2009 10th International Conference on Document Analysis and Recognition

How to Improve a Handwriting Recognition System

Haikal El Abed, Volker Märgner
Technische Universitaet Braunschweig,
Institute for Communications Technology (IfN),
Schleinitzstrasse 22, 38106 Braunschweig, Germany
{elabed, v.maergner}@tu-bs.de

Abstract

The recognition of handwritten characters, words, and text arouses great interest today. To develop the best working system is subject of many papers published. With this paper, methods to improve the performance of existing word recognition systems are discussed. The availability of a sufficient data sets for training and testing the system assumed, optimization algorithms are presented. The usage of different feature sets and the combination of different recognizers are proposed. Tests with Arabic handwriting recognition systems using the reference IfN/ENIT-database show the usefulness of the proposed methods. An improvement of the recognition rate of up to 28% *of the best single system is achieved.*

1. Introduction

Handwritten character and word recognition systems have achieved a considerable improvement over the past years. Many different methods and algorithms to recognize isolated handwritten digits [17], characters [14, 15], and whole words [6] have been developed and tested for many different languages.

Usually a character or word recognition process is presented as a sequence of different steps. It begins with a pre-processing step, including segmentation and noise reduction, followed by a feature extraction used for the calculation of a description of a word. The extraction of these features is a difficult task with two aims: firstly, to identify the relevant features, and secondly, to find all of them. This step is a source of recognition errors as well as a potential research filed for improving the performance of classification methods.

The subsequent step in the recognition process is the classifier itself. The problem of recognizing a handwritten word as a whole can be considered as a sequence of decisions where feature vectors are grouped into smaller "decision units", e.g. characters, and sequentially recognized. The sequence of these "decision units" represents the unknown word. To solve such a recognition problem, different classification methods are used. Using more than one classifier may increase the ambiguity by choosing an appropriate classifier for a given application. However, we show in the second part of this paper, how to use the availability of different classifiers to implement a new high-performance classifier.

Some other possible processing steps which can be used to improve the recognition process are: The realization of a system without the explicit use of a lexicon and the integration of post-processing modules, like linguistic methods. The selection of training and testing data is also a very important task. The data must be relevant to the task and sufficient to train all parameters of the classifier, accompanied by another data set to test the quality of the realized system. Standard and available databases are one important reason to validate and compare recognition systems [13].

In this work how to improve a handwriting recognition system using different simple and state-of-the-art feature sets and combining different classification approaches is discussed. Section 2 provides a description of four feature extraction algorithms used in the comparisons of recognition performance experiments. In Section 3, some basic definitions of different combination rules are presented. Section 4 gives a description of used data, information about the classifiers which are used for the combination tests, and a detailed discussion of the test results. Concluding remarks are presented in Section 5.

2. Feature Extraction

The feature extraction step selects and prepares data which is used by a classifier to achieve the recognition task. In our case, we use Hidden Markov Models (HMMs), which require temporal information about the input data. This information is unavailable in images of handwritten words to be recognized, so it must be simulated. A common method

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.11

is the sliding window technique for the extraction of the off-line features. The window scans the input image from right to left or left to right depending on the writing direction. The size of the window and the overlap between consecutive windows are system parameters. This technique of feature extraction is typically used in HMM based systems for the recognition of off-line text and is also employed for multiple types of off-line features.

Three off-line methods of feature extraction and one on-line method are used in this work. The choice of these features is based on a state-of-the-art study. The different features were used successfully in different applications in handwriting recognition.

2.1. Pixel Values

The pixel values were used in [3] in conjunction with a word normalization method. We retrieve these values from normalized images and apply a Karhunen-Loéve-Transformation (KLT) to reduce the number of features used by the system. We use 150 values chosen from the initial vector, composed of 360 values. These features are computed from a window with a width of 5 pixels and a 3 pixel overlap of consecutive windows.

2.2. Densities and Moment Invariants

The density of black pixels is calculated by a re-sampling procedure, i.e. the window is divided into cells and the density value of the black pixels of each cell is used. Additionally, the moment invariants of Hu [8] are calculated for each window and concatenated with the density values. The feature vector is composed of 92 values (density and moment invariants) computed from an 8 pixel window size (without overlap).

2.3. Pixel Distribution and Concavities

This feature extraction approach [1] uses a sliding window to calculate pixel distribution and concavities of a word image. The extracted features are classified into the following types: distribution features based on foreground (black) pixel densities, and concavity features. The result of the feature extraction is a 24-feature vector per frame, where 16 features are from pixel distribution, and the remaining 8 present the concavity properties of each frame.

2.4. On-line Features

The use of on-line features for off-line handwriting recognition is possible even without using methods for the recovery of the temporal information from the off-line image. Based on the algorithm described in [5] and [2], we use the extracted features to model a handwritten word on the basis of the beta-elliptic approach, scanning the word image from left to right or from right to left (e.g. writing direction of Arabic script). After the transformation of the input image into a sequence of coordinates, each part of connected word (PAW) is segmented into a sequence of graphemes. Resulting from these of these steps is a 21 feature vector [10] related to beta-elliptic properties of each grapheme.

3. System Combination

Combination methods are organized in three main classes based on the output type of the classifiers, abstract level, ranked list of classes, and measurement level outputs [11].

The output of a system $S_{i,j}$, given a sample word x_k, consists of an ordered sequence of m pairs of values composed of the system output word $y_{i,j}(x_k)$ together with its confidence value $w_{i,j}(x_k)$, where $S^{(1)}_{i,j}(x_k) = y_{i,j}(x_k)$ and $S^{(2)}_{i,j}(x_k) = w_{i,j}(x_k)$. The function $S_{i,j}(x_k)$ is defined as follows:

$$S_{i,j}(x_k) = \{(y_{i,j}(x_k), w_{i,j}(x_k))\} \tag{1}$$

where $i \in \{1, 2, \ldots, n\}$ is the index of a recognition system, $j \in \{1, 2, \ldots, m\}$ is the index of the j-best output, and $k \in \{1, 2, \ldots, N\}$ is the index of a word in the dataset.

3.1. Data and Systems Analysis

To define the upper limits of recognition rates based on the existing data and classification architecture, we analyse the behavior of the systems on the training data. For this step we use the Bayes algorithm (Eq. 2) to calculate the a posteriori probabilities of each input class. Each input word from the test data is assigned to the class with the maximum a posteriori probability (Eq. 3 for $i = 1, \ldots, n$).

$$\begin{gathered} P(w_{i,j}(x_k)|S_{1,j}, \ldots, S_{n,j}) = \\ \frac{P(S_{1,j}, \ldots, S_{n,j}|w_{i,j}(x_k))P(w_{i,j}(x_k))}{P(S_{1,j}, \ldots, S_{n,j})} \end{gathered} \tag{2}$$

$$\max_i P(S_{1,j}, \ldots, S_{n,j}|w_{i,j}(x_k)) \tag{3}$$

3.2. Abstract level

Voting Methods In this paper, two forms of voting rules are used. The first method is based on the simple majority voting strategy. This method does not use any confidence level but only counts how many systems respond with the same output to an input image. The second method is the weighted majority voting (W_{Mv}). This method uses classification confidences (Eq. 3.2) to overcome the problem of unbalanced or dependent classification systems.

$$W_{Mv}(x_k) = \max_{S_{l,1} \in \{S_{1,1},\ldots,S_{n,1}\}} (\sum_i w_{i,1}(x_k))$$

3.3. Rank level

Borda Count (Bc) The combination method Borda is adapted to pattern classification problems [9]. Each system $S_{i,j}$ is considered as a voter and the result classes are the candidates. The basic idea of the Borda combination method is to use the ranking information (the r-best results from the entire result list) to come to a decision, not just the first best results of each system. It also returns a complete ranked list of the possible results.

$$Bc(x_k) = \max_{\{S_{l,j}, w_{l,j}\}} (\sum_{i=1}^{n} (r - rank(S_{i,j}(x_k), w_{i,j}(x_k)) + 1))$$

For each result in the r-best result lists, the value "rank of the result+1" is assigned.

Rank Count (Rc) and Modified Rank Count (M_{Rc}) The basic idea of the rank-based method [7] is to attribute a cost function c_i to each classification system. In addition to the cost function, a system confidence value a_i is assigned to each system. This system confidence can be used as a general rank function for the different systems. The rank count method is given by the following equation:

$$Rc(x_k) = \max_{\{S_{l,j}, w_{l,j}\}} (\sum_{i=1}^{n} (a_i + c_i(rank(S_{i,j}(x_k), w_{i,j}(x_k)))))$$

In this paper two forms of this method for the combination of different systems are used. The first one is based on the definition of the rank count method (Rc). For each class the corresponding weight as output value of the cost function c_i is assigned. In the second method (M_{Rc}) [4], the cost function c_i is defined as a product of rank and weight of each word image x_k. In both methods a system confidence $a_i = 0$ was choused.

3.4. Measurement level

Maximum, Minimum and Median Rules The maximum and minimum combination rules are defined in the equations 4. The resulting decisions represent the highest (for the maximum rule, and the lowest for minimum rule) sum of confidences $\bar{w}_s$ (s represent the number of different words in the output list all classifiers) of the same word output of all results in the j-best output list and from all n classifiers. For the $Median$ rule,sum of confidences $\bar{w}_s$ are sorted by value and the x_k.

$$Max(x_k) = \max_{l=1}^{n} \bar{w}_s(x_k) \; ; Min(x_k) = \min_{l=1}^{n} \bar{w}_s(x_k) \quad (4)$$

Sum and Product Rules The sum and product combination methods presented in equations 5 are comparable to the definition of maximum and minimum rules:

$$Sum(x_k) = \sum_{l=1}^{n} \bar{w}_s(x_k) \; ; Prod(x_k) = \prod_{l=1}^{n} \bar{w}_s(x_k) \quad (5)$$

Table 1. Recognition rates in % of the HMM recognizer using different feature extraction methods on sets d and e.

Method	set d	set e
Pixel Values	**86.28**	**63.90**
Densities and Moment Invariants	83.86	51.57
Pixel Distribution and Concavities	67.68	49.48
On-line Features	81.21	50.01

4. Tests and Results

The proposed methods were tested with handwritten Arabic words using the IFN/ENIT-database (v2.0p1e) [16]. This database contains 32492 images of Arabic handwritten words. The database is divided in 5 sets (a-e).

4.1. Systems

Feature Extraction We tested the different feature sets using the HMM Toolkit (HTK) for the implementation of the recognition system. The same recognition system is used for the different feature sets and the systems were trained and tested using the IfN/ENIT database. The codebook size is 256 elements for all the systems. We use right to left discrete HMMs. Table 1 shows recognition results for the systems with different feature sets. In the first and second columns, dataset d results, which are part of the training data, and test data results from set e are shown, respectively.

System Combination The ICDAR 2007 competition [12] compared 14 Arabic handwritten word recognition systems submitted from 9 groups (some groups delivered more than one system). Table 2 shows the results with the new data from set f. The results of 11 systems are combined using the following schema: the best two (on set f) systems are combined first, then weaker performing systems are added

Table 2. Recognition rates of individual recognizer in % using the test set f.

ID	top 1	ID	top 1	ID	top 1	ID	top 1
01	61.70	07	82.77	10	81.65	13	81.47
05	59.01	08	**87.22**	11	81.93	14	80.18
06	83.34	09	79.10	12	81.81		

successively. Ten different combination possibilities (combining 2, 3, ..., 11 systems) are tested and evaluated using variations of combination methods.

Due to the fact that our tests were constructed with classifiers developed independently by different research groups, the confidences for the recognition results vary in the values (max, min, steps) and weights. For this reason, confidence values of different systems have to be normalized to make them comparable. The new confidence value $w_{i,1}^{norm}(x_k)$ for a recognized word image x_k is calculated based on the normalized difference of the highest and lowest confidence in a test set with N word images. With a system $S_{i,j}$, a sample word image x_k, and its original confidence $w_{i,j}^{orig}(x_k)$, the new confidence $w_{i,j}^{norm}(x_k)$ is defined by using the following equation:

$$w_{i,j}^{norm}(x_k) = \frac{w_{i,j}^{orig}(x_k)}{\max\limits_{l \in \{1,\ldots,N\}}(S_{i,j}^{(2)}(x_l)) - \min\limits_{l \in \{1,\ldots,N\}}(S_{i,j}^{(2)}(x_l))}$$

4.2. Results and Discussion

Feature Extraction The goal of the performed tests with different feature extraction methods is to point out which combination of feature sets can be used to decrease recognition error rates. Table 1 presents the recognition results of the single recognizer. The system using the feature extraction method based on pixel features performs at least 12% better than the other systems on the test set e. Table 3 shows the results of the application of different combination methods on the 4 classification systems. The modified rank count method (M_{Rc}) yields the best results in combining different systems. In comparison to the individual recognition rates, an improvement of about 28% is observed (increase of recognition rate from 63.90% to 81.93%).

System Combination The first column of Table 4 shows as a kind of upper bound the percentage of word images out of the test set f, which were recognized correctly by at least one of the combined recognizers (or-case). A theoretical upper bound generated with the Bayes analysis (section 3.1), is calculated in a first step. Experimental results have shown, that the analysis of the training results using Bayesian rules can give a first idea about the limits of the used features, classifiers, and combination rule.

In each row of table 4 the highest recognition is displayed in bold digits. The recognition results of the weighted majority voting (W_{Mv}) and the Borda count (Bc) methods are not better than the other methods in any of the combination possibilities. The rank count method in the first version (Rc) gives a best result in the case of combination of 5 systems as well as combination of 3, 4, and 6 systems. The methods Max, Sum, and $Prod$ yield good results combining multiple systems. The methods Min and $Median$ are appropriate for combining 2 or 3 systems.

The second rank count method (M_{Rc}) yields the best results in combining 2, 7, 8, 9, 10, and 11 systems. The M_{Rc} method also gives the overall best result with a recognition rate of 94.71% combining all 11 systems. Additionally, the zigzag effect observed in previous works, caused by a combination based on majority voting of odd and even numbers of classifiers, can be observed in the results of the combination of a different number of classifiers with different combination rules (see table 4). The modified rank method M_{Rc} seems to be more robust and even all 11 systems can be combined with an overall better recognition rate. Comparing these results with the best performing single system, an increase in recognition rate from 87.22% to 94.71% is achieved (improvement of the recognition rate of about 8.5%).

5. Conclusions

In this paper we addressed the problem of improving the performance of handwriting recognition systems by proposing the use of different feature extraction methods and different classifier combination approaches. It was found that combining the same classifier four times using different feature sets improves the system performance by 18% compared to the best single recognizer. A better recognition rate could be reached with the combination of 11 different systems (different classifier and different set of features). Comparing the different combination methods our test showed that the modified rank count performed best. It may be more effective to combine systems using the same classifier and different feature sets instead of searching for the one and optimal feature set. Combining the results of recognition systems, different in feature sets and classification module, may result in further system improvement. An important precondition of this result is the training of the systems with a data set, large enough to represent the variance of real world data and also large enough to train all parameters of the classifiers.

Table 3. Recognition rates in % using different combination methods on sets d and e.

	OR	S_{Mv}	Bc	W_{MV}	Rc	M_{Rc}	Max	Min	$Median$	Sum	$Prod$
training set: set d											
2	94.94	75,69	91.54	87.75	**93.20**	92.75	83.86	86.77	87.75	87.75	87.75
3	97.62	78,84	95.93	93.13	96.11	**96.78**	91.95	78.69	83.80	93.13	93.13
4	98.00	63,99	95.92	94.34	95.90	**96.97**	88.29	56.84	83.19	94.34	94.33
test set: set e											
2	76.83	38,67	69.04	64.86	70.20	**70.38**	51.60	63.90	64.86	64.86	64.86
3	84.19	42,57	76.30	71.03	74.99	**77.62**	63.83	51.55	56.12	71.03	71.03
4	87.57	35,47	80.72	75.20	78.37	**81.93**	65.09	31.05	57.58	75.20	75.19

Table 4. Recognition rates in % of different combination methods on set f.

	OR	S_{Mv}	Bc	W_{Mv}	Rc	M_{Rc}	Max	Min	$Median$	Sum	$Prod$
2	93.23	77,33	67.15	89.11	90.82	**91.02**	83.34	87.22	89.11	89.11	89.11
3	93.71	82,29	58.00	88.71	**90.65**	90.36	87.30	81.94	85.39	88.71	88.66
4	95.21	76,37	61.40	89.22	**91.55**	91.10	90.08	69.27	85.41	89.22	89.70
5	95.43	79,78	73.80	89.46	**91.90**	91.81	87.05	69.51	82.00	89.46	90.61
6	95.50	79,36	81.77	89.76	**92.04**	91.90	84.81	68.61	82.40	89.76	91.54
7	96.19	77,58	86.80	91.13	92.97	**92.99**	87.74	61.04	80.64	91.13	92.48
8	96.52	75,10	90.47	92.73	93.83	**94.00**	91.67	52.57	82.71	92.73	93.51
9	96.57	77,98	88.85	92.84	93.77	**93.99**	88.88	52.51	77.85	92.84	93.25
10	96.69	58,37	89.85	93.30	93.82	**94.49**	90.81	38.89	78.71	93.30	93.75
11	96.78	56,53	90.79	93.61	94.11	**94.71**	91.81	29.10	71.74	93.61	93.96

References

[1] R. Al-Hajj, L. Likforman-Sulem, and C. Mokbel. Arabic handwriting recognition using baseline dependant features and hidden markov modeling. In *8th Inter. Conf. on Document Analysis and Recognition*, pages 893–897, 2005.

[2] H. Boubaker, M. Kherallah, and A. M. Alimi. New strategy for the on line handwriting modeling. In *9th Inter. Conf. on Document Analysis and Recognition*, pages 23–26, 2007.

[3] H. El Abed and V. Märgner. Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten arabic words. In *9th Inter. Conf. on Document Analysis and Recognition*, pages 974–978, 2007.

[4] H. El Abed and V. Märgner. Improvement of Arabic handwriting recognition systems - combination and/or reject? In *Document Recognition and Retrieval XVI, 7247–10*, 2009.

[5] A. Elbaati, M. Kherallah, A. M. Alimi, and A. Ennaji. De l'Hors-Ligne Vers un Système de Reconnaissance En-Ligne : Application à la Modélisation de l'Écriture Arabe Manuscrite Ancienne. In *Semaine du document numrique, SDN (ANAGRAM)*, 2006.

[6] H. Fujisawa. A view on the past and future of character and document recognition. In *9th Inter. Conf. on Document Analysis and Recognition*, pages 3–7, 2007.

[7] S. Günter. *Multiple classifier systems in offline cursive handwriting recognition*. PhD thesis, Institute of Computer Science and Applied Mathematics (IAM), 2004.

[8] T. M. Hamdani and A. M. Alimi. *Artificial Neural Networks and Genetic Algorithms*, chapter β-SVM a new support vector machine, pages 63–68. Springer, 2003.

[9] T. K. Ho, J. Hull, and S. Srihari. Decision combination in multiple classifier systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(1):66–75, 1994.

[10] M. Kherallah, F. Bouri, and A. M. Alimi. On-line arabic handwriting recognition system based on visual encoding and genetic algorithm. *Engineering Applications of Artificial Intelligence*, 22(1):153–170, 2009.

[11] L. I. Kuncheva. *Combining Pattern Classifiers. Methods and Algorithms*. Wiley, 2004.

[12] V. Märgner and H. El Abed. ICDAR 2007 arabic handwriting recognition competition. In *Proceedings of the 9th Inter. Conf. on Document Analysis and Recognition*, volume 2, pages 1274–1278, 2007.

[13] V. Märgner and H. El Abed. *Arabic and Chinese Handwriting Recognition*, volume 4768, chapter Databases and Competitions: Strategies to Improve Arabic Recognition Systems, pages 82–103. Springer, LNCS, 2008.

[14] S. Mori, K. Yamamoto, and M. Yasuda. Research on machine recognition of handprinted characters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(4):386–405, 1984.

[15] G. Nagy. Advanced character recognition. In *6th Inter. Conf. on Document Analysis and Recognition*, pages 2–6, 2001.

[16] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri. IFN/ENIT-database of handwritten Arabic words. In *CIFED*, pages 127–136, 2002.

[17] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam. Computer recognition of unconstrained handwritten numerals. *Proc. IEEE*, 80(7):1162–1180, 1992.