Automatic Processing of Brazilian Bank Checks

Rafael Palacios*, Anshu Sinha** and Amar Gupta**

*Universidad Pontificia Comillas, Madrid, Spain

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

Abstract

This paper focuses on determining the value of the check, which is the first step needed for automatic processing. The multi–staged approach proposed in this paper deals with four aspects: the detections of the numerical amount field on the check; the segmentation of the string into a set of characters; the recognition of isolated characters; and the syntactic verification of each output. All modules have been tunned to recognize Brazilian checks. Through the use of feedback mechanism between recognition and segmentation, the adoption of a new normalization approach, the use of several neural networks in parallel, and the incorporation of syntactic verification methods, our research team has been able to significantly enhance the overall accuracy levels.

Keywords: Document imaging, check processing, optical character recognition, unconstrained handwritten numerals, neural networks.

1 Introduction

According to the Federal Reserve Bank, checks accounted for 60% of the non-cash transactions in 2001 in the United States [19]. More checks are written for bill payment or remittance that for any other purpose (25.7% of check volume); and the use of credit cards and debit cards is increasing primarily at the point of sale. In Brazil and other countries, the volume of checks is also important and there is a significant interest in the banking industry for new approaches that can process paper checks automatically. Checks are currently processed at significant cost for: determining the value of the check as specified by the account holder; physically transporting the check from the place where the account holder entered that amount to the location of the bank in which the account is maintained; and initiating appropriate transactions at the intervening banks and financial institutions.

This paper focuses on the first step in automatic processing that consists of reading the information contained in the check and translating it into electronic format. 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 (OCR) systems [12]. The other fields may be handwritten, typed, or printed, but most commonly they are handwritten by individuals since many companies are making payments using electronic transfers instead of printing checks. These fields 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 official value of the check is the amount written in words, while the amount written in numbers 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 other field, altogether. The amount of the check is entered by the employee into the computer, and it then it may be read by another employee to reduce the chances of incorrect readings.

Automatic check processing has been an area of research in image processing for a long time, but it has only been in recent years that complete systems with reading accuracy in the range of 20–60% and reading error in the range of 1–3%, have begun to be installed [33]. Checks present the full challenge of totally unconstrained writing, in contrast with other applications of character recognition and document analysis [9]. The position of the fields is not fixed and there are no lines or boxes to specify the location where every digit should be written within one field.

The system described in this paper locates and reads the courtesy amount, which is the main field that banks use to process the check. Other researchers have also described or implemented systems to read courtesy amount in checks [28, 32], and some of these systems are geared to a particular writing language; for example [31] has been developed for Korean checks, [38, 43] for checks written in French, and [1, 30] for U.S. checks. Further, some check processing systems focus on reading the legal amount [27]; see [2] for Brazilian checks, [36] for English language, and [38, 23, 24] for French and English. Also the date is checked in the system at Concordia University (Canada) [18, 60]. Finally, a Japanese system for automatic verification of bank checks is based on the extraction and recognition of the seal imprints [62]. This illustrates the broad, almost universal, interest in the area of automatic reading of bank checks.

To read the courtesy amount, we have developed the approach summarized in Figure 1. The first step in the process is to detect the courtesy amount within the image of the check which involves, a conversion from the gray scale image into a binary data format. The most challenging part of the process is the segmentation process, which involves dissecting the courtesy amount field into individual characters. The latter 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 is based on neural networks to classify digits with very high levels of confidence. This classifier was selected based on the success achieved in handwriting recognition and other pattern recognition applications [26]. Before neural networks can be applied, the segments are normalized to minimize the impact of different writing styles and to adjust the size of the image to the number of nodes in the input layer. The final post-processing module verifies the syntax of the amount to minimize the instances involving incorrect readings.

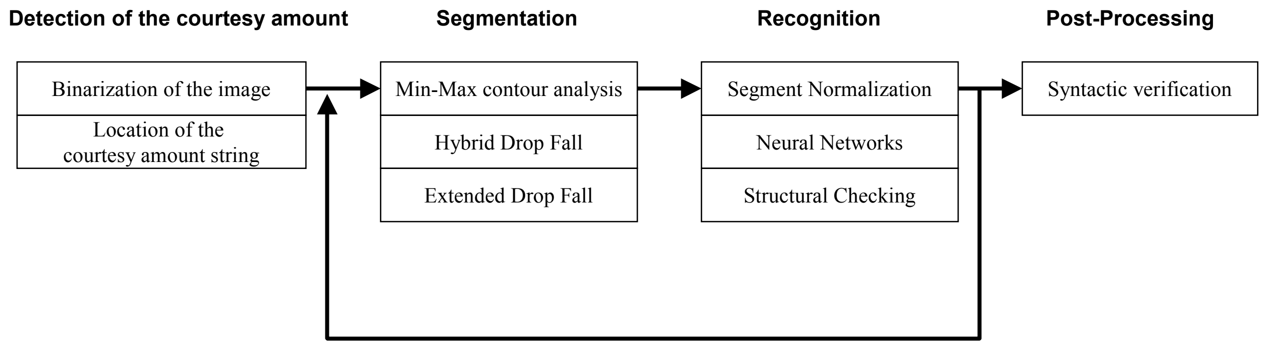

Figure 1: Description of key steps

2 Detection of courtesy amount

Bank checks may come in different sizes, have different backgrounds, and the courtesy amount area may be defined in different ways. Since paper checks are scanned in gray scale mode, the first step in the recognition process is a conversion into black and white that properly removes the background. The best results are obtained using a dynamic threshold based on the histogram of the level of gray in the image [49]. Then the system has to identify which area of the check contains the pixels that conform the courtesy amount string. One of the approaches developed by our team for identifying the courtesy amount string in any kind of bank check involves three stages: organizing the information in blocks of connected components; identifying potential string candidates; and formulating the decision of which string represents the amount of the check [5]. But in the case of checks from Brazil, the courtesy amount is always located at the upper right corner and usually delimited by vertical and/or horizontal lines. So in this case the location of the courtesy amount is accomplished by searching for horizontal and vertical lines in the upper right part of the image. Based on the relative position of these lines, the area of interest is then defined as a rectangle embedded within the lines. Since the size and the aspect ratio of this rectangle are fairly constant, the amount field can be accurately determined, even if the area is not delimited by multiple lines. This process is shown in Figure 2. Other researchers have proposed a less general approach to extract information from Brazilian checks by reading the MICR line to identify the bank, and using a database of backgrounds and content layouts [39].

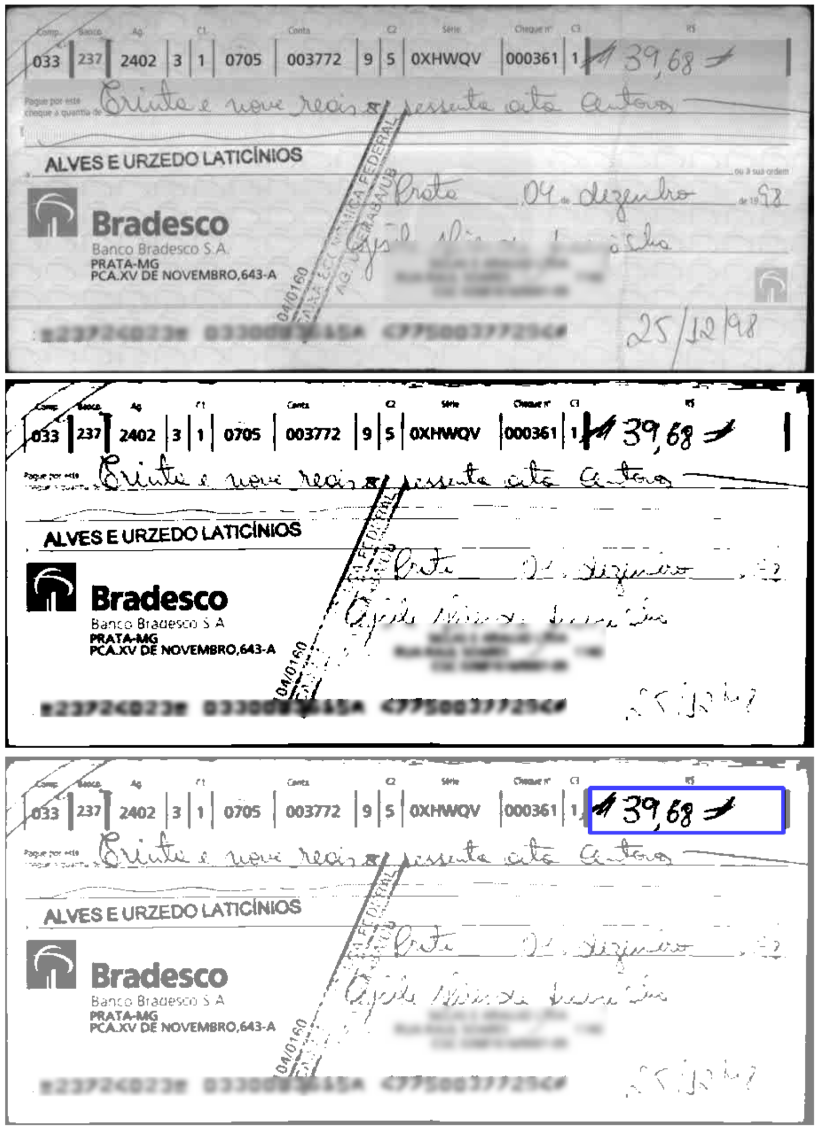

Figure 2: Detection of Courtesy Amount

3 String Segmentation

Segmentation of the amount string is the process of extracting individual digits to be recognized in a classifier. Character segmentation techniques are well described in [11]. Since the courtesy amount field may be written in different formats, sizes, and writing styles, segmentation process is complicated. The segmentor is expected to divide the courtesy amount into isolated digits without any prior knowledge of the digits in the image or even what a digit is. It has no information about the width or the height of the character and these data vary greatly depending on the digit. One approach employed by some researchers has been to attempt to recognize complete words instead of trying to segment the word into isolated letters [33]. Clearly, this solution can only be applied in very constrained situations that involve a predetermined set of valid strings of characters. More often, the images are divided into digits using structure-based techniques [51, 11] and then recognized as individual characters; this approach is called segment-then-recognize. This process may involve the separation of touching characters, and the merging of character fragments with other pieces. These are difficult tasks especially because multiple ways exist to combine fragments of characters into individual characters. In other applications segmentation is easier since the number of digits is fixed, such as zip codes [13, 37, 41, 44, 45] and recognition of characters in plates of private Brazilian vehicles [54].

3.1 Feedback Strategy for Segmentation

Our approach for segmentation is based on a recursive function that uses splitting algorithms to divide blocks into isolated digits. The system begins the segmentation process by making a few obvious separation of characters, the primitives obtained are pre-classified as digit, fragment, multiple, or delimiters. Then the fragments are merged with digits or other fragments and analyzed again. The segmentation is assumed to be correct if all the individual digits are recognized with adequate confidence, and the digits which are not recognized properly are considered to be connected numbers. These blocks are divided and recognized again in a feedback loop until a solution is found with all the segments recognized. This approach is similar to other strategies proposed in [14, 17, 41] and combines the characteristics of structural dissection methods and pure recognition-based segmentation [11]. A more complicated multi-agent architecture has been proposed to read Brazilian checks [3]. The accuracy of the multi-agent approach is currently low, however it was shown that interaction between recognition and segmentation was very helpful.

3.2 Dividing Blocks

The goal of our segmentation process is to represent each digit as a 2 dimensional bitmap array. This is one of the most difficult steps in unconstrained text recognition. In general, forms intended to be read by document understanding systems (DUS) [9] are designed to include identification codes and position marks in them, so the segmentation can be done based on relative coordinates. This is not the case of paper check, where the segmentor has no prior knowledge of the digits in the image and has no information about the width or the height of the character. Given this, it should also be clear that it can be very hard for the segmentor to determine whether an extracted segment represents 2 digits or 1, or even only a part of a digit. The algorithms that are used for segmentation, then, must make use of heuristics that address the nature of joints between characters [57, 58].

Using the binarized courtesy amount block image as input, the segmentation module identifies the isolated blocks and applies contour splitting algorithms to find possible paths to separate touching characters [7, 11]. Several dissection algorithms can be used for this task, those implemented in the system are Hybrid Drop Fall (HDF) [34, 16] algorithm, and Extended Drop Fall algorithm (EDF) [52], which are based on "hit and deflect strategy" proposed in [?]. Both 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 behavior of HDF and EDF algorithms is not the same since they use different movement rules to split the characters. Both algorithms are described in detail in [49]. One example of the results obtained by the drop fall algorithms is shown in Figure 3. In this example, the second direction (starting point at the upper right side) attains the best separation.

Figure 3: Different results of the algorithms HDF and EDF depending on the starting point and direction.

4 Recognition Module

The recognition module is based on neural networks techniques to classify images of isolated digits as the numbers they represent. Prior to the actual recognition process, a set of preprocessing algorithms is applied to the character images standardized in order to minimize structural variations so that the recognizer does not have the impossible task of accounting for many different writing styles. Then a set of neural networks is used to classify the segment into one digit or a special character, with high accuracy and speed. Another module for structural checking can be used to increase the accuracy even more.

4.1 Segment Normalization

The preprocessing of isolated character images involves slant correction, size normalization and thickness normalization. In size normalization, the segments are scaled down to a pre-determined size to match the fixed number of input nodes of the network. Slant correction and thickness normalization are used to minimize the impact of structural variations in the segment. These operations are nonlinear, thus applying the pre-processing algorithms in different order will yield different results. The processing time can also vary depending on the order of the algorithms. Since thickness normalization may require up to 90% of the total processing time, reducing the size before applying the thinning algorithm reduces the processing time significantly. After extensive testing it was found that the best results, according to image quality and processing time, are obtained applying the algorithms as described in Figure 4.

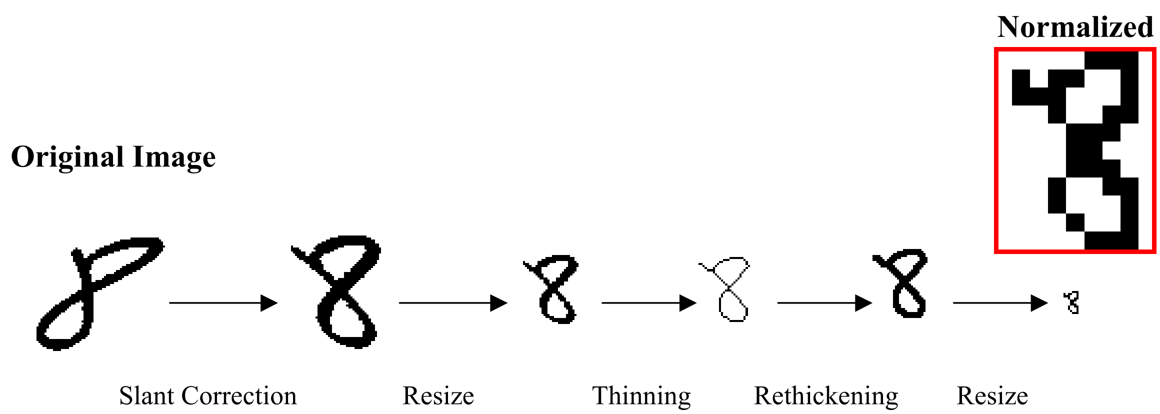

Figure 4: Normalization process

Slant correction is applied at the highest resolution to obtain the same kind of image for the same set of numbers irrespective of the personal writing tilt. The algorithm used is very efficient so it can be run at the highest resolution in a very short time [8, 49].

The first resize is applied to adjust the image to a standard 39x27 size, irrespective of the personal writing size. This resize is necessary to perform the thickness normalization accurately and faster than working with a bigger matrix. Resize operation is performed using the same vertical and horizontal scaling factors, therefore preserving the original aspect ratio. As a consequence, some columns will be filled with white pixels in slim characters or some rows will be filled with white pixels in wide characters or sets of connected digits. Better accuracy has been obtained by preserving the aspect ratio of the digits that in previous work [56, 49]. In addition, this has shown good results detecting cases of bad segments such as segments made of multiple digits [50].

Thickness normalization is performed in two steps. First, a thinning algorithm is applied to reduce the strokes to a thickness of one pixel. Then, a uniform rethickening process is performed to obtain thickness of several pixels. The thinning algorithm used is a parallel method based on the algorithm described in [63]. As evaluated in [40], 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 [46]. Additional improvements on this algorithm were attainted by Carrasco and Forcada in [10] to obtain more elegant skeletons.

The re-thickening algorithm is a dilation morphological algorithm [22] involving a 2x2 kernel matrix. Therefore it sets to 1 (black) four of the neighbor points of every pixel in the skeleton. The result of this approach exhibits a uniform thickness of about 2 pixels.

The final step in the normalization process is a resize to a 13x9 matrix, this is the matrix that contains 117 pixels which are the inputs of the neural network. Without the first resize, the thinning process is very slow and re-thickening should be performed according to the original size, because bigger matrices require thicker strokes in order to obtain similar results after the final resize. It was also found that 117 pixels is enough information to attain accurate recognition and this size is less sensitive to overtraining than using higher resolution samples (unless the number of digits in the training set is huge). The new normalization approach improved the accuracy of the neural network compared to the previous procedure that normalized the digits into a bigger matrix (16x16) without preserving the aspect ratio [49]. The approach described above increased the recognition rates from 82.6% to 92.4%, while reducing the level of incorrect readings from 8.1% to 2.1%.

4.2 Neural Network Based Recognition

Template matching, structural analysis and neural networks have been the most popular classification methods in character recognition. In general, neural network techniques have been most successfully used for handwritten data recognition. They perform better than traditional image processing techniques alone. Template matching is a technique most useful for machine-printed data and has been successfully employed in this area. However, handwritten data vary so widely that template matching becomes too complex and unreliable. While Fourier transforms have been used for this task with some success, the associated computation time involved is too large. Structural recognition modules face a similar problem since the structure of handwriting cannot be algorithmically quantified and computation time is limited.

The advantage of using a neural network for handwritten character recognition is that it can construct nonlinear decision boundaries between the different classes in a non-parametric fashion, and thereby offer a practical method for solving highly complex pattern classification problems. Furthermore, the distributed representation of the input's features in the network provides an increased fault tolerance in recognition; thus character classification can occur successfully when part of the input is broken off and not present in the image, as well as when extra input signals are present as a result of noise [55]. This is a very important characteristic for a recognition module in this application, since the input that the recognizer receives is the result of the segmentation module, which sometimes does not produce perfect digits. This unconstrained environment for recognition can cause the precise cases noted above; namely, noisy images and inputs with broken structure or additional pieces added on (see Figure 5).

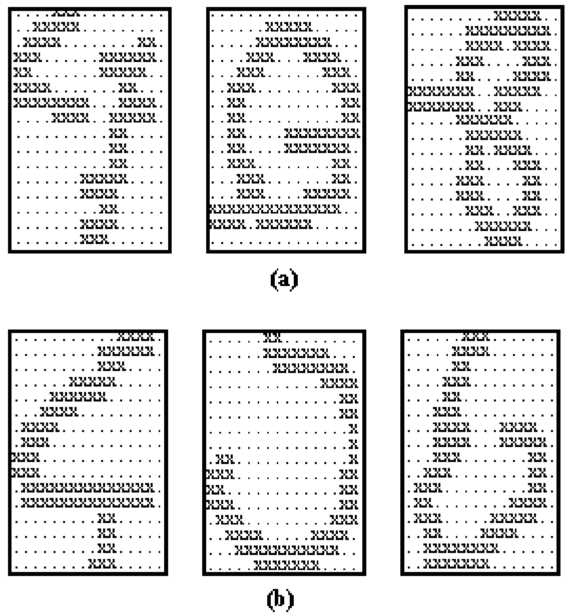

Figure 5: Input to the neural network; (a) noisy segments that have pieces attached; (b) segments with pieces broken off.

By using a neural network approach, the recognizer is able to absorb such variations in the input. However, no 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 a particular problem environment. So, a network trained to recognize just digits offers a better accuracy for check amount recognition than a generalized neural network trained to recognize both letters and numbers.

The model implemented in this case is a three-layer, fully connected, feed-forward multi-layer perceptron (MLP). The MLP structure is the simplest to understand and to implement, and it is the most widely used for character recognition [15, 55].

The MLP structure implemented has three layers: an input layer of 117 nodes, a hidden layer of 50 nodes and an output layer with 11 nodes. The nodes of the input layer correspond to the pixels of the input bitmap obtained after normalization. The hidden layer of 50 nodes is used to extract increasingly meaningful patterns from the input data. The number of nodes in the hidden layer was obtained by general trial and error, taking into account accuracy and computation time. Another important feature to note about the network is the use of only one hidden layer. Some studies have indicated that the number of hidden layers is generally not a factor in the performance of the network [15]. In fact, it has been rigorously proved that the performance achieved with any number of hidden layers can be matched with one layer of hidden nodes. Therefore, the more important factor is the number of nodes and the amount of computation that they can accomplish.

The outputs correspond to the digits from '0' to '9', the symbol '#' which is used to classify bad segments. A function that compares the levels obtained at each output node, selects the digit with the highest confidence or produces the value `NOT_RECOGNIZED` if the winner is not clear or has a low confidence. The segmentation feedback loop requires a `NOT_RECOGNIZED` value in the case of incorrectly segmented digit or if the segment contains connected digits. This information allows to try different split algorithms in segments that are not classify as digits.

The recognition module was trained with pre-segmented digits from NIST database of handwritten forms [21] and segments produced by the segmentation module using images of real checks. In contrast, some researchers have trained their recognition systems just with pre-segmented digits from standard databases in order to attain very high accuracy levels [61]. Such systems bypass the problems related to segmentation, and cannot be compared to the recognition module used for reading bank checks in an equitable manner. In our experience, the handwriting styles in Brazil and the U.S. vary significantly for some digits, especially number 1 and 7. Figure 6 shows some examples of the writing styles found in Brazilian checks compared to the styles included in the standard database of the NIST that was generated scanning forms written by individuals in the U.S.

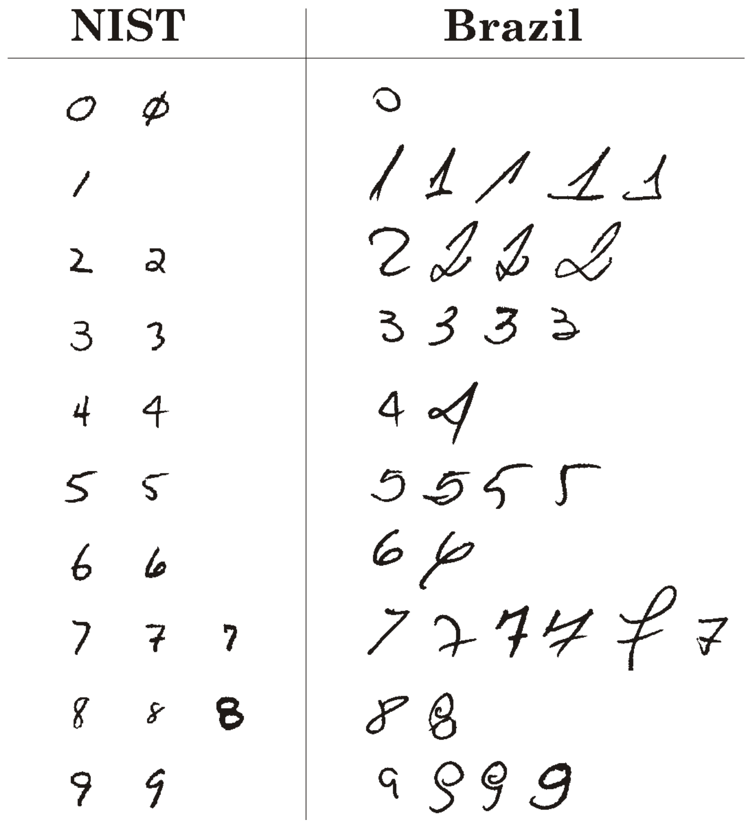

Figure 6: Different writing styles found in NIST database and Brazilian checks

In Brazil, and many other countries, individuals tend to use special symbols at the beginning and at the end of the amount, to avoid the possibility of adding numbers to change the value of the check; we call "delimiters" to these symbols (see Figure 7). Tests showed that the accuracy of the recognition module is better if trained with segments containing fragments of digits and segments containing connected digits or special symbols, because the neural network is more likely to detect these cases [50]. The number of incorrect classification of multiple segments as digits can be reduced from 51.6% to 6.2% by providing samples of touching digit pairs during the training process. The use of delimiters in the courtesy amount occurred on 36% of the checks in Brazil, and only half of them could be pre-classified as delimiters by their size and aspect ratios. Many of these symbols are transmitted to the neural network which tries to recognize them as digits; this highlights the importance of classifying them as delimiters prior to the recognition stage.

However, if the neural network is not correctly trained and it classifies special symbols as digits, then the automated reading process fails for 17% of the checks from Brazil.

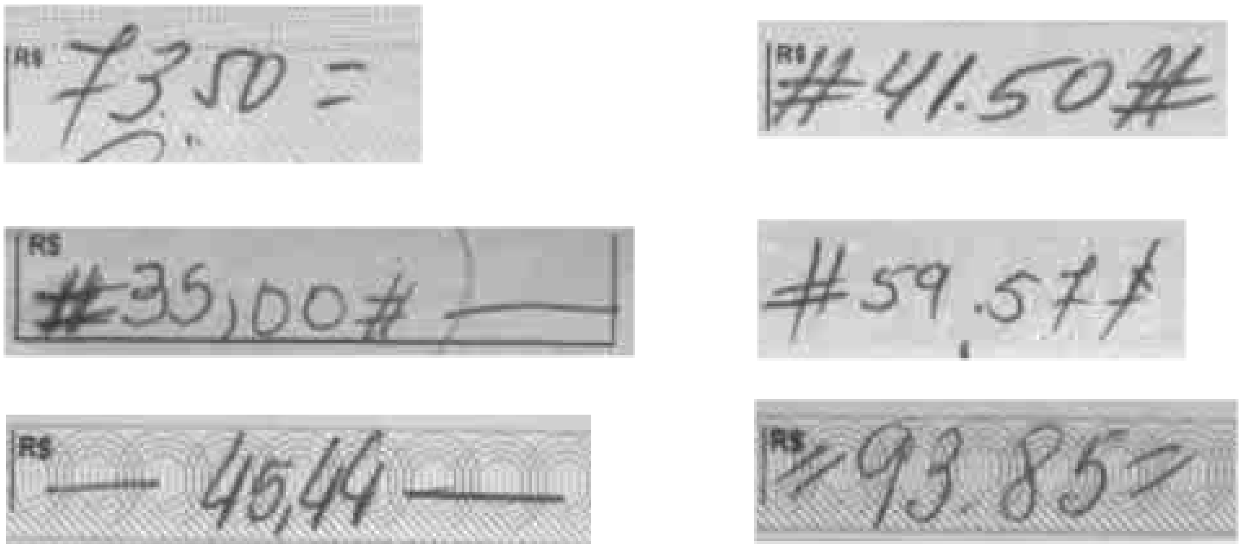

Figure 7: Some examples of delimiters in Brazilian checks

The standard back-propagation algorithm was used to train the networks. These algorithms use input patterns whose classifications are known, measuring the error at each output node based on the known value. The error is then propagated backwards through the network and used to adjust the free parameters of the network, or the weights of the nodes. The weights of the network represent plausible microinferences which, when applied to the input, reveal patterns that satisfy the largest number of these constraints and contradict the least [55].

Two networks of the same structure trained exactly with the same data result in different parameters if the initial weights are assigned randomly. By running multiple classification systems in parallel, one can increase the accuracy of the classifier [4, 6, 35, 53, 55, 59]. As such, the system was designed to use several neural networks in parallel and one analysis function, called arbiter, to select the output comparing the results of every network. This approach is depicted if Figure 8.

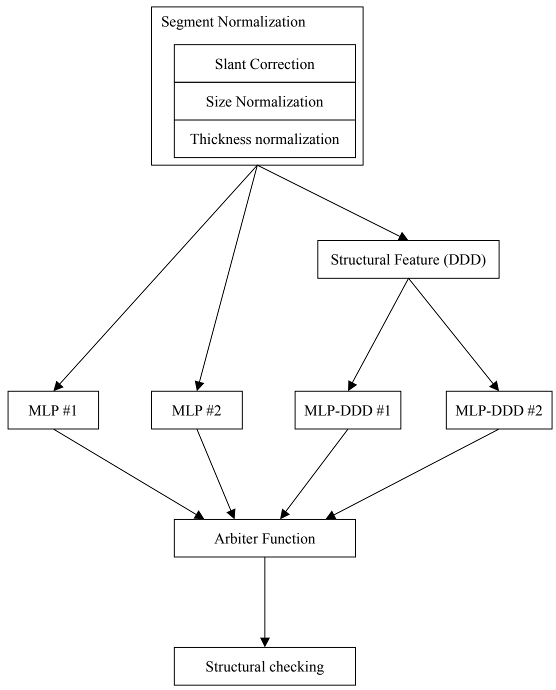

Figure 8: General scheme of Recognition Module

Several combinations of parallel neural networks have been tested. Using four MLP, two of them using a structural feature along with the normalized image of the digit, improved the recognition rate on check characters from 82.6% to 86.8%, and reduced incorrect reading from 8.1% to 6.5% [56, 49]. The structural feature considered was the directional distance distribution (DDD) feature discussed by Oh and Suen [48]. The most recent system used 3 MLP in parallel, that were trained with NIST digits (normalized according to the procedure described above); it improved correct recognition from 92.4% to 93.2%, and reduced incorrect readings from 2.1% to 1.6% [50].

4.3 Structural Checking

A structural checker was added to the recognition module to verify that the recognized digit is correct. The structural analysis minimizes the number of incorrect readings by checking specific structural properties in the image for the value produced by the arbiter in those cases where the global confidence is lower than 90%. It proved to be especially useful at checking cases involving '0' and also eliminating some cases of number '5' where it was incorrectly read as '6' or as '3'.

The algorithm uses various heuristics to verify values. For example, it employs loop detection to look for loops where there should be loops, i.e. '6', '8', '9', '0'. It can also identify open loops such as those found in '3' or '5'. Since the network architecture is designed to absorb errors in the input image such as broken strokes because of segment separation or other reasons; this can cause a number such as '5' to be recognized as a '6' or vice versa. Their structures are similar, however the '5' has an open loop on the bottom where the '6' has a closed loop. Using loop detection on such digit it is possible to identify whether it has a closed loop or not. The figure below (Figure 9) shows examples of cases where the structural checker can be used to ensure results of the neural network. The small '*'s in the images denote the locations where the stroke could have been broken off. (a) shows a half of a '0' that had been incorrectly classify as a '6' by the network, this segment should be rejected. (b) shows how the '9' can resemble an '8'. The *s in the images denote the locations where the stroke could have been broken off.

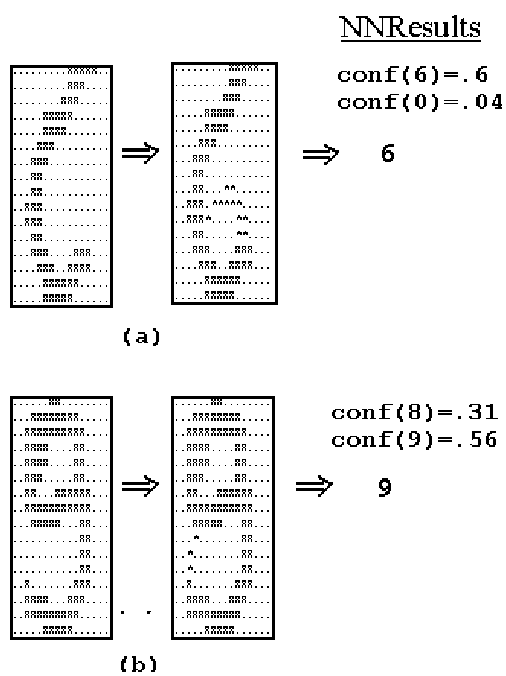

Figure 9: Cases where the neural network requires the aid of the structural verifier.

By applying the structural post-processing algorithms, several incorrect readings are eliminated. Nevertheless, as a side effect, some correct reading cases are also rejected because they do not pass the strict structural test. In other applications, rejection of a value translates to rejection of an entire amount which can severely reduce recognition rates. However, our approach is based on a feedback loop, so rejected characters at this point will mandate the segmentation module to use supplementary split algorithms in order to produce different separation paths. Therefore implementing structural checking inside the segmentation loop improves the recognition rate while minimizing the false positive rate.

5 Post Processing

The final step in reading the courtesy amount is aimed at reducing the number of incorrect results even further. In this post processing stage, the resulting string is analyzed to determine whether the recognized value is a valid amount for a bank check. Special attention is paid to the punctuation in order to eliminate incorrect results. This module is based on a set of rules that describe the valid amount formats for a given language [49]. The rules shown in this paper correspond to the valid Brazilian styles.

Contextual knowledge can often be exploited to enhance the recognition accuracy in most OCR applications. The courtesy amount can also be verified with the legal amount, written as a text sentence [23, 27, 32]. Some systems that read monetary amounts exploit contextual information via syntax verification. This involves the notion of pattern parser [51] that determines the validity of a string in terms of monetary amounts. The A2iA Interchange system [38] includes the verifier module within the recognition module to read French checks. While in another French system [28], the syntactic analysis takes place after the recognition system. In the system being described in this paper, the syntactic verifier is applied after recognition and is intended to minimize the number of wrong readings and to show the values in a standard format [29]. The post processing module is based on a set of rules that incorporate the meanings of the decimal separator and the grouping symbol. There is no distinction between a period and a comma, both are treated as separators, and the analysis is based on the number of digits in each group. It is done in this way because the period sign and the comma sign are difficult to distinguish in handwritten form, and because they have opposite connotations in different countries. As an example, the norms for the new Euro currency [47] allow for the use of the decimal symbol and the digit grouping symbols according to the national rules and practices of every country.

The input to the post-processing module is a text string made of digits, punctuation and '#' characters.

The most basic syntax of the string in a Brazilian check can be expressed as the following regular expression [20] in the following way:

```
[#]*[0-9,]*[#]*
```

This expression means that the amount string may begin or end with none, one, or more special delimiter characters. Between the delimiters there could be any combination of digits from 0 to 9 or a comma separator may exist. After eliminating the delimiters, the syntax of the number can be analyzed in more detail paying attention to the sequence of digits and punctuation. The only accepted styles (for values lower than 1,000,000.00) are those that conform with the following regular expressions:

Set 1: If the decimal part exists, which is the most common case, it must contain 2 digits. The regular expressions for amounts with decimal part are:

```
^[0-9]*,[0-9][0-9]$

^[0-9],[0-9][0-9][0-9],[0-9][0-9]$

^[0-9][0-9],[0-9][0-9][0-9],[0-9][0-9]$

^[0-9][0-9][0-9],[0-9][0-9][0-9],[0-9][0-9]$
```

Set 2: Without decimal part the separator is optional. So the regular expressions for amounts without decimal part are:

```
^[0-9]*[,]*$

^[0-9],[0-9][0-9][0-9][,]*$

^[0-9][0-9],[0-9][0-9][0-9][,]*$

^[0-9][0-9][0-9],[0-9][0-9][0-9][,]*$
```

In any case it is possible to group characters within the integer part, but these groups must be comprised of exactly three digits. If no grouping is done, all the digits can be written sequentially.

In the check recognition system proposed by Hussein et al. [30], a parser based on a Deterministic Finite Automata (DFA) is proposed within a module called "segmentation critic module", this module was used to ensure proper segmentation of U.S. checks before recognition. The algorithm used to verify the correct syntax in Brazilian checks is essentially a Deterministic Finite Automaton (DFA) shown in Figure 10. It reads the string from right to left and the states change depending on the kind of symbol found. The alphabet Σ is the set of symbols that the string is comprised of (equation 1). In this case the alphabet has only two elements: D for digit or P for punctuation. V is a finite set of valid states in the system, and $F \subseteq V$ is the set of final states, which are represented as double circles in the graph. The transitions from one state to another depend on the characters found in the string and are clearly marked in the graph.

$$
\begin{aligned}
\Sigma &= \{ 'D' , 'P' \} \\
V &= \{ q_0 , q_1 , q_2 , q_3 , q_4 , q_5 , q_6 , q_7 , q_8 , q_9 , q_{10} , q_{11} , q_{12} \} \\
F &= \{ q_3 , q_6 , q_7 , q_8 , q_9 , q_{10} , q_{11} , q_{12} \}
\end{aligned} \quad \textbf{(1)}
$$

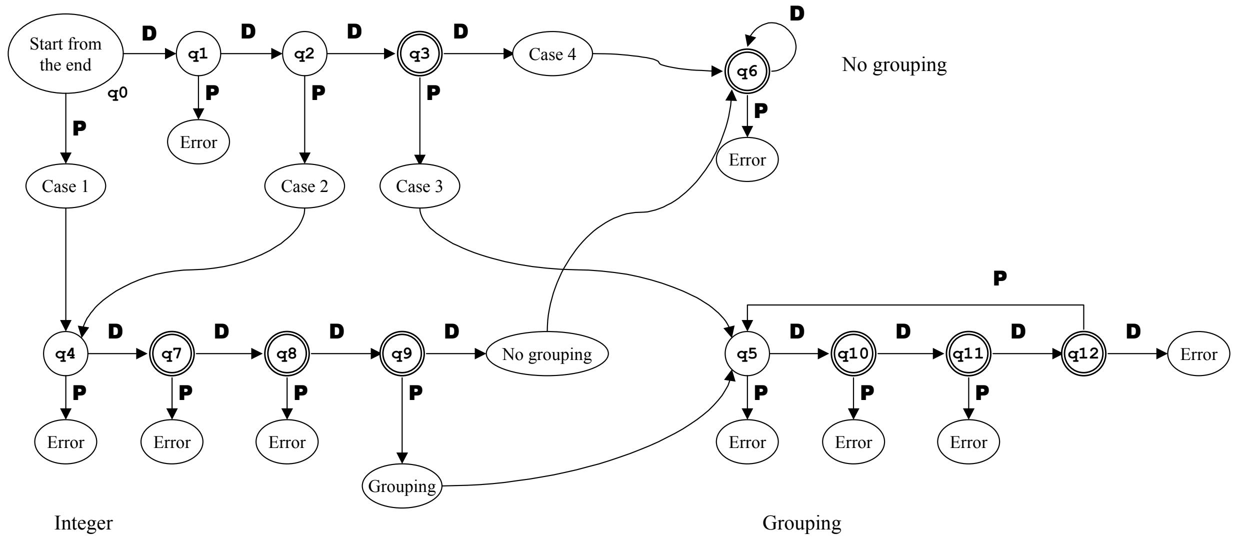

Figure 10: Deterministic Finite Automaton to classify and check amount strings

After the analysis of a few characters, one identifies four possible cases. All four cases are further explained in [49], here is a brief summary:

Case 1 means that the string ends with a punctuation mark. In this case one has to read a number that may or may not have grouping punctuation, this is done using the Integer section of the DFA (begining at q_4 in Figure 10).

Case 2 means that the string has 2 decimal digits, preceded by a punctuation character, therefore one has to read any kind of integer (again state q_4).

Case 3 finds a punctuation mark that precedes the right-most 3 digits. So the string does not have decimal part and the punctuation is understood as the grouping character. Hence the integer part is required to use grouping symbols and is analyzed using the Grouping section of the DFA (state q_5 in Figure 10).

Case 4 is reached after reading four consecutive digits without punctuation. In this case, no decimal part or grouping symbol has been detected. From now on, the system does not expect to see any punctuation and the string will be checked using the DFA for state q_6 in Figure 10.

If the string is completely read without reaching an error state, then the value is accepted and printed on the screen. The following image (Figure 11) shows an example of the prototype analyzing a check from Brazil. The result from reading the courtesy amount is the number that appears on the text box called Value. Accepted strings are printed in a standard way using grouping separators and two decimals.

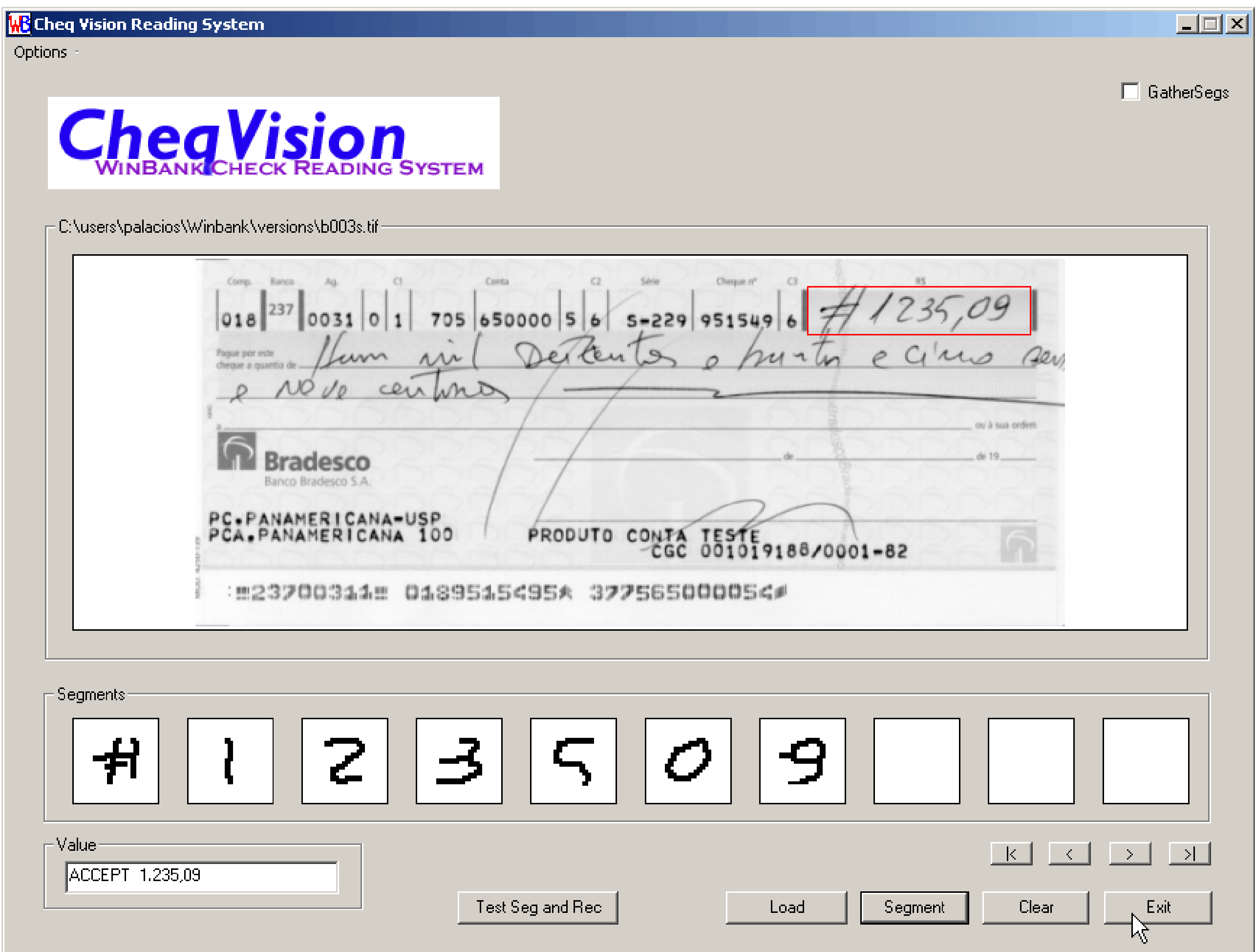

Figure 11: Recognition of a Brazilian check using prototype application

6 Conclusion

This paper has presented a comprehensive approach for check recognition that can be utilized for "reading" checks from Brazilian banks. Some of the algorithms and techniques described in this paper have been applied to read checks from other countries, as well as from other environments that involve reading of handwritten information.

The architecture incorporates multiple modules. The segmentation module takes advantage of the results of the recognition module, via a feedback mechanism. Several splitting algorithms, based on the drop-fall methodology, are used to separate connected or overlapped digits. In the section on recognition of individual characters, a new normalization approach was presented. The process involves slant correction, resize, thinning, rethickening, and a second resize. This normalization approach attained recognition rates ten points better that previous approaches. Single digit recognition is performed using a set of neural networks running in parallel and a structural checker to minimize the likelihood of erroneous readings. Finally, the post processing module takes advantage of contextual information in monetary amounts. This module has been developed as a generalized syntactic checker with the primary objective of reducing the incidence of reading errors. Syntactic rules, described as regular expressions, and the deterministic finite automata tuned to analyze Brazilian checks were discussed in detail.

Digits handwritten on forms by individuals in the U.S. differ significantly in style from those written in paper check in Brazil; this makes it important to train training the neural network with all relevant styles. Another characteristic of the courtesy amount on Brazilian checks is the extensive use of lines and special symbols as delimiters of the amount. Some of these delimiters can be eliminated after size analysis, but the smaller ones will get transmitted into the neural network. Our research highlighted the need to use a recognizer that is able to classify multiple segments and delimiters as part of the endeavor to read Brazilian checks properly.

7 Acknowledgments

The authors thank other members of the project team for the valuable contributions. We also thank representatives of a number of banks located in Brazil for assisting with research related to the specific domain. Parts of the research described in this paper are covered by a patent [25]; we thank the co-inventors, and others who were involved in the process.

8 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] E. S. Angelotti, E. Scalabrin, B. C. Ávila, and F. Bortolozzi, "A Paraconsistent System of Autonomous Agents for Brazilian Bank Check Treatment", *Proceedings 20th International Conference of the Chilean Computer Science Society*, pp. 89-98. 2000

[4] N. Arica and F. T. Yarman-Vural, "An overview of Character Recognition Focused on Off-Line Handwritting", *IEEE Transactions on Systems, Man, and Cybernetics – PartC: Applications and Reviews*, 31(2):216-233, 2001.

[5] 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.

[6] 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.

[7] H. Baird, H. Bunke, P. S. P. Wang. *Document Analysis and Recognition*, World Scientific, Singapore, 1994.

[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. Wolrd Scientific. ISBN 981-02-2270-X, 1997.

[10] 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.

[11] 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.

[12] 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

[13] 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.

[14] 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.

[15] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*, MIT Press and McGraw-Hill, Cambridge, MA, and New York. 1991.

[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 Board, "Fed Announces Results of Study of the Payments System. First Authoritative Study in 20 Years". Press Release, November 14th , 2001.

[20] J. E. F. Friedl, *Mastering Regular Expressions: Powerful Techniques for Perl and Other Tools*. Ed. O'Reilly and Associates. ISBN: 1-56592-257-3, 1997

[21] M.D. Garris, J.L. Blue, G.T. Candela, P.J. Grother, S.A. Janet, C.L. Wilson. "NIST Form-Based Handprint Recognition System (Release 2.0)", US Dept. of Commerce, Technology Administration. National Institute of Standards and Technology. NISTIR 5959. 1997.

[22] R. C. González, R. E. Woods, *Digital Image Processing*, Second Edition, Prentice Hall. ISBN 0-201-18075-8. 2002

[23] 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

[24] 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.

[25] A. Gupta, M. V. Nagendraprasad, P. S. P. Wang, "System and method for character recognition with normalization", U.S. Patent No. 5633954, 1997.

[26] 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

[27] 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.

[28] 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.

[29] F. Hu, *A syntax Verification Module for the Winbank System*, Advanced Undergraduate Project, Massachusetts Institute of Technology, 1999.

[30] 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.

[31] 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.

[32] 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

[33] 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.

[34] S. Khan, *Character Segmentation Heuristics of Check Amount Verification*. Master Thesis, Massachusetts Institute of Technology. 1998

[35] 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.

[36] 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

[37] F. Kimura and M. Shridhar, "Segmentation-Recognition Algorithm for Zip Code Field Recognition", *Machine Vision and Applications* 5 (3) 199-210. 1992.

[38] 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.

[39] A. L. Koerich and L. L. Lee, "Automatic Extraction of Filled Information from Bankchecks", *IEEE Int. Conference on Systems, Man and Cybernetics. Computational Cybernetics and Simulation.* Vol. 3, pp. 2151-2156, 1997.

[40] 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.

[41] E. Lecolinet and J-V. Moreau, "A New System for Automatic Segmentation and Recognition of Unconstrained Zip Codes", *Proceedings of the Sixth Scandinavian Conference on Image Analysis*, Oulu, Finland, vol 1, pp. 584-92, 1989.

[42] 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.

[43] 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.

[44] O. Matan,H.S. Baird,; J. Bromley, C.J.C. Burges, J.S. Denker, L.D. Jackel, Y. Le Cun, E.P.D. Pednault, W.D Satterfield, C.E. Stenard, T.J. Thompson, "Reading Handwritten Digits: A Zip Code Recognition System", *Computer*, Vol. 25(7), pp. 59-64, 1992

[45] B. T. Mitchell and A. M. Gillies, “A Model-Based Computer Vision System for Recognizing Handwritten Zip Codes”, *Machine Vision & Applications*, Vol 2 (4), pp 231-243, 1989.

[46] M. V. Nagendraprasad, P. S. P. Wang, and A. Gupta "Algorithms for Thinning and Rethickening Digital Patterns" *Digital Signal Processing*, 3, 97-102 , 1993.

[47] Official page on "Questions & Answers on the euro and European Economic and Monetary Union". http://europa.eu.int/euro/quest/, Jan 2002

[48] 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.

[49] R. Palacios and A. Gupta, "A System for Processing Handwritten Bank Checks Automatically", submitted to *Image and Vision Computing*. 2002.

[50] R. Palacios and A. Gupta, "Training Neural Networks for Reading Handwritten Amounts on Checks", accepted for the *International Conference on Signal, Speech and Image Processing* (WSEAS-ICOSSIP 2002).

[51] M. Prundaru and I. Prundaru, “Syntactic Handwritten Numeral Recognition by Multi-Dimensional Gramars”, *Proceedings of SPIE: The International Society for Optical Engineering*, Vol 3365, pp. 302-309, 1998.

[52] J. Punnoose, *An Improved Segmentation Module for Identification of Handwritten Numerals*, Master Thesis, Massachusetts Institute of Technology. 1999.

[53] A. F. R. Rahman, R. Rahman, M. C. Fairhurst, "Recognition of Handwritten Bengali Characters: a Novel Multistage Approach", *Pattern Recognition*, Vol 35, pp. 997-1006. 2002.

[54] A. Da Rocha Gesualdi, M. P. De Albuquerque, J. M. Seixas, L. P. Caloba, and M. P. De Albuquerque, "Recognition of characters in plates of private Brazilian vehicles using neural networks", *Proceedings XIII Brazilian Symposium on Computer Graphics and Image Processing*, pp 333-333. 2000.

[55] E. Roman, A. Gupta, J. Riordan."Integration of Traditional Imaging, Expert Systems, and Neural Network Techniques for Enhanced Recognition of Handwritten Information." IFSRC Work Paper #124-90. 1990.

[56] A. Sinha, *An Improved Recognition Module for the Identification of Handwritten Digits*, Master Thesis, Massachusetts Institute of Technology. 1999.

[57] P. L. Sparks, *A Hybrid Method for Segmenting Numeric Character Strings*, Master Thesis, Massachusetts Institute of Technology. 1990.

[58] P. L. Sparks and M. V. Nagendraprasad, and A. Gupta, "An Algorithm for Segmenting Handwritten Numerals". *Second Internationl Conference on Automation, Robotics, and Computer Vision*, Singapore, 1992.

[59] 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.

[60] 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

[61] 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.

[62] 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.

[63] P. S. P. Wang, and Y. Y. Zhang, "A fast and flexible thinning algorithm", *IEEE Transactions on Computers*, 38(5), 1989

[64] P. S. P. Wang. *Character and Handwriting Recognition: Expanding Frontiers*. Ed. World Scientific. ISBN 981-02-0710-7, 1991.