A Modular System to Recognize Numerical Amounts on Brazilian Bank Cheques

L. S. Oliveira[1−3], R. Sabourin[1−3], F. Bortolozzi[1] and C. Y. Suen[3]
[1]PUCPR Pontifícia Universidade Católica do Paraná (PPGIA-LARDOC)
Rua Imaculada Conceição 1155, 80215-901 - Curitiba, PR - BRAZIL
{soares,fborto}@ppgia.pucpr.br

[2]ETS - Ecole de Technologie Supérieure (LIVIA)
1100, rue Notre Dame Ouest, Montreal, H3C 1K3, CANADA
sabourin@gpa.etsmtl.ca

[3]CENPARMI Centre for Pattern Recognition and Machine Intelligence
1455 de Maisonneuve Blvd. West, Suite GM 606 - Montreal, H3G 1M8, CANADA
suen@cenparmi.concordia.ca

Abstract

This paper presents a modular system to recognize numerical amounts on Brazilian bank cheques. The system uses a segmentation-based recognition approach and the recognition function is based on a Recognition and Verification strategy. Our approach consists of combining the outputs from different levels such as segmentation, recognition and post-processing in a probabilistic model. A new feature set is introduced to the verifier module in order to detect segmentation effects such as over-segmentation and under-segmentation. Finally, we present experimental results on two databases: numerical amounts and NIST SD19. The latter aims at validating the concept of modular system and showing the robustness of the system over a well-known database.

1 Introduction

Automatic processing of bank cheques has been a very popular task for handwriting recognition research. This is motivated by the availability of relatively inexpensive CPU power, and the possibility to reduce considerably the manual effort involved in this task. Comparated to a zip-code recognition problem, bank cheque systems have to take into account the large variability in the representation of a numerical amount, e.g., the number of components to be identified.

We present in this paper a modular recognition system for handwritten numerical amounts on Brazilian bank cheques. This system takes a segmentation-based recognition approach where an explicit segmentation algorithm determines the cut regions and provides a multiple spatial representation. This kind of system has to solve a crucial problem: distinguishing, at the recognition stage, a sequence corresponding to an inter-character segmentation from another relative to an intra-character segmentation. In order to deal with this problem we have used a strategy based on Recognition and Verification where the integration of all modules is done through a probabilistic model, which is inspired by information theory models [5].

The main contribution of this work is related to the verification module. We propose a new feature set, which takes into account multi-level concavity analysis and contextual information, in order to feed the verifier module. We present also the concept of modular recognition system, and we show how such a recognition system can deal with different applications. In order to validate such a concept and to evaluate the robustness of the system, we present some experiments on NIST SD19.

2 Probabilistic Model and Modular System

The goal of the probabilistic model is to define a function that combines all the system modules in order to allow a sound integration of all knowledge sources used to infer a plausible interpretation. The probabilistic model that we are using has been applied to speech recognition [6], handwritten word recognition [9] and handwritten digit recognition [3]. Such a model estimates the most probable interpretation of the written amount M (noted $\widehat{M}$). Its input corresponds to an image I after pre-processing and segmentation which provides a list of elementary sub-images. In a

Bayesian framework, $\widehat{M}$ is given by the maximum posterior probability:

$$P(\widehat{M}/I) = \max_{M} P(M/I) \quad (1)$$

Due to the restricted space that we have here, we present just the final equation (Equation 2) of the probabilistic model. A complete description can be found in [3].

$$P(M/I) \approx \prod_i \frac{P(v_i/[i_{s_{(i)}} \ldots i_{e_{(i)}}]) \times P([i_{s_{(i)}} \ldots i_{e_{(i)}}])}{P(v_i)} \times \prod_i P(s_i/v_i) \times P(M,V) \quad (2)$$

where $P(v_i/[i_{s_{(i)}} \ldots i_{e_{(i)}}])$ represents the result supplied by the recognition, $P([i_{s_{(i)}} \ldots i_{e_{(i)}}])$ is *a priori* probabilities of the images to be recognized, $P(v_i)$ are *a priori* probabilities of classes recognized by the recognition module, $P(s_i/v_i)$ defines the probability of the fragmentation of characters in the variant considered and $P(M,V)$ is the joint probability of amounts and their variant.

Since a handwritten digit string recognition system has several potential applications, it is very interesting to build a system that deals with different contexts. Due the magnitude and complexity of this kind of systems, they are usually divided into several modules, where each one assumes specific functions in order to make the system construction easier. In order to get a better comprehension of the modules in terms of a generic system, we introduce two definitions related to the system modules: Application Dependent (AD) modules and Task Dependent (TD) modules.

We define as AD those modules that should be changed (replaced or removed) to cope with another context. The best example of an AD module is the post-processor which is usually the part of the system that has more knowledge about the application. We define as TD those modules related to the task, e.g., digit string recognition. These modules can be used for various applications. In Figure 1, the white boxes represent TD modules while the grey boxes represent AD modules.

In Figure 1 we can see two different versions of the same system. The first version, which considers all modules (white and grey boxes), was designed to process numerical amounts of Brazilian bank cheques, while the second version, which does not consider the grey boxes, was designed to process strings of digits from the NIST database. As we can notice, through of the exclusion of four AD modules (*Contextual Feature*, *Structural Features*, e_3 and e_{13}) and the modification of the *Syntactic Analysis* module, we have built a new system specialized in another context. Figure 1 also presents the relationship between the system modules and its estimators. As we can notice, the proposed system takes into account the estimators of recognition ($P(v_i/[i_{s_{(i)}} \ldots i_{e_{(i)}}])$) and post-processing ($P(M,V)$) presented in Equation 2. This means that the *a priori* probabilities of the images to be recognized, *a priori* probabilities of classes recognized by the recognition module and the probability of the fragmentation of characters are not considered in this work.

3 Description of the Modules

In this section we describe all system modules depicted in Figure 1. We are using binary images (300 dpi). The system output provides a list of hypotheses associated with a probability.

3.1 Component Detection and Segmentation

The component detection module operates in three steps: connected component analysis, delimiter detection and grouping. The first step segments the string image into connected components and eliminates very small ones by filtering. We define delimiters all those components that have as main function to identify parts of the numerical amount. Usually in Brazilian documents, the point "." is used to delimit a 3-tuple of digits and the comma "," is used to identify the cents portion in the numerical amount. The grouping step tries to minimize the effects of the fragmentation by detecting potential parts and grouping each of them to its nearest neighbor.

The segmentation module that we have used in this system is based on the relationship of two complementary sets of structural features, namely, contour/profile and skeletal points. Such an algorithm takes into account an over-segmentation context and its final objective is to provide the best list of hypotheses of segmentation paths without any *a priori* knowledge of context, such as the number of characters to be segmented. More details about this method can be found in [7].

3.2 Recognition and Verification Strategy

The Recognition and Verification scheme looks straightforward, with a verification module embedded in the traditional classification system, which only has a general-purpose recognizer. The goal of a general-purpose recognizer is to attribute a given input to one of the n existent classes of the system, while the pattern verifier takes the role of an expert to precisely evaluate the result of the recognizer in order to compensate its weakness by particular training, and consequently to make the whole system more reliable. Usually, a pattern verifier is applied after a general-purpose recognizer and it is designed to "plug and play", i.e., it is used without knowing the implementation details of the recognition modules.

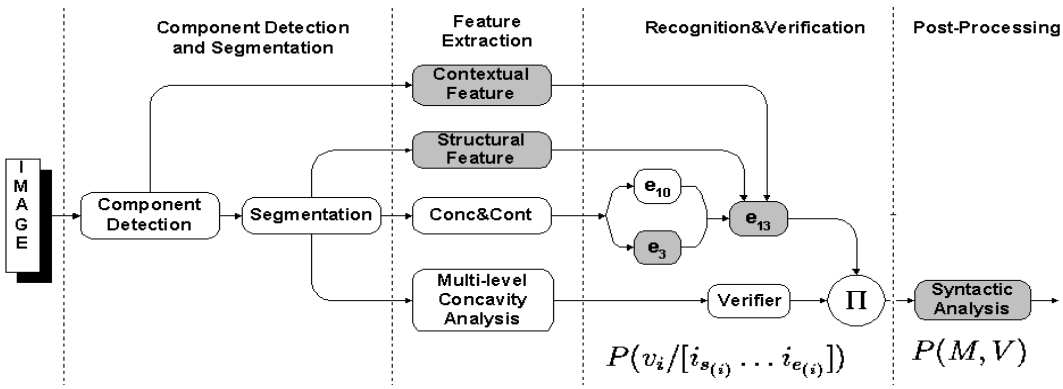

Figure 1. Block diagram of a numeric string recognition system.

Our general-purpose recognizer is composed of 3 modules (MLP Neural Networks): e_{10}, e_3 and e_{13} (Figure 1). e_{10} and e_3 are specialized in ten numerical and three non-numerical ("#", "," and ".") classes respectively. Both classifiers use a mixture of concavity and contour based features. e_{13} combines the e_{10} and e_3 outputs, one contextual feature of position and four structural features. Such a scheme of combination has been supplied the best recognition rates for numerical amounts on Brazilian bank cheques. The contextual feature of position is responsible for minimizing the confusion between the digit "1" and the symbol comma ",". As we can observe in Figure 2, the sole information capable of solving the confusion between the digit "1" and the symbol comma "," is the position of the component in the image context.

Figure 2. Confusion between the digit one "1" and the symbol comma ",".

In order to reduce the confusion between the numerical classes "4" and "7" and the non-numerical symbol block "#", we have used feature points of the skeleton. Four components have been considered: crossing points, end points and two directional points, which are detected when the skeletal path changes its direction in the horizontal or vertical axis. The database used contains 11,400, 2,000 and 4,000 images for training, validation and testing respectively. For all three sets, 80% of the database consists of digit images while the rest is composed of non-numerical images. The recognition rates achieved by the classifiers e_{10}, e_3 and e_{13} on the test set were 99.2%, 99.0% and 98.9% respectively. It is important to remark that such recognition rates are achieved when isolated characters are submitted to the recognizer. However, when these classifiers are used within a complete system, they face more complex problems such as over-segmentation and under-segmentation. In the first problem, intra-character segmentation hypothesis provides better results than any of inter-character segmentation hypothesis. Figures 3a and b show some examples of this problem. The second problem, the lack of segmentation cuts produces better results than the correct segmentation hypothesis (Figure 3c).

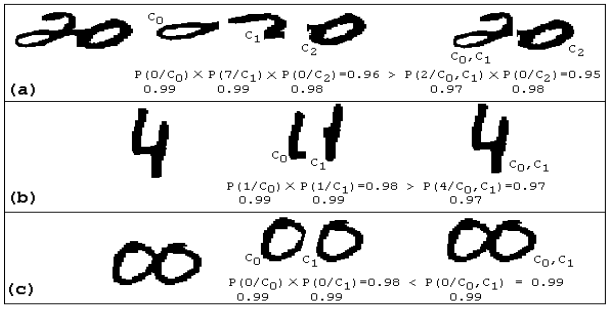

Figure 3. Misclassification caused by over-segmentation (a and b) and under-segmentation (c).

Instead of using heuristics to deal with these problems, we developed a verifier based on a neural network (MLP), which is a more robust and reliable strategy. The main objective of the proposed verifier is to improve the performance of the classifier by detecting noises generated by the segmentation algorithm, more specifically over-segmentation and under-segmentation. Thus, the verifier takes into account three different classes: over-segmentation, isolated characters and under-segmentation. The segmentation algorithm automatically generated the

over-segmentation samples. The under-segmentation samples are composed of touching digit pairs. Natural isolated characters and correctly segmented characters constitute the isolated character samples.

In order to discriminate these three classes, we developed a new feature set called Multi-level Concavity Analysis. First of all, we introduce the definitions of *Concavity Levels*. We define as *Initial Concavity Level* (ICL) for a background pixel, the number of black neighbors that it has in 4-Freeman directions. We do not consider pixels which have only one black neighbor. A fourth label is also considered in order to identify the background pixels located outside of a closed contour but with four black neighbors.

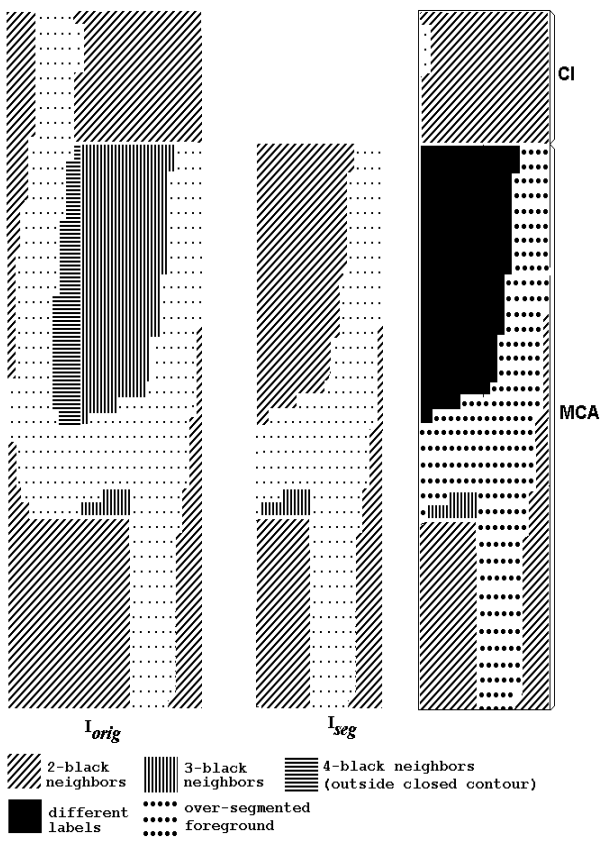

Figure 4. Multi-level Concavity Analysis (MCA) and Contextual Information (CI).

The first step of this analysis is to label with ICL the over-segmented piece (I_{seg}) and its corresponding original piece of handwriting (I_{orig}). Figure 4 exhibits an example of this procedure. Afterwards, the following verification is carried out: for each background pixel found in both images (I_{orig} and I_{seg}), we assigned to the final image (represented by the MCA in Figure 4) a specific label (represented by the black area) when the ICLs are different, otherwise, we assigned the same ICL found in both images. The same procedure is carried out over the foreground pixels. However, in such a case we assigned a specific label (represented by the bold dot) to MCA when the pixels found in both images share the same label. The latter procedure aims at computing the relative area of the over-segmented part.

The second part of this analysis concerns with I_{seg} contextual information (represented by CI in Figure 4), which is extracted taking into account the I_{seg} complement. As we can observe in Figure 4, the complement that we are using is limited to I_{seg} width. Figure 4 also shows the final labeling, which is composed of Multi-level Concavity Analysis (MCA) and Contextual Information (CI) about the over-segmented piece.

The last part of the feature vector is devoted to reduce the confusion between isolated and under-segmented characters. In order to carry out this task, we are using 6 measurements which evaluate the width and height of the image at three locations ($\frac{1}{5}$, $\frac{1}{2}$ and $\frac{4}{5}$ of the bounding box width and height). Finally, the Multi-level Concavity Analysis with Contextual Information is divided into six regions and 42 components normalized between 0 and 1 are considered. The 6 components related to the profile distances are also normalized between 0 and 1. Therefore, the final vector has 48 components. We have used 40,500, 4,000 and 4,000 images for training, validation and testing respectively. The recognition rate reached by this verifier was 99.02% on the test set.

In order to make the output of the verifier suitable to the probabilistic model presented by Equation 2, the probability of the general-purpose classifier is multiplied by the probability found by the verifier for isolated character class. Figure 5 shows how the verifier solved the confusion for the example presented in Figure 3a.

3.3 Hypothesis Generation and Post-Processor

The generation of n best hypotheses of an amount is carried out by means of a Modified Viterbi Algorithm [4], which ensures the calculation of the n best paths of the segmentation-recognition graph. Since the n best hypotheses are computed incrementally, we obtain a list of decreasing probabilities. Afterwards, each hypothesis is submitted to the post-processor module, which verifies whether it satisfies the application rules.

In opposition to the legal amount, the grammar for numerical amount is not very rich. Consequently, all syntactic rules must be based on the non-numerical symbols found in the numerical amount. Usually in Brazil, two delimiters ("#") are affixed at the beginning and at the end of the numerical amount, a point "." symbol is sometimes used to delimit a 3-tuple of digits and a comma "," symbol is used to identify the cents portion in the numerical amount. In

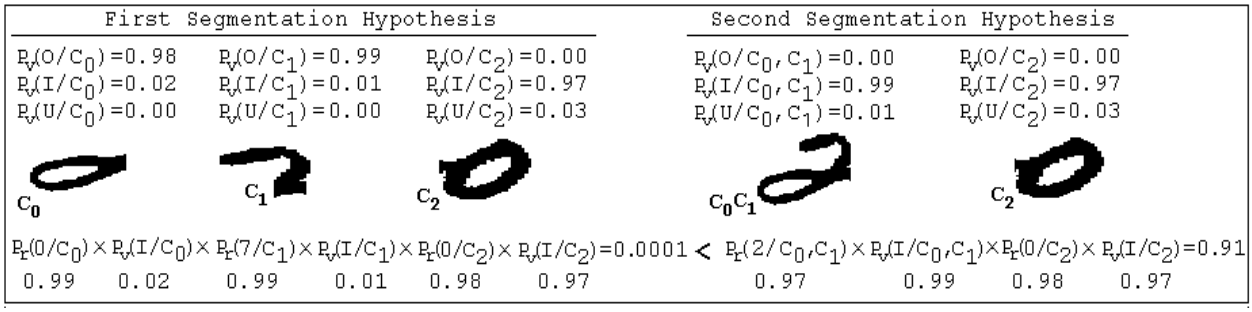

Figure 5. Use of the verifier (O : over-segmented, I : Isolated digits, U : under-segmented, P_v : probability supplied by verifier and P_r : probability supplied by the general-purpose recognizer).

addition to these rules, the Central Bank of Brazil decided that the cents portion should be present in the numerical amount [2]. In order to deal with such rules we developed a deterministic automaton which is associated with the term $P(M, V)$ of the Equation 2. Once such an automaton is formed, we turned it suitable to the probabilistic model in the following way: if the current hypothesis is verified by the automaton, then $P(M, V) = 0.99$, otherwise $P(M, V) = 0,01$.

4 Experimental Results

For our experiments a test database containing 503 numerical amounts was used. Table 1 shows the recognition rates (zero-rejection level) achieved in four different configurations of the system. This table allows us to evaluate the efficiency of the proposed verifier and the post-processor. Basically, the post-processor resolves confusions between digits and symbols. For instance, 140,00 confused with 1#0,00. In such a case, the automaton does not accept a symbol between two digits.

Considering the rank-1 for experiment 1-2-3, we divided the total error of the system into three classes: segmentation, recognition and verification. The segmentation errors are caused by under-segmentation, which is due to a lack of basic points in the neighbourhood to the connection stroke [7] (1.9%). The recognition errors are confusions of the general-purpose recognizer (38.6%), fragmentation (7.6%) and confusion generated by segmentation effects such as ligatures and noises produced by segmentation cuts (30.8%). In spite of the fact that the verifier supplied a remarkable improvement in terms of recognition rates (26.2%), it generated a new class of errors, which is basically related to the confusion between the isolated character class and the under-segmented class (21.1%). Figure 6a shows the examples of fields containing touching or broken characters that were correctly recognized by the system while Figure 6b shows the examples of misclassified fields.

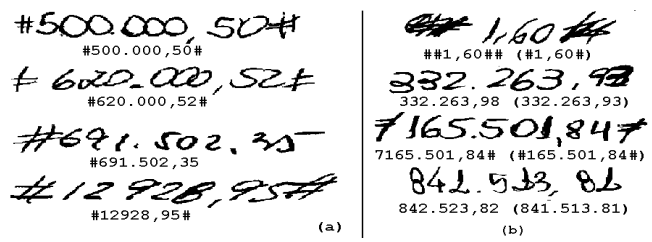

Figure 6. Examples of numerical amounts on Brazilian bank cheques. (a) Recognized (b) Not recognized.

In order to validate the concept of modular system as well as to show the robustness of our system on a well-known database, we ran an experiment consisting of NIST SD19 (hsf_7), which is an update of NIST SD3. Since this database is composed of digit strings only, all Task Dependent Modules (grey boxes in Figure 1) related to the numerical amount task were removed, except the Post-Processor module, which was changed to cope with another task. In such a case, the post-processor considers only the string length. Once we are dealing with different handwritten styles (Brazilian and North American), both general-purpose recognizer (e_{10}) and verifier were retrained. In order to train e_{10} the the training and validation sets were composed of 195,000 and 28,000 samples from hsf_{0,1,2,3} series respectively while the test set was composed of 60,089 samples from hsf_7 series. The recognition rates (zero-rejection level) achieved by the general-purpose recognizer were 99.66%, 99.63% and 99.13% on the training, validation and test sets respectively. The verifier was trained with 40,000 samples and it reached a recognition rate of 98.90% on test set. We have considered 12,800 strings of digits with lengths of 2,3,4,5,6 and 10. These

strings exhibit different problems such as, touching, overlapping and fragmentation (see Ref. [1]). Table 2 presents the recognition rates (zero-rejection level) for these experiments.

Table 1. Recognition rates (%) for the numerical amounts - 1: General-purpose recognizer, 2: Verifier, 3: Post-Processor.

Rank	System Modules			
	1	1-2	1-3	1-2-3
1	35.18	61.43	48.11	70.58
2	47.70	68.98	61.23	75.74

Table 2. Recognition rates for the NIST SD19 - 1: General-purpose recognizer, 2: Verifier, 3: Post-Processor.

String Length	System Modules			
	1	1-2	1-3	1-2-3
2	91.52	95.57	96.41	97.38
3	87.62	93.03	94.25	95.67
4	84.75	90.23	91.82	93.96
5	81.80	88.96	90.90	93.01
6	84.04	88.47	92.53	94.32
10	75.52	84.85	89.07	90.65

As we can observe in Table 2 the verifier gave a considerable improvement over the recognition rates achieved by the general-purpose classifier. However, the recognition rates achieved by the complete system (1-2-3) show that the verifier can be improved in some aspects. We have noticed that the main weakness of the verifier lies in the discrimination of isolated character classes and under-segmented class. Figure 7 presents some examples of such a confusion. The recognition rates achieved in these experiments are comparable to or better than those reported in literature [1, 8, 10].

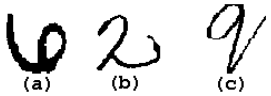

Figure 7. Images misclassified by the verifier.

5 Conclusion

We have presented in this paper a modular system under construction, which is able to cope with different applications. Combination of different levels such as segmentation, recognition and post-processing is made within a multi-hypothesis approach and a probabilistic model, which allows a sound integration of all knowledge sources used to infer plausible interpretation. We proposed a new feature set based on multi-level concavity analysis and contextual features to feed our verifier module. We have shown experimental results for numerical amounts on Brazilian bank cheques and strings of digits from NIST SD19 (hsf_7) as well. The results achieved supported the effectiveness of the proposed verification scheme.

Acknowledgements

The authors wish to thank Pontifícia Universidade Católica do Paraná and Paraná Tecnologia which have supported this work.

References

[1] A.Britto-Jr., R.Sabourin, E. Lethelier, F. Bortolozzi, and C.Y.Suen. Improvement handwritten numeral string recognition by slant normalization and contextual information. In *7th IWFHR*, pages 323–332, Amsterdam-Netherlands, 2000.

[2] Banco Central do Brasil. *Manual de Normas e Instruções - Circular n° 001825*, October 1990. http://www.bcb.gov.br/mPag.asp?codP=106&cod=112&perfil=1 (in portuguese).

[3] E.Lethelier, M.Leroux, and M.Gilloux. An automatic reading system for handwritten numeral amounts on french checks. In *3rd ICDAR*, pages 92–97, Montreal-Canada, 1995.

[4] G. Forney-Jr. The viterbi algorithm. *Proc. IEEE*, 61(3):268–278, 1973.

[5] G.E.Kopec and P.A.Chou. Document image decoding using markov source models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6):602–617, 1994.

[6] H.C.Leung. Speech recognition using stochastic explicit-segment moduling. In *2nd European Conference on Speech Communication and Technology*, pages 931–934, 1991.

[7] L.S.Oliveira, E. Lethelier, F. Bortolozzi, and R.Sabourin. A new segmentation approach for handwritten digits. In *15th ICPR*, volume 2, pages 323–326, Barcelona-Spain, 2000.

[8] S.W.Lee and S.Y.Kim. Integrated segmentation and recognition of handwritten numerals with cascade neural networks. *IEEE Trans. on Systems, Man, and Cybernetics, Part C:Applications and Reviews*, 29(2):285–290, 1999.

[9] T.M.Bruel. A system for the off-line recognition of handwritten text. In *12th ICPR*, volume 2, pages 129–133, Jerusalem-Israel, 1994.

[10] T.M.Ha, M.Zimmermann, and H.Bunke. Off-line handwritten numeral string recognition by combining segmentation-based ans segmentation-free methods. *Pattern Recognition*, 31(3):257–272, 1998.