Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta, September 15-17, 2005 (pp251-255)

Improving a Bank-Check Processing System with New HMM-based Algorithms

S. IMPEDOVO, M.G. LUCCHESE
Dipartimento di Informatica – Università degli Studi di Bari - Via Orabona, 4 – 70126 Bari - ITALY

Abstract: - In this paper a new version of a bank-check processing system is presented. Many important integrations are proposed in order to improve its classification accuracy. This goal is achieved by using the Computer Aided Software Engineering tools provided by the "Khoros" software framework. New algorithms, based on Hidden Markov Models, both for courtesy and legal amount recognition, are implemented and added to those previously used for image pre-processing, amounts recognition and classifier combination. A visual programming environment is used to re-assemble the bank-check processing system that can be easily modified and extended. Experimental results, using own and international databases, show the obtained improvements.

Key-Words: - Pattern Recognition, Bank-check Processing System, Hidden Markov Models, Software Framework.

1 Introduction

It's well-known that bank-checks are documents with colored backgrounds and different pre-printed and user-handwritten areas with a large variety of layouts, depending on the issuing bank.

Algorithms for bank-check image processing and for recognition of handwritten numerals and words have attracted great interest during these last thirty-five years. Depending on the dynamic programming [1] proposed in literature, new algorithms still today continue to receive increasing attention from scientists for the best results that can be achieved. During this long period many versions of a bank-check processing system have been produced. A first prototype, consisting of modules for bank-check image acquisition and pre-processing, CMC-7 detection and recognition, signature verification, layout processing, legal and courtesy amount recognition and amount validation, was proposed in the past [2, 3]. After that, the prototype was modified and updated in order to adapt it to the new European currency [4]. The introduction of the Euro in Europe has changed the way of writing both the courtesy and the legal amounts on checks. In consequence of this event, it was necessary to produce new algorithms and in some cases to modify the pre-existing ones in order to treat the centesimal parts on both the legal and the courtesy amounts. New solutions for the detection of a handwritten comma in the courtesy amount and of a slash followed by two handwritten digits in the legal amount were adopted. Furthermore, the operators for the recognition of legal amounts were specialized to handle a new specific grammar and a reduced lexicon consisting of only 48 basic words.

Now, in order to improve the recognition rate for digits and basic words, an approach based on Hidden Markov Models has been implemented and integrated in the previous bank-check processing system. The CASE tools provided by the Khoros framework are used for the creation and the management of a great number of new algorithms. This strategy leads to an efficient prototype of the software system. An overview of the new bank-check processing system is reported in section 2. Section 3 reports the organization of a new toolbox, the identification of the software objects and the system prototyping. Section 4 shows the experimental results.

2 Overview of the new bank-check processing system

Firstly, a bank-check is scanned at 200 dpi with 16 gray levels and transformed into an image of BMP format. After a binarization process it is reduced to only two colored levels: black (foreground of the image) and white (background of the image).

The Character Magnetic Code (CMC-7) is extracted for retrieving from a bank reference database the information about the specific bank-check layout, like the position of the user-handwritten fields for courtesy amount, legal amount and signature. After that, a pre-processing phase is used to detect and to correct slants, to smooth digit and word images, to remove underlines [5, 6] and to segment in basic elements [7]. In the system a specific module for legal amount treatment is used. Several attempts to solve the problem of handwritten legal amount recognition have been done up to now. Any legal amount is obtained by joining some basic words, according to syntactical rules, forming a small lexicon. Obviously, the type of information to be exploited is strictly related to the bank-check issuing country.

Three main approaches have been implemented in the Legal Amount Recognition Module [8, 9]: the segmentation-free, the perception-oriented and the segmentation-based methods. The segmentation-free method tries to recognize the word as a whole by extracting a description from the whole input word (i.e. number of vertical strokes, ascenders, descenders, loops etc.) and by searching for the most similar reference word in the lexicon. In the perception-oriented approach the regular and the singular parts are taken into account separately. The set of singularities allows the division of each legal amount in units of basic words. By analyzing the sequence of singularities, the basic words are grouped into 18 singularity classes. They are recognized and isolated. By detecting the separator words, the amounts included between 1 and 999 are recognized by matching their code against the codes of a suitable set of prototypes. A warping function can be used to compare the unknown hand-traced-word and the template words. In the warping, not only the distribution of the singularities, but also the size of the regular regions between templates and unknown sample and the position of singularities are taken into account. The segmentation-based approach, by using the minima along the vertical direction, identifies strokes in word letters. Then, the description of the sequence of strokes corresponding to each letter can be considered. With this description, classifiers are used to find the best match and the recognition is carried on.

Also the courtesy amount is an important field in a bank-check. It hasn't any legal function, because it is only a “courtesy” field, but it is useful in the system prototype for the correct recognition and for the validation of the legal amount.

Many pre-existing software objects continue to be still valid in the Courtesy Amount Recognition Module. Furthermore, a new recognition approach has been adopted. New algorithms based on the HMM theory are implemented and included into this module. In conjunction with the pre-existing ones, these algorithms are combined by a multi-expert system for digit recognition, consisting of a voting strategy [10]. The combination shows a significantly improvement of the performance of the prototype.

At the end, the Amount Validation Module checks the consistency of the legal and courtesy entire amount recognition and it verifies the signature correctness [11].

3 Bank-check Processing System Prototyping

The prototyping phase, carried on by using the Khoros framework, consists of the following toolboxes, that include programs and software objects related to the specific application domain. The entire software living cycle is supported by the programming tools Craftsman, Composer, Guise and Cantata [12, 13]. In the following a briefly description of a new toolbox and of new algorithms for bank-check processing is reported.

3.1 Identification of a new different application domain, creation of the toolbox and of related software objects

By using the Craftsman tool, a new toolbox, called “basic_words_recognition”, has been created in addition to the pre-existing ones, that are "image preprocessing", “layout analysis”, "segmentation", "combination methods", "legal amount recognition" and "digit recognition". In this new toolbox many software objects for the recognition of basic words in legal amounts are implemented.

Furthermore, other new programs for the classification of handwritten digits are created and added in the digit_recognition toolbox.

The algorithms grouped within each of these toolboxes were divided into subcategories according to their specific function. This organization allows a very fast and immediate access to each software component during the developing phase and during the maintenance phase.

3.2 Development of programs within the toolboxes

The tools Composer and Guise have been intensively used in this phase to create the programs that will be described in the next sub-sections.

3.2.1 Creation of the "basic_words_recognition" toolbox and its software objects

Any legal amount is a unique word obtained by combining some of the basic words, according to specific syntactic rules.

The "basic_words_recognition" toolbox, just created, contains operators that use the Hidden Markov statistical models for the recognition of 48 basic words. In recent years Hidden Markov Models are being used both in on-line and off-line handwritten text recognition [1]. An HMM topology is defined as a five-tuple <Q, O, A, B, Pi>: “Q” the finite set of N hidden states, “O” the set of M observable symbols, “A” the matrix of transition probabilities between states, “B” the matrix with the emission probabilities of observable symbols for each state and “Pi” the set of the initial state probabilities.

It is a double stochastic process with a hidden or not observable stochastic process that can be observed only through another set of stochastic processes that produce a sequence of observable symbols. The success of an HMM depends on easy to use learning algorithms enabling the re-estimation (Expectation Maximization, Baum-Welch re-estimation) of model variables, given a predefined HMM topology, and the recognition (Viterbi, Backward-Forward).

This topology influences the testing results and the total performance of the recognizer. In this experience 48 path-discriminant continuous density HMMs (each for a specific basic word, recognizing the sequence of characters forming it) are trained and tested. Each has a left to right topology, two-state transitions and each state with the same number of gaussian mixtures. A specific topology is selected heuristically since an optimal resulting model configuration is obtained in regard to error minimization [14]. The Baum-Welch algorithm was used with a fixed number of 5 iterations.

A set of observations is obtained by utilizing a strategy of an implicit segmentation [15]. Feature vectors are extracted by a sliding window that moves column by column from left to right across the handwritten word. These vectors are considered as observations. Each window is divided into 16 regular sub-regions. The ratio between the number of foreground pixels, included in each sub-region, and the total number of foreground pixels in the related window is considered as an element of the vector. In this manner the series of windows allow the extraction of the feature vector.

3.2.2 Creation of new software objects stored into the "digit_recognition" toolbox

Up to now a lot of algorithms for digit recognition have been put into the prototyping system. They vary both in features types and classification techniques [16, 17, 18]. There are algorithms with both global and local features extraction methods. They are stored into a pre-existing toolbox, called "digit_recognition" and they are “template matching”, “histogram”, “crossing line”, “region”, “contour slope”, “modified contour slope”, “characteristic loci” and “zoning”.

Since algorithms for the recognition of handwritten digits are not exhaustive and the toolboxes in the Khoros framework can be easily updated, new operators are created in order to achieve a higher recognition rate. New HMMs for off-line handwritten digit recognition are created with a lexicon of 10 elements. A sequence of observations is obtained by extracting a feature vector from every digit sample using a Freeman chain coding on the contour pixels. We have created one HMM per digit, adopting ergodic HMMs with a model-discriminant strategy. Each of them has been trained by using the Baum-Welch algorithm starting from a pre-determinate set of initial parameters for each model.

The recognition phase used the Backward-Forward algorithm. Varing the number of states in the HMM algorithms the best classification results over the test data are achieved. They were inserted into the Courtesy Amount Recognition Module in addition to the pre-existing classifiers in a multi-classifier solution.

3.3 System prototyping

The new prototype of bank-check processing system has been assembled within the Cantata visual-programming environment. It has been expanded by adding into specific modules, included in particular processing phases contained into the system, the most suitable software objects.

They are selected from those just created and dynamically connected each other forming a high-modular and easy-to-reuse data flow diagram.

Figure 1 shows an example of the visual program for digit recognition.

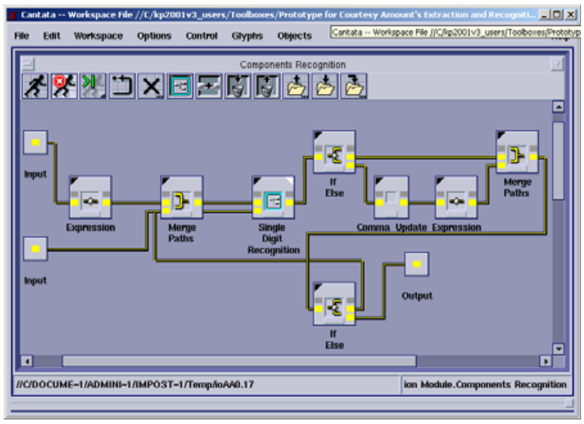

Fig.1 An example of visual program for digit recognition

4 Experimental Results

The extended Bank-check Processing System is tested on 1600 Italian bank-check samples of 85 different writers extracted from a recently developed database. The performance on Legal Amount and Courtesy Amount is evaluated in terms of recognition rate, by effecting a writer-independent classification with HMM-based algorithms. The results of the experiments are reported in the following. Each module is tested on the appropriate input data files and trained on data that differ from those available for tests.

Any Italian legal amount is written as a unique word. It consists of a sequence of basic words, created according to syntactic rules, forming a small lexicon. After the introduction of the Euro currency, only 48 simple words are sufficient to represent Italian amounts. They are reported in Figure 2.

uno, due, tre, quattro, cinque, sei, sette, otto, nove, dieci, undici, dodici, tredici, quattordici, quindici, sedici, diciassette, diciotto, diciannove, venti, ventuno, ventotto, trenta, trentuno, trentotto, quaranta, quarantuno, quarantotto, cinquanta, cinquantuno, cinquantotto, sessanta, sessantuno, sessantotto, settanta, settantuno, settantotto, ottanta, ottantuno, ottantotto, novanta, novantuno, novantotto, cento, mille, mila, unmilione, milioni.

Fig.2 Updated Italian legal amount lexicon

Figure 3 shows us an example of a legal amount. Its whole part is obtained by combining four handwritten basic words: “mille”, “due”, “cento” and “cinquanta”.

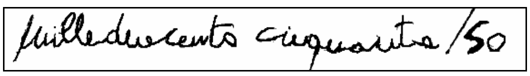

Fig.3 Example of Italian legal amount

Software objects, with continuous density HMM recognizers of basic words in LAR Module, are trained on 9600 training samples (200 samples x 48 basic words). Since the classification performance over the test data changes as a function of the number of states in the HMMs, several configurations are evaluated by training and testing the programs to find the optimal solution. Numbers of states from 16 to 28 and from 18 to 30 are used, respectively, with mixtures of gaussians consisting of 2 or 3 components. The best results in terms of correct recognition are 69.87%, by using HMMs with 28 states and mixtures of 2 components, and 78.41%, by using HMMs with 30 states and mixtures of 3 components. Segmentation-free, segmentation-based and perception-oriented methods have a performance, respectively, of about 69.05%, 64.80% and 21.00% in terms of recognition rate. The experimental results demonstrate that the recognition rate is highly increased with the use of the HMMs for basic words in the Legal Amount Recognition Module. Furthermore, discrete HMM recognizers for handwritten digits in CAR are trained and tested. In this case, the training database consists of 4500 training samples (15 different authors, 3 samples for each author and 10 classes to be recognized). Figure 4 reports recognition results of 10 HMM-based classifiers, each for a handwritten digit.

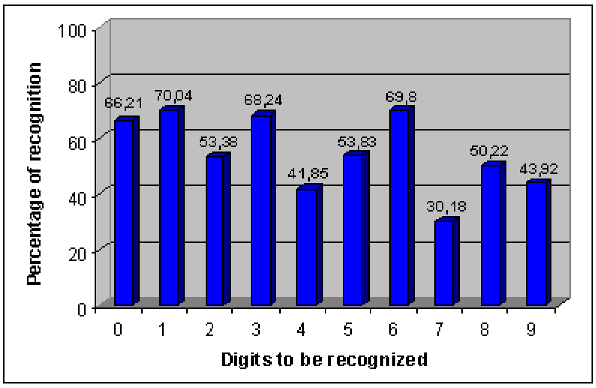

Fig.4 Recognition rates on testing digits with HMM classifiers

About 2% of average improvement of the recognition performance is expected by testing the results in progress, if all pre-existing handwritten digit recognition algorithms (trained and tested on CEDAR database – BR directory) are combined with these 10 HMM-based recognizers, put into a multi-classifier system by means of the Majority Vote combination method.

5 Conclusions

A bank-check processing system involves many fundamental tasks that are data acquisition, pre-processing (smoothing, noise filtering and segmentation), character and word recognition, signature verification and consistency validation. Since the development environment, based on the Khoros framework, is oriented to a dynamically software reuse and to a rapid structure prototyping, the system can be easily updated. In this paper an extended version of the bank-check processing system has been presented. Since generally algorithms are not exhaustive and the toolboxes can be easily updated, new software objects are implemented. Together with the pre-existing ones, a set of new algorithms for digit and basic word recognition, based on the HMM theory, has been created. The system has been re-assembled in an easy-way by utilizing the Khoros visual programming environment. An improvement on recognition results on 1600 Italian bank-checks demonstrates the feasibility of the updated system prototype. The experimental results, carried out in the field of handwritten digit and basic word recognition, point out the effectiveness of the HMM-based algorithms and make clear promising research directions.

References:

[1] L. R. Rabiner, and B. Juang, "An Introduction to Hidden Markov Models," *IEEE Signal Processing Magazine*, 1986, Vol. 3, pp. 4-16.

[2] S. Impedovo, P.S.P. Wang, and H. Bunke, *Automatic Bankcheck Processing*, World Scientific Publ., Singapore, 1997.

[3] S. Impedovo, and A. Salzo, "A New Prototype of Bank-Check Processing System", *Proc. of 4th IAPR*, Rio de Janeiro, Dec. 11-13 2000, pp. 349-360.

[4] D. Impedovo, N. Greco, M.G. Lucchese, A. Salzo, and L. Sarcinella, "Bank-check Processing System: Modification Due to the New European Currency", *Proc. of 7th ICDAR*, Edinburgh, Scotland, Aug. 3-6 2003, pp. 343-348.

[5] G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo "Removing Underlines from Handwritten Text: An Experimental Investigation", *Progress in Handwriting Recognition*, S. Impedovo and A.C. Downton Eds., World Scientific, 1997, pp. 497-501.

[6] X. Ye, M. Cheriet, C. Y. Suen, and K. Liu, "Extraction of bankcheck items by mathematical morphology", *Int. J. Document Analysis and Recognition*, Vol. 2, N. 2/3, December 1999, pp.53-66.

[7] R.G. Casey, and E. Lecolinet, "Strategies in Character Segmentation – A Survey", *Proc. of 3rd ICDAR*, Montréal, Canada, Aug. 14 - 15 1995, pp. 1028-1033.

[8] J.C. Simon, and O. Baret, "Cursive words recognition", From *Pixels to Features III-Frontiers in Handwriting Recognition*, S. Impedovo and J.C. Simon Eds, Elsevier Publ., 1992, pp. 241-260.

[9] V. Di Lecce, G. Dimauro, A. Guerriero, S. Impedovo, G. Pirlo, A. Salzo, "A New Hybrid Approach for Legal Amount Recognition", *Proc. of 7th IWFHR*, Amsterdam, Sep. 11-13 2000, pp.199-208.

[10] S. Impedovo, A. Salzo, "Evaluation of Combination Methods", *Proc. of 5th ICDAR*, Bangalore, India, Sep. 20-22 1999, pp. 394-397.

[11] G. Kim, and V. Govindaraju, "Bankcheck recognition using cross validation between legal and courtesy amounts", *Automatic Bankcheck Processing*, S. Impedovo, P.S.P. Wang and H. Bunke Eds. World Scientific Publ., 1997, pp.195-212.

[12] Khoral Research Inc., *Toolbox Programming Manual*, 1998.

[13] J. R. Rasure, and K. Konstantinides, "The Khoros software development environment for image and signal processing", *IEEE Transactions on Image Processing*, Vol. 3, 1994, pp.243-252.

[14] S. Günter, and H Bunke, "Optimizing the Number of States, Training Iterations and Gaussian in an HMM-Based Handwritten Word Recognizes", *Proc. of 7th ICDAR*, Edinburgh, Scotland, Aug. 3-6 2003, pp. 472-476.

[15] A. Vinciarelli, and J. Luettin, "Off-Line Cursive Script Recognition Based on Continuous Density HMM", *Proc. of 7th IWFHR*, Sep. 11-13 2000, pp. 493-498.

[16] D. S. Lee, and S.N. Srihari, "Handprinted Digit Recognition: A comparison of algorithms", *Proc. of 3rd IWFH*, Buffalo, NY, 1993, pp. 153-163.

[17] V. Di Lecce, G. Dimauro, A. Guerriero, S. Impedovo, G. Pirlo and A. Salzo, "Zoning Design for Handwritten Numeral Recognition", *Proc. of 7th IWFHR*, Amsterdam, Sep. 11-13 2000, pp. 583-588.

[18] G. Dimauro, S. Impedovo, R. Modugno, and G. Pirlo, "Numeral Recognition by Weighting Local Decisions", *Proc. of 7th ICDAR*, Edinburgh, Scotland, Aug. 3-6 2003, pp. 1070-1074.