



MIT Sloan School of Management

**Working Paper 4364-02
May 2002**

READING COURTESY AMOUNTS ON HANDWRITTEN PAPER CHECKS

Rafael Palacios, Amar Gupta, and Patrick. S.P. Wang

© 2002 by Rafael Palacios, Amar Gupta, and Patrick. S.P. Wang. All rights reserved. Short sections of text, not to exceed two paragraphs, may be quoted without explicit permission provided that full credit including © notice is given to the source.

This paper also can be downloaded without charge from the
Social Science Research Network Electronic Paper Collection:
http://ssrn.com/abstract_id=314765

Reading Courtesy Amounts on Handwritten Paper Checks

Rafael Palacios*, Amar Gupta**, and Patrick. S.P. Wang***

*Universidad Pontificia Comillas, Madrid, Spain

**Massachusetts Institute of Technology, Cambridge, MA, U.S.A.

***Northeastern University, Boston, MA, U.S.A.

Abstract

In spite of the rapid advances in computers and communication technologies, a number of large-scale applications continue to rely heavily on the use of paper as the dominant medium, either in an intra-organizational or an inter-organizational environment. One major example of this category of paper intensive applications is the check processing application. In a number of countries, the value of each check is read by human eyes before the check is physically transported, in stages, from the point it was presented to the location of the branch of the bank which issued the blank check to the concerned account holder. This process of manual reading of each check involves significant time and cost. In this paper, a new approach is proposed to read the numerical amount field on the check; this field is also called the courtesy amount field. In the case of check processing, the segmentation of unconstrained strings into individual digits is a challenging task because one needs to accommodate special cases involving: connected or overlapping digits, broken digits, and digits physically connected to a piece of stroke that belongs to a neighboring digit. The system described in this paper involves three stages: the segmentation of the string into a series of individual characters; the normalization of each isolated character; and the recognition of each character based on a neural network classifier.

1 Introduction

The paper check is the most popular form for non-cash payment. About 50 billion checks are processed in the United States alone every year, according to a report of the Federal Reserve Bank issued in November 2001 [19]. Paper checks still account for two-thirds of all banking transactions, despite the rapid growth in payment by credit and debit cards and other electronic means of payment. Since most of the checks need to be partially processed by hand, there is a significant interest in the banking industry for new approaches that can read paper checks automatically.

Character recognition systems are of two types: On-line systems, where the sequence in which the characters are written is known; and Off-line systems, where only the final image is available [58]. In most information technology applications, on-line processing requires a greater effort because of time constraints; however, in optical character recognition the most difficult area is that of off-line reading [3].

The account number and the bank code are printed on the checks in magnetic ink (MICR) and are the only fields that can be processed automatically with near-perfect accuracy. Since the MICR character set is a special type font, these fields can be easily read using magnetic machines or optical character recognition (OCR) systems [13]. The other fields may be handwritten, typed, or printed; they contain the name of the recipient, the date, the amount to be paid (textual format), the courtesy amount (numerical format) and the signature of the person who wrote the check.

The amount written in numbers on the check is supposed to be for courtesy purposes only and is therefore called "courtesy amount". Nevertheless, employees at the banks usually read only the amount from the courtesy amount field and ignore the amount written in words (which is the official amount). The amount of the check is entered by the employee into the computer, which then prints the amount at the bottom of the check in magnetic ink. The latter printed amount is used in all subsequent operations.

Automatic check processing has been an area of active research in image processing for a long time, but it has been only in the recent years that complete systems with reading accuracy in the range 20–60% and reading error in the range 1–3%, have begun to be installed [32]. Researchers have also described or implemented systems to read courtesy amount in checks [27, 30], and some of these systems are geared to a particular writing language; for example [29] has been developed for Korean checks, [37, 41] for checks written in French, and [1, 28] for U.S. checks. Further, some check processing systems focus on reading the legal amount [26]; see [2] for Brazilian checks, [35] for English language ones, and [37, 22, 23] for French and English ones. Also the date is checked in the system at Concordia University (Canada) [18, 54]. Finally, a Japanese system for automatic verification of bank checks is based on the extraction and recognition of the seal imprints [56]. This illustrates the broad, almost universal, interest in the area of automatic reading of bank checks.

In general, forms intended to be read by document understanding systems (DUS) are designed to include identification codes and position marks in them [9]. Most of these forms require the writer to use black pen only; this is not the case, however, with the checks processed in the US. Individuals can use pens with inks of any color. Often the checks contain guidelines or boxes to specify the location where a particular piece of information should be recorded. Also there may be restrictions on the size of the strings, both in terms of height and length. A more limiting constraint would be to preprint forms with individual boxes for each character; this restriction minimizes the incidence of connections among adjacent characters. Before information on documents can be "read", several preprocessing steps (binarization, noise reduction and line removal [52, 21, 14]) can be applied. After such preprocessing, the characters can be segmented and recognized more easily. While the recognition of isolated characters is not very difficult, segmenting string of numbers into individual digits is a formidable task. Segmentation is easy when one imposes the type of restrictions described above; however, the banks have been reluctant to adopt such restrictive measures with respect to paper checks. Overall, checks present the full challenge of totally unconstrained writing because they have not been designed for processing by document understanding systems.

A system based on emerging optical character recognition techniques and neural network techniques can help to process checks automatically in a fast and less expensive manner. The system described in this paper has been applied to read checks from Brazilian and U.S. banks. It performs all the tasks necessary to translate the image of the check coming from an optical scanner into a format that can be readily processed by the central computer of the bank. The system locates and reads the courtesy amount which is the main field that banks use to process the check.

Our approach for automated reading of courtesy amounts in checks is summarized in Figure 1. Starting with the image of the courtesy amount, the system enters into a segmentation process which is responsible of dissecting the courtesy amount field into individual characters (Section 2). This task is performed using a feedback mechanism that helps to determine if the sets of segments, produced by different dividing methods, are correctly recognized. The recognition module (Section 3) uses a combination of neural networks and structural techniques to classify digits with very high levels of confidence. The final post-processing module [47] verifies the syntax of the amount to minimize the instances involving incorrect readings.

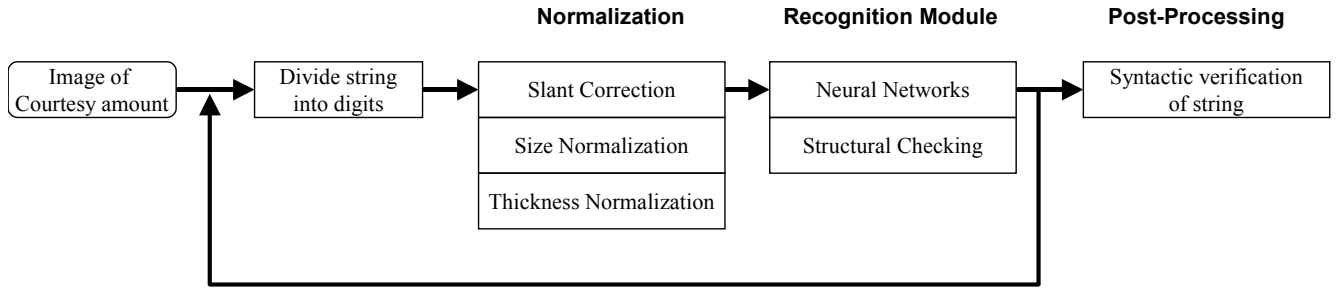


Figure 1: Key steps involved in Reading Courtesy Amount

2 Segmentation Module

Several methods and strategies for segmentation are discussed in [12]. Since the courtesy amount field may be written in different formats, sizes, and styles, the segmentation process is complicated. This may involve the separation of touching characters, and the merging of character fragments with other pieces. These are difficult tasks specially because multiple ways exist to combine fragments of characters into individual characters. This problem exists in other languages, sometimes even in printed text; for example, the problem of segmentation of Arabic characters is analyzed in [10].

Some researchers have attempted to recognize complete words [34] instead of trying to segment the word into isolated letters. This approach can only be applied in situations that involve a predetermined set of valid strings of characters. More often, the images are divided into digits using structure-based techniques [51, 15, 17] and then recognized as individual characters; this approach is called segment-then-recognize. A different approach, called segment-by-recognition [43, 44, 31], works by

applying the recognition algorithm within a sliding window that is moved along the text line. If the window covers portions of two consecutive characters, then the recognition module produces the result "unknown". However, when the sliding window takes the position and the size to match a complete character, then recognition occurs. The coordinates and the size of the segments are obtained from the position of the window. Since these systems make frequent calls to the recognition module, they need to utilize very fast and very accurate algorithms. Another drawback is that slanted text offers lower accuracy, as a consequence of using rectangular windows.

Courtesy amount is not just a sequence of digits, as people tend to use other symbols for writing monetary amounts. With respect to U.S. checks, a finite state automaton was proposed in [1] and [28] to segment and analyze the variety of styles found to express cents in fraction format. In the case of Brazilian checks, the decimal part of the amount is rarely found in a style other than scientific format. A major characteristic of Brazilian checks is the extensive use of delimiters as suffixes or prefixes of the amounts (some examples are shown in Figure 2). In countries where the currency symbol is written after the number, delimiters are used in front of the number. In countries where there is no decimal part in monetary values, delimiters are common at the end of the number.

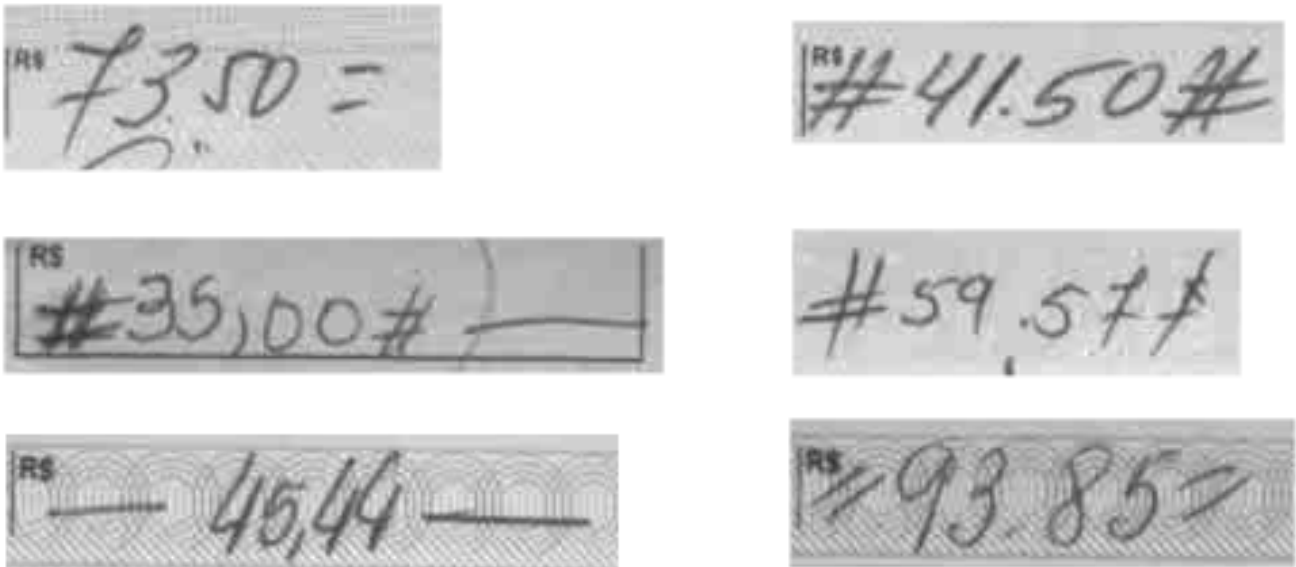


Figure 2: Some examples of delimiters in Brazilian checks

2.1 Segmentation Main Loop

The segmentation algorithm implemented by our team makes the best guess to generate a sequence of digits and symbols, and then receives feedback from the recognition module to readjust the splitting functions if necessary. The segmentation is assumed to be correct if all the individual digits are recognized with adequate confidence. The system begins the segmentation process by making a few obvious separation of characters, so the digits which are not recognized properly are considered to be comprised of connected numbers. Rejected blocks are split and recognized again in a feedback loop until

a solution is found with all the segments recognized. (The segmentation loop was shown earlier in Figure 1). The system also merges fragments of digits with neighboring digits or fragments of such neighboring digits to create new segments. If these segments are not recognized, then different merging strategies are tested. This approach is similar to the strategy proposed by Congedo [15] and Dimauro [17] that also alternates between segmentation and recognition; however their systems lack the ability to correct fragmented digits.

2.2 Splitting Algorithms

The segmentation module identifies isolated blocks of pixels called connected components [4]. Every connected component should ideally represent one digit, but in practice there are many digits touching a neighbor digit and therefore both become one single connected components. In these cases the system applies contour splitting algorithms [6, 12] to find possible paths to separate touching characters.

Several splitting algorithms are used for this task: a contour analysis, based on the one described by Blumenstain et al [7], the Hybrid Drop Fall (HDF) algorithm [33, 16], and the Extended Drop Fall algorithm (EDF) [48]. Contour analysis is a structural method that looks for minima and maxima in the vertical projections of the contours to find a dividing path. While Drop fall algorithms simulate the path produced by a drop of acid falling from above the character and sliding downward along the contour. When the drop gets stuck, it "melts" the character's line and then continues to fall. The dividing path found by a Drop Fall method depends on three aspects: a starting point, movement rules, and orientation. If the starting point is selected far from the middle line between the two characters, then the drop will fall around the exterior contour of one of the digits cutting out a little piece of just one of the digits or not cutting anything at all. Hybrid Drop Fall and Extended Drop Fall methods are described in [47]. An example of the dividing paths that can be obtained is shown in Figure 3. Drop Fall algorithms can be applied from left-to-right or right-to-left and also from top-to-bottom or in negative gravity. Hence every Drop Fall algorithm produces 4 possible dividing paths as represented by one row each in Figure 3.

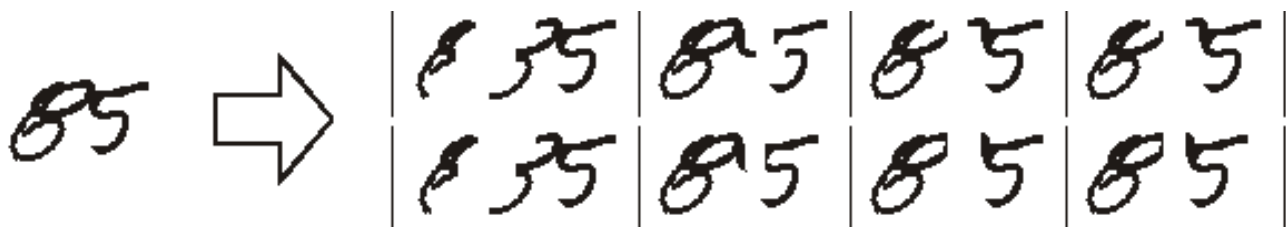


Figure 3: Different results of the drop fall algorithm depending on the starting point and direction.

2.3 Selecting Correct Dividing Path

Using three different splitting algorithms, two of them involving Drop Fall concept, produces a set of 9 possible paths to divide a connected component into individual digits: one from the contour analysis, four from HDF, and four from EDF. The second part of the split process involves deciding which path is the correct one. This determination can be made using a

neural network approach as proposed in [36]. An alternative is to undertake an exhaustive search and to select the path on the basis of the set of confidence levels provided by the recognition module [40].

For performance reasons, the current system avoids unnecessary recognition attempts as much as possible; therefore, it uses a heuristic approach to choose a path and then makes good use of the information coming from the recognition module in a feedback scheme. Paths are ranked heuristically using structural features, and taking into account the kind of algorithm used to generate the path. Several characteristics are used to rank paths:

- Number of cuts made to divide the segment: In most cases, only one cut is necessary; for example, just one long cut divides connected zeros better than two cuts.
- Length of the cut: If there is only one cut, the shorter cut usually brings a better separation.
- Type of junction: concave/convex corner.
- Referred Side: For Drop Fall methods, better rank is obtained if the cut is obtained on the same side (upper or lower) where the drop fall algorithms began, because top-left and top-right orientations tend to make erroneous cuts in the lower half of the image while lower-left and lower-right tend to make errors in the top half component.

Segmentation interacts with the recognition module to identify both digits that may comprise the particular segment. The segmentation is assumed to be correct if both digits are recognized correctly. If any character is not recognized as a digit with a high level of confidence, then alternative paths are tested. In the event that only one of the two parts is recognized, then the other part is assumed to be comprised of connected characters and the segmentation process is repeated recursively for that part until a complete solution is found. During the recognition of the courtesy amount, the system makes multiple attempts at merging and separation, and frequently it has to undo some of them to explore alternative segmentation scenarios.

When small pieces of characters are identified as "fragments of character", they are merged with the neighboring segment, taking into account relevant criteria such as proximity and overlap. The vertical overlap of segments takes precedence over proximity because the former yields better results. One example is the number '5' with a disjointed top stroke. This stroke overlaps the rest of number '5' and it could be located very close to the following digit. Nevertheless, if the digits are not recognized, a different merging option will be attempted in the next iteration of the feedback loop.

3 Recognition Module

Neural Networks constitute the heart of the recognition module used by us to classify segments into digits. In order to use a small and efficient neural network algorithm, a set of preprocessing algorithms is applied to the character images prior to the actual recognition process. The preprocessing of isolated character images involves slant correction, size normalization

and thickness normalization. As these transformations are not linear, if the algorithms are applied in a different order, they will yield a different result. Figure 4 shows two normalization examples. In the first one, slant normalization is followed by size normalization and then thickness normalization, while in the second case, the size normalization is not performed until the end. The quality looks slightly better in the second case of this example, nevertheless the first approach was selected for its speed because the thickness normalization algorithm performs much faster on small images.

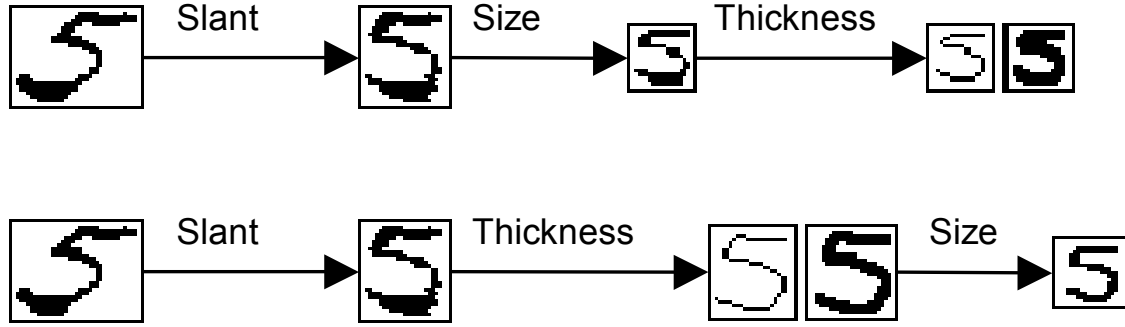


Figure 4: Two different ways of digit normalization: (slant+size+thick) and (slant+thick+size)

In order to compare both normalization procedures, 60089 digits from NIST database 19 (Handprinted Forms and Characters Database) were normalized using Matlab. The mean normalization time for the first algorithm is 0.54 s, while the mean processing time of the second normalization approach is 1.83 s. All the results are shown in Table 1 (the processing times in the C version of the prototype system are smaller). The normalization algorithms are explained in detail in the following sub-sections.

Table 1: Mean normalization times using Matlab

Digit	0	1	2	3	4
Number of samples	5893	6567	5967	6036	5873
Mean time normalization1 (s)	0.4664	0.7691	0.4830	0.5235	0.5228
Mean time normalization2 (s)	1.6728	0.8716	1.9988	1.8725	2.0546
Digit	5	6	7	8	9
Number of samples	5684	5900	6254	5889	6026
Mean time normalization1 (s)	0.5095	0.5279	0.4832	0.5820	0.5626
Mean time normalization2 (s)	1.9820	1.9177	1.6377	2.2323	2.0303

3.1 Slant Correction

The slant correction preprocessing algorithm is applied to obtain images of digits irrespective of the personal writing tilt. Most recognition methods are tolerant to minor variations from the training set; however their accuracy is degraded by the presence of slant. The algorithm used is based on the idea that if a numeral is rotated through a series of slanted positions, it usually attains its minimum width when it is least slanted. Considering α to be the angle that one wants to rotate the digit in the anti-clockwise direction (and $-\alpha$ clockwise), the equation to transform a bit-map by that angle is [8]:

$$\forall p(x,y)=1, \quad b(x',y')=1$$

where:

$$\begin{cases} x' = x - y \tan(\alpha) \\ y' = y \end{cases} \quad (1)$$

All the pixels within the same row are moved horizontally by a given amount, but they are never moved to a different row. Every row is displaced to the right or to the left (depending on the sign of α) by an amount that increases linearly with the height of the row over the baseline.

The angle that yields the least slanted position is obtained in a binary search strategy defined towards minimizing the total width of the character. The algorithm begins with an initial value for $\alpha=0^\circ$ and an initial increment $\Delta\alpha=20^\circ$. Then it analyzes $\alpha+\Delta\alpha$ and $\alpha-\Delta\alpha$ to find a better solution. When the same solution is repeated, the increment is divided by two. The process is repeated until a precision of 1 degree is reached. An example of slant correction is shown in Figure 5.

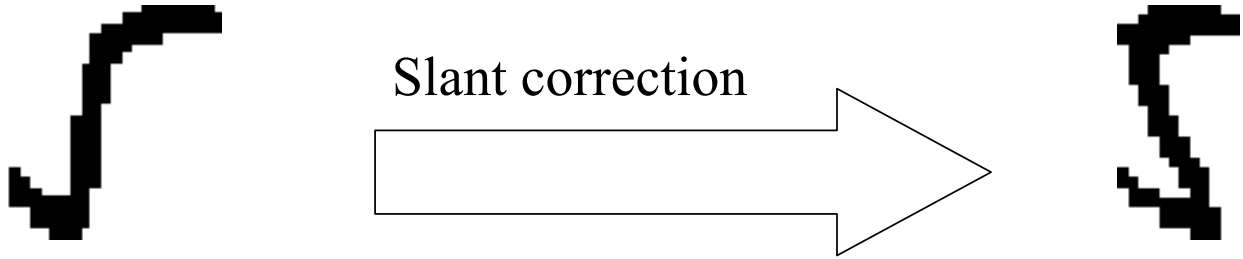


Figure 5: Slant correction applied to tilted number 5

3.2 Size Normalization

The images of the digits should be scaled to a standard matrix to make the recognition process independent of the writing size. Since every pixel in the image corresponds to one input to the neural network, the normalized size should not be very big to avoid using a complex and inefficient structure. It was found that a standard size of 16x16 pixels does not loss significant information while 256 inputs is a reasonable size to the neural network. The process of scaling to fit in a

predefined size may change the aspect ratio of the image, as the scale factors in the horizontal and vertical directions may be different. The resulting character exactly fills the area it is supposed to scale into.

Size normalization usually involves a reduction of the digit size, since check images are usually scanned at a resolution of 300 dpi that yields digits of about 80 pixels in height. For size reduction, our algorithm weights the color of the set of pixels in the original image that map over the same pixel in the 16x16 matrix. In this process, it is not critical if the resulting digits look thinner or thicker than the original, as the module described in the following section takes care of this problem.

3.3 Thickness Normalization

Thickness normalization is performed in two steps. First, a thinning algorithm is applied to reduce the strokes to a thickness of one pixel, or sometimes a few pixels. Then, a uniform rethickening process is performed to obtain thickness of several pixels.

The thinning process involves the transformation of the raw black and white image into a line drawing of unit thickness by deleting redundant pixels without damaging the appearance of the character. This process, also called skeletonization, must preserve the basic structure and connectivity of the original pattern. Algorithms proposed in the literature to solve this problem can be classified into two main categories: sequential and parallel [38]. In the serial method, the value of a pixel at the n iteration depends on pixels from the n iteration, and usually also on pixels from earlier iterations. In the parallel method, each pixel is obtained as a function of values from the previous iteration only (so the computation can be performed in parallel for every pixel of the image). The algorithm used by us is a parallel method based on the algorithm described in [57]. As evaluated in [39], this algorithm offers one of the best results in terms of performance and accuracy, as compared to nine other thinning algorithms. Since that evaluation, the algorithm was improved even further by eliminating a number of time consuming steps [45]. Additional improvements on this algorithm were attained by Carrasco and Forcada in [11] to obtain more elegant skeletons. The algorithm is presented in [47].

After scaling the input pattern down to one pixel thickness, it is necessary to re-thicken the skeleton uniformly to obtain a standard thickness. Therefore the final thickness of the digit is independent of the type of pen used, and at the same time the character appears clear of noise at the edges.

The re-thickening algorithm simply sets to 1 (black) all of the neighbor points of every pixel in the skeleton. Usually there are 6 white pixels surrounding every black pixel, but they can be 7 if the pixel is an ending, or less than 6 if the pixel is in a joint. The result of this approach exhibits a uniform thickness of about 3 pixels. One example of thickness normalization, applied to number '5' is shown in Figure 6.



Figure 6: Thickness normalization applied to number 5

3.4 Neural Network Based Recognition

Template matching, structural analysis and neural networks have traditionally been popular classification methods for character recognition, but neural networks are increasingly proving to offer better and more reliable accuracy for handwriting recognition [50, 25]. No single neural network model appears to be inherently better than others to a significant extent; instead, higher accuracy rates are achieved by tailoring network models to the particular problem environment. The structure selected for this module is the multilayer perceptron (MLP), which is the most widely used type of network for character recognition [49]. Other researchers have employed radial basis function networks (RBFN) and time delay neural networks (TDNN) for character recognition in French checks [41] and have attained similar results [50].

Two networks of the same structure trained exactly with the same data result in different parameters if the initial weights are assigned randomly. As such, it is possible to have several networks that in theory classify in the same way, but in practice produce slightly different results. By running multiple classification systems in parallel, one can increase the accuracy of the classifier [49, 53, 5, 3]. As such, the system was designed to use four networks running in parallel. Their results are analyzed by an arbiter function and a final checking is performed at the end. The structure is shown in Figure 7.

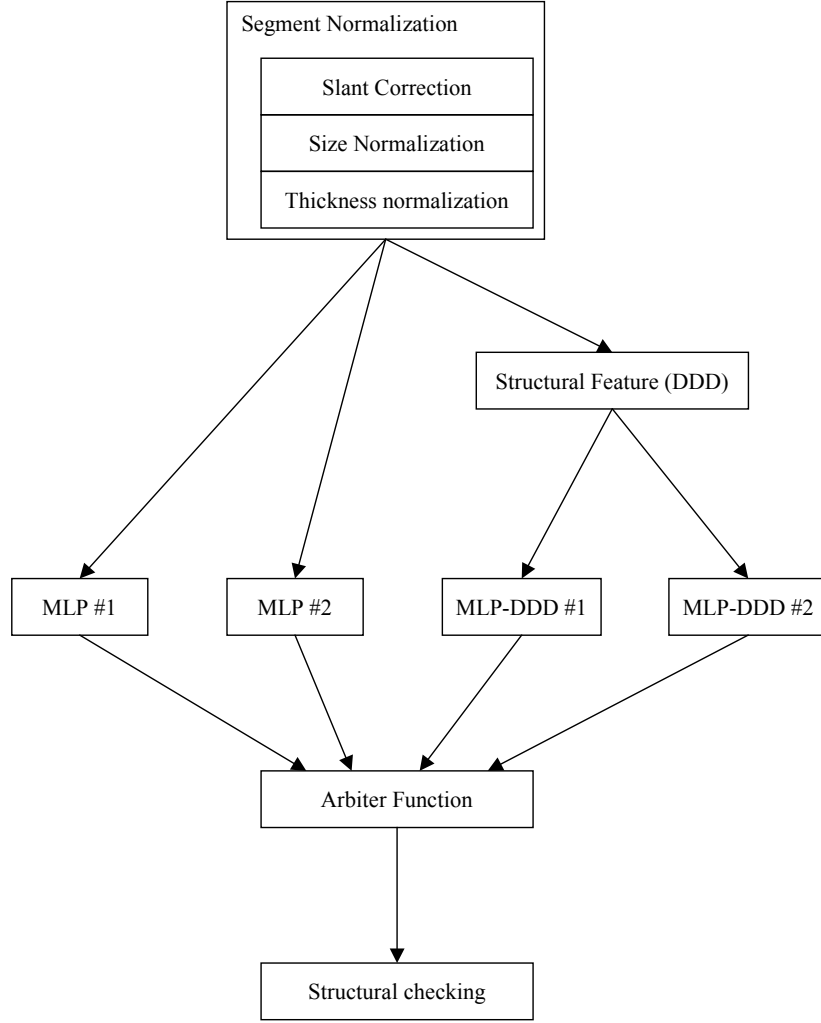


Figure 7: General scheme of Recognition Module

All the networks have a similar structure, based on three-layered, fully connected, feed-forward MLP, with 80 nodes in the hidden layer. The input is a 16x16 matrix of the normalized image of the segment; however, two of the networks also incorporate a structural feature as an input. So two networks have 256 inputs and the other two have 257 inputs. The structural feature considered is the directional distance distribution (DDD) feature discussed by Oh and Suen [46]. The outputs correspond to the digits from '0' to '9', the symbol '#' which is frequently used as a delimiter in Brazil.

Neural networks were trained by back-propagation method using 3103 segments from real checks, and were tested for accuracy of results using 1444 segments. The proportion of each digit (0 through 9), and '#' symbols was chosen to match the corresponding rates that exists in real checks. The accuracy for just one MLP networks was 80% of correct readings with 10% of incorrect readings and 10% of rejections for low confidence, while the use of MLP-DDD techniques improved the accuracy numbers to 85% of correct reading with 6% of incorrect readings.

Then another module, called arbiter, analyzes this information and produces the global output. The arbiter can decide that the segment has not been recognized properly if the global confidence has not attained the prescribed threshold or the values of the two highest confidence parameters are similar. Using these characteristics, the overall recognition rate produced by the arbiter is 87% of correct reading and 6.6% of incorrect reading.

In addition to the array of networks used to enhance the overall accuracy, a structural post-processing was added to the recognition module to verify that the recognized value is correct. The structural analysis module minimizes the number of incorrect readings by checking specific structural properties in the image for the value produced by the arbiter in cases where the global confidence is lower than 90%. It proved to be especially useful at checking cases involving '0' and also at eliminating some cases of number '5' where it was incorrectly read as '6' or as '3'. By applying the structural post-processing algorithms, several incorrect readings are eliminated. However, as a side effect, some correct reading cases are also rejected because they do not pass the strict structural test. Since the goal is to have minimum number of errors even though the reject rate may increase, we accepted this consequence as being consistent with our design objectives. The final overall accuracy of the recognition module is 84% of correct reading and 5.5% of incorrect reading. The statistical results are shown on Table 2.

Table 2: Accuracy of the recognition module using a database of segments from real checks that include: normal digits, delimiter symbols, and segments of touching and overlapping digits.

	Correct	Incorrect	Rejected
MLP	79.9%	10.2%	9.9%
MLP-DDD	85.2%	6.0%	8.7%
Arbiter	86.8%	6.5%	6.6%
After structural Post-Processing	84.0%	5.5%	10.5%

Most researchers have opted to train their recognition systems with huge databases of digits in order to attain very high accuracy levels [55]. These systems used digits from the NIST database of handwritten forms [20] or other databases of pre-segmented digits. But it was found that the segmentation feedback loop requires a `NOT_RECOGNIZED` value in the case of incorrectly segmented digit (Figure 8) or if the segment contains connected digits. In our opinion, the recognition module must be trained with segments containing fragments of digits and segments containing connected digits to be able to produce `NOT_RECOGNIZED` values properly in these cases.

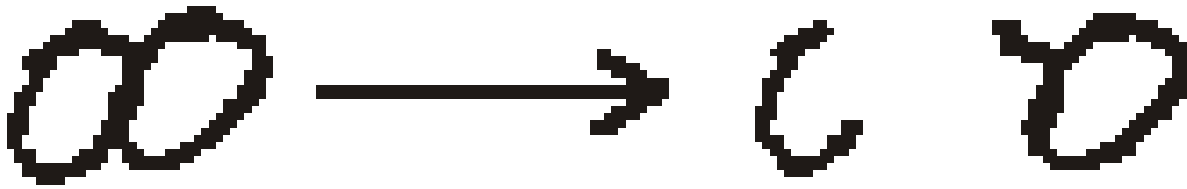


Figure 8: Connected digits may be incorrectly separated by the segmentation loop, in this case each segment should be classified as `NOT_RECOGNIZED`

A neural network designed to recognize just digits should produce a very low confidence in case of fragmented characters or pairs of joint digits, so that those segments can be rejected. After extensive testing, it was found that a neural network designed to recognize normal digits as well as special segments is more accurate at rejecting bad segmented cases. When a bad segmented character is rejected, the segmentation procedure makes use of other splitting algorithms until it finds the correct path to separate the digits properly. Therefore, by training the network with poor quality inputs, the overall performance reading strings of number that include touching and overlapping digits is substantially enhanced.

4 Prototype Program

A prototype has been developed using the above approach to show how a commercial application might be able to read checks of poor quality from many different countries. The program has been programmed in C++ under Microsoft Windows operating system. The graphical user interface, shown in Figure 9, allows to load the image of a check and then to perform automatic recognition.

The program transforms the image from gray-scale into black and white, then locates the courtesy amount area, and finally begins the iterative process of segmentation and recognition described in this paper. After recognition, the output is verified to ensure high validity and accuracy levels [47].

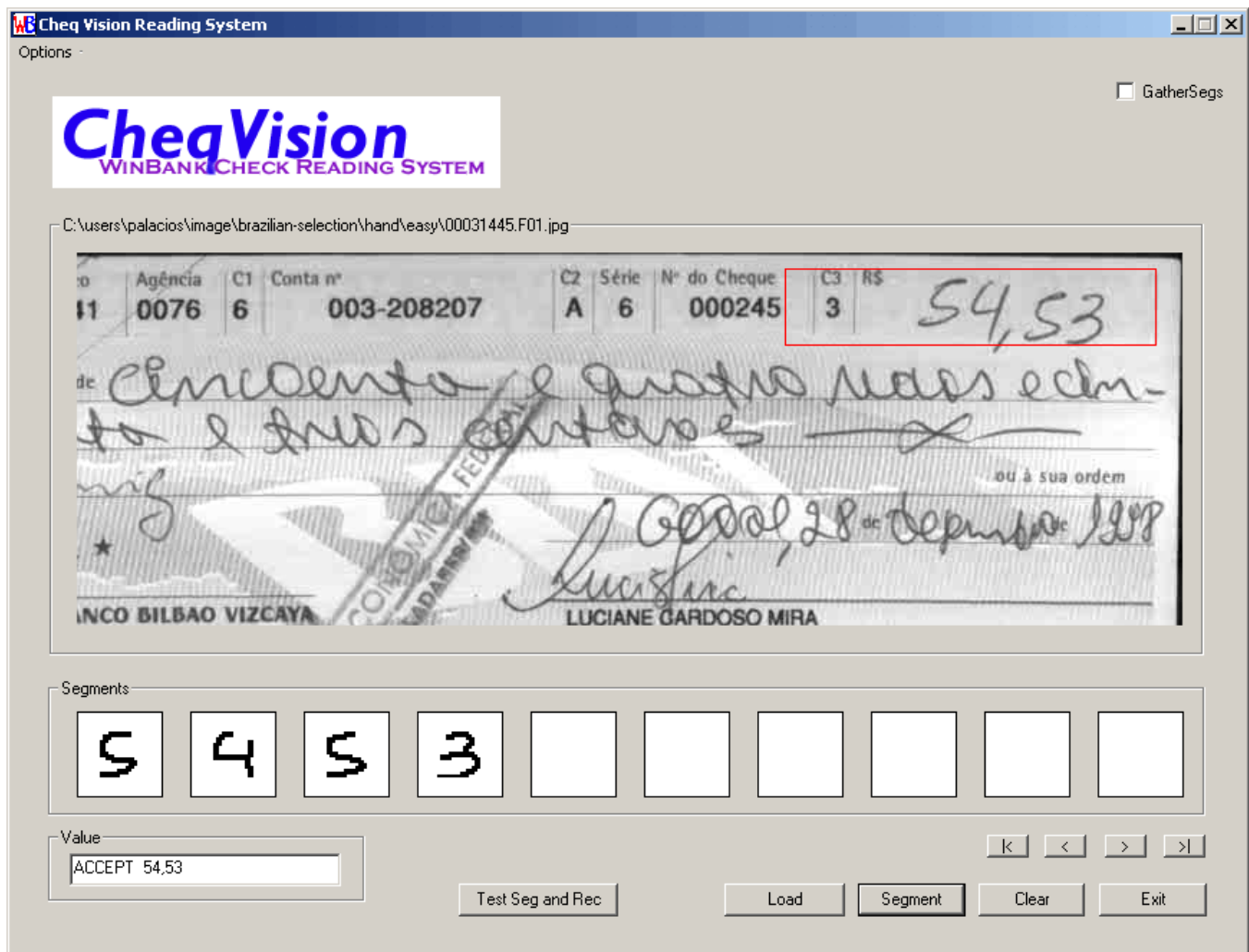


Figure 9: Recognition of a Brazilian check using the prototype application

5 Conclusion

This paper has presented an approach for recognition of handwritten amounts that can be utilized for "reading" bank checks automatically, being the first step in paper-less processing. The algorithms and techniques described in this paper can be applied to read checks from many countries or to different environments that can benefit from automatic reading or image processing.

The segmentation of the amount into individual digits is performed using several splitting algorithms. Segmentation module may receive some feedback from the recognition module and will select a different splitting algorithm if any digit is not properly recognized. Several splitting algorithms based on the drop-fall methodology have been presented in this paper. Neither algorithm provides the best separation of connected or overlapped digits in every possible situation, but the system tries the most likely one first; if it fails then it uses the other techniques.

In the section on recognition of individual characters, the need of normalization of digit size, thickness of keystrokes, and slant has been highlighted. The neural network architecture employs a set of four neural networks of different types that are run in parallel to minimize the likelihood of erroneous readings. This architecture offers superior accuracy and performance, as compared to structures comprised of single nets only. The importance of a classifier that is able to detect incorrectly segmented digits has become increasingly visible and we are undergoing additional research in this topic.

6 Acknowledgments

The authors thank fellow members of the project team for their valuable contributions. We also thank representatives of a number of banks located in North America, South America, Europe and Asia for assisting with research related to their respective domains. Parts of the research described in this paper are covered by a patent [24]; we also thank the co-inventors, and others who were involved in the process.

7 References

- [1] A. Agarwal, A. Gupta, K. Hussein, "Bank Check Analysis and Recognition by Computers" in Handbook of Character Recognition and Document Image Analysis. Editors: H.Bunke, and P.S.P. Wang. World Scientific, 1997.
- [2] C.O. de Almendra Freitas, A. el Yacoubi, F.Bortolozzi, R.Sabourin. "Brazilian Bank Check Handwritten Legal Amount Recognition", Proceedings XIII Brazilian Symposium on Computer Graphics and Image Processing: 97-104, 2000.
- [3] N. Arica and F.T. Yarman-Vural, "An overview of Character Recognition Focused on Off-Line Handwriting", IEEE Transactions on Systems, Man, and Cybernetics – PartC: Applications and Reviews, 31(2):216-233, 2001.
- [4] A. Agarwal, K. Hussein, A. Gupta, and P.S.P. Wang. "Detection of Courtesy Amount Block on Bank Checks" Journal of Electronic Imaging, Vol. 5(2), April 1996, pp. 214-224.
- [5] A.S. Atukorale, P.N. Suganthan, "Combining Classifiers Based on Confidence Values", Proceedings of the 5th International Conference on Document Analysis and Recognition, pp.37-40, 1999.
- [6] H. Baird, H. Bunke, P.S.P. Wang. Document Analysis and Recognition, World Scientific, Singapore, 1994.
- [7] M. Blumenstein and S. Verma, "A Neural Based Segmentation and Recognition Technique for Handwritten Words". IEEE International Conference on Neural Networks Vol. 3, 1738-1742. 1998.
- [8] R. Bozinovic, and S. Srihari, "Off-line Cursive Script Word Recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 2, pp 68-83, 1989.
- [9] H.Bunke and P.S.P.Wang. Handbook of Character Recognition and Document Image Analysis. Ed. World Scientific. ISBN 981-02-2270-X, 1997.
- [10] B.M.F. Bushofa and M. Spann, "Segmentation and Recognition of Arabic Characters by Structural Classification", in Image and Vision Computing, 15:167-179, 1997.
- [11] R.C. Carrasco, and M.L. Forcada, "A Note on the Nagendraprasad-Wang-Gupta Thinning Algorithm", Pattern Recognition Letters, Vol 16, pp. 539-541, 1995.
- [12] R.G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", in IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(7):690-706, 1996.
- [13] F. Chin, F. Wu, "A Microprocessor-Based Optical Character Recognition Check Reader", Proceedings of the third International Conference on Document Analysis and Recognition, Vol 2:982-985, 1995
- [14] E. Cohen, J.J. Hull, and S.N. Srihari. "Understanding Handwritten Text in a Structured Environment: Determining Zip Codes from Addresses", in Character and Handwriting Recognition: Expanding Frontiers, Editor: P.S.P. Wang, pp 221-264, World Scientific Publishing Co. 1991.

- [15] G. Congedo, G. Dimauro, S. Impedovo, G. Pirlo "Segmentation of Numeric Strings." Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. II 1038-1041. 1995.
- [16] S. Dey "Adding Feedback to Improve Segmentation and Recognition of Handwritten Numerals", Master Thesis, Massachusetts Institute of Technology. 1999.
- [17] G. Dimauro, S. Impedovo, G.Pirlo, and A.Salzo. "Automatic Bankcheck Processing: A New Engineered System", International Journal of Pattern Recognition and Artificial Intelligence, Vol 11(4), pp467-503, 1997.
- [18] R. Fan, L. Lam, and C.Y. Suen, "Processing of Date Information on Cheques", Progress in Handwriting Recognition, Editors A.C. Downton and C. Impedovo, World Scientific, 1997, pp.473-479.
- [19] Federal Reserve System (U.S. Government), "Payments System Study Results. Nov 2001", <http://www.federalreserve.gov/boarddocs/press/general/2001/20011114/default.htm>, Mar 10, 2002
- [20] M.D. Garriss, J.L. Blue, G.T. Candela, P.J. Grother, S.A. Janet, C.L. Wilson. "NIST Form-Based Handwritten Recognition System: Release 2.0", US Dept. of Commerce, Technology Administration. National Institute of Standards and Technology. 1997.
- [21] R.C.Gonzales and P.Wintz. Digital Image Processing, Addison-Wesley, Reading, MA (1987)
- [22] N.Gorski, V.Anisimov, E.Augustin, O.Baret, D.Price, and J.C.Simon, "A2iA Check Reader: A Family of Bank Recognition Systems", Fifth International Conference on Document Analysis and Recognition, pp 523-526, Bangalore, India. (1999)
- [23] D. Guillevic and C.Y. Suen, "HMM-KNN Word Recognition Engine for Bank Check Processing", Proceeding of the International Conference on Pattern Recognition, Vol 2. 1998.
- [24] A. Gupta, M.V. Nagendraprasad, P.S.P. Wang, "System and method for character recognition with normalization", U.S. Patent No. 5633954, 1997.
- [25] I. Guyon and P.S.P. Wang. Advances in Pattern Recognition Systems using Neural Network Technologies. Ed World Scientific. ISBN 981-02-144-8. 1993
- [26] K. Han and I. Sethi, "An Off-line Cursive Handwritten Word Recognition System and its Application to Legal Amount Interpretation", Automatic Bankcheck Processing, Wold Scientific Press, 1997, pp. 295-308.
- [27] L. Heutte, P. Barbosa-Pereira, O. Bougeois, J.V. Moreau, B. Plessis, P. Courtellemont, and Y. Lecourtier, "Multi-Bank Check Recognition System: Consideration on the Numeral Amount Recognition Module", International Journal of Pattern Recognition and Artificial Intelligence, Vol 11(4), pp595-618, 1997.
- [28] K. Hussein, A. Agarwal, A. Gupta, and P.S.P. Wang. "A Knowledge-Based Segmentation Algorithm for Enhanced Recognition of Handwritten Courtesy Amounts", Pattern Recognition, Vol 32(2), pp. 305-316, Feb 1999.
- [29] T.C. Jee, E.J. Kim, Y. Lee, "Error Correction of Korean Courtesy Amounts in Bank Slips using Rule Information and Cross-Referencing", in Proceedings of the International Conference on Document Analysis and Recognition, pp 95-98. 1999.
- [30] G. Kaufmann and H. Bunke, "A System for the Automated Reading of Check Amounts – Some Key Ideas", Proceedings of the 3rd International Association of Pattern Recognition Workshop on Document Analysis Systems. Nagano, Japan, 1998, pp. 302-315
- [31] J. Keeler, D. Rumelhart, "A Self Organizing Integrated Segmentation and Recognition Neural Network". Proceedings of SPIE: The International Society for Optical Engineering Vol. 1710 744-755. 1992.
- [32] S. Kelland and S. Wesolkowski, "A comparison of Research and Production Architectures for Check Reading Systems", Fifth International Conference on Document Analysis and Recognition, pp 99-102, Bangalore, India.. 1999.
- [33] S. Khan, "Character Segmentation Heuristics of Check Amount Verification", Master Thesis, Massachusetts Institute of Technology. 1998
- [34] J.H. Kim, K.K. Kim, C.P. Nadal, and C.Y. Suen, "A Methodology of Combining Hmm and Mlp Classifiers for Cursive Word Recognition", Proceedings of the International Conference on Pattern Recognition, Barcelona, Spain, Vol 2, pp. 319-322, 2000.
- [35] K.K. Kim, J.H. Kim, Y.K.Chung, and C.Y. Suen, "Legal Amount Recognition Based on the Segmentation Hypotheses for Bank Check Processing", Proceedings of the Sixth International Conference on Document Analysis and Recognition: 964-967, 2001

- [36] F. Kimura and M. Shridhar, "Segmentation-Recognition Algorithm for Zip Code Field Recognition", *Machine Vision and Applications* 5 (3) 199-210. 1992.
- [37] S. Knerr, V. Anisimov, O. Beret, N. Gorski, D. Price and J.C. Simon, "The A2iA Intercheque System: Courtesy Amount and Legal Amount Recognition for French Checks", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol 11(4), pp505-547, 1997.
- [38] L. Lam, S.W. Lee, and C.Y. Suen, "Thinning Methodologies – A Comprehensive Survey", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):869-885, 1992.
- [39] L. Lam and C.Y. Suen, "An Evaluation of Parallel Thinning Algorithms for Character Recognition", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):914-919, 1995.
- [40] S.W. Lee, D.J. Lee, H.S. Park, "A New Methodology for Gray-Scale Character Segmentation and Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (10) 1045-1050. 1996.
- [41] E. Lethelier, M. Leroux, M. Gilloux, "An Automatic Reading System for Handwritten Numeral Amounts on French checks." 3rd International Conference on Document Analysis and Recognition, vol 1:92-97, 1995.
- [42] J.S. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ: Prentice Hall, pp. 469-476. 1990.
- [43] G. Martin, "Centered-Object Integrated Segmentation and Recognition of Overlapping Handprinted Characters". *Neural Computation* 5, 419-429. 1993.
- [44] G. Martin, R. Mosfeq, J. Pittman, "Integrated Segmentation and Recognition Through Exhaustive Scans or Learned Saccadic Jumps". *International Journal of Pattern Recognition and Artificial Intelligence* 7 (4) 831-847. 1993.
- [45] M. V. Nagendraprasad, P. S. P. Wang, and A. Gupta, "Algorithms for Thinning and Rethickening Digital Patterns" *Digital Signal Processing*, 3, 97-102, 1993.
- [46] I. Oh and C.Y. Suen, "A Feature for Character Recognition Based on Directional Distance Distributuion", *Proceedings of 4th International Conference on Document Analysis and Recognition. ICDAR*. 1997.
- [47] R.Palacios and A.Gupta, "A system for processing handwritten bank checks automatically", submitted to *Image and Vision Computing*, Feb 2002.
- [48] J. Punnoose "An Improved Segmentation Module for Identification of Handwritten Numerals", Master Thesis, Massachusetts Institute of Technology. 1999.
- [49] E. Roman, A. Gupta, J. Riordan."Integration of Traditional Imaging, Expert Systems, and Neural Network Techniques for Enhanced Recognition of Handwritten Information." IFSRC . No. 124-90. 1990
- [50] A. Sinha, "An Improved Recognition Module for the Identification of Handwritten Digits", Master Thesis, Massachusetts Institute of Technology. 1999.
- [51] P.L. Sparks and M.V. Nagendraprasad, and A. Gupta, "An Algorithm for Segmenting Handwritten Numerals". *Second International Conference on Automation, Robotics, and Computer Vision*, Singapore, 1992.
- [52] S.Srihari, C.Wang, P.Palumbo, and J.Hull. "Recognizing Address Blocks on Mail Pieces", *AI Magazine* 8(4):25-40. Dec. 1987.
- [53] C.Y. Suen, C. Nadal, R. Legault, T. Mai, and L. Lam, "Computer Recognition of Unconstrained Handwritten Numerals", *Proceedings of the IEEE*, 80(7):1162-1180, 1992.
- [54] C.Y. Suen, Q. Xu, and L.Lam, "Automatic Recognition of Handwritten Data on Cheques – Fact or Fiction?", *Pattern Recognition Letters*. 20:1287-1295, 1999
- [55] C.Y. Suen, J. Kim, K. Kim, Q. Xu, and L. Lam, "Handwriting Recognition – The Last Frontiers", *Proceedings of the International Conference on Pattern Recognition*, vol. 4:1-10, 2000.
- [56] K. Ueda, T. Mutoh and K. Matsuo, "Automatic Verification System for Seal Imprints on Japanese Bankchecks", in *Proceedings of the International Conference on Pattern Recognition*, vol 1. 1998.
- [57] P.S.P.Wang, and Y.Y.Zhang, "A Fast and Flexible Thinning Algorithm", *IEEE Transactions on Computers*, 38(5), 1989
- [58] P.S.P.Wang. *Character and Handwriting Recognition: Expanding Frontiers*. Ed. World Scientific. ISBN 981-02-0710-7, 1991.