analysis and Applications*, vol. 6, pp. 97–121, 2003.

[2] T. Caesar, J. Gloger, and E. Mandler, "Preprocessing and feature extraction for a handwriting recognition system," in *Proceedings of the Second International Conference on Document Analysis and Recognition*, Oct 1993, pp. 408 –411.

[3] M.-P. Schambach, "Automatische modellierung gebundener handschrift in einem hmm-basierten erkennungssystem," Ph.D. dissertation, Fakultät für Informatik, Universität Ulm, Ulm, Germany, 2004.

[4] B. Yanikoglu and P. A. Sandon, "Segmentation of off-line cursive handwriting using linear programming," *Pattern Recognition*, vol. 31, pp. 1825–1833, 1998.

[5] M. Pastor, A. Toselli, and E. Vidal, "Projection profile based algorithm for slant removal," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 3212, pp. 183–190.

[6] S. Espana-Boquera, M. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 99, pp. pre–print, 2010.

[7] P. Roeder, "Adapting the rwth-ocr handwriting recognition system to french handwriting," Master's thesis, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen. Germany, Dec 2009.

[8] A. Juan, A. H. Toselli, J. Domnech, J. Gonzàlez, I. Salvador, E. Vidal, and F. Casacuberta, "Integrated handwriting recognition and interpretation via finite-state models," *International Journal of Pattern Recognition and Artifial Intelligence*, vol. 2004, pp. 519–539, 2001.

[9] P. Dreuw, D. Rybach, G. Heigold, and H. Ney, *RWTH OCR: A Large Vocabulary Optical Character Recognition System for Arabic Scripts*. London, UK: Springer, Apr. 2011, ch. Part I: Development, p. To Appear. [Online]. Available: http://www-i6.informatik.rwth-aachen.de/rwth-ocr/

[10] E. Augustin, M. Carré, G. E., J. M. Brodin, E. Geoffrois, and F. Preteux, "Rimes evaluation campaign for handwritten mail processing," in *Proceedings of the Workshop on Frontiers in Handwriting Recognition*, 2006, pp. 231–235.

[11] E. Grosicki and H. El Abed, "Icdar 2009 handwriting recognition competition," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, 2009, pp. 1398–1402.

[12] U.-V. Marti and H. Bunke, "A full english sentence database for off-line handwriting recognition," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, ser. ICDAR '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 705–708. [Online]. Available: http://dl.acm.org/citation.cfm?id=839279.840504

[13] H. Pesch, "Advancements in latin script recognition," Master's thesis, RWTH Aachen University, Aachen, Germany, Nov. 2011.

[14] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney, "Hierarchical hybrid MLP/HMM or rather MLP features for a discriminatively trained gaussian HMM: a comparison for offline handwriting recognition," in *IEEE International Conference on Image Processing*, Brussels, Belgium, Sep. 2011. [Online]. Available: http://www-i6.informatik.rwth-aachen.de/rwth-ocr/

[15] R. Bertolami and H. Bunke, "Hidden markov model-based ensemble methods for offline handwritten text line recognition," *Pattern Recognition*, vol. 41, no. 11, pp. 3452–3460, Nov. 2008.

[16] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, May 2009.