RECOGNITION OF COURTESY AMOUNTS ON BANK CHECKS BASED ON A SEGMENTATION APPROACH

Li Qiang Zhang

A thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

September 2001

0-612-64090-6

Abstract

Recognition of Courtesy Amounts on Bank Checks based on a Segmentation Approach

Li Qiang Zhang

A segmentation based courtesy amount recognition (CAR) system is presented in this thesis. This system consists of four modules, which are preprocessing, segmentation, recognition and syntactical analysis. The noise is first removed in the preprocessing module. Then, the segmentation module seeks to extract characters (digits, punctuation marks and so on) from the the numeral strings. The recognition module aims at classifying each pattern into one of the ten numerals ('0' - '9') and double zeros ('00'), rejecting ambiguous patterns. Finally, the syntactical analysis module parses the recognition results to provide an acceptable courtesy amount.

Two sets of features, *shape* features and *spatial* features, are developed to locate the punctuation marks. In addition, a hypotheses-then-verification approach for segmentation of the numeral strings is introduced. The emphasis is on finding all possible segmentation hypotheses and then verify them. A two-level segmentation module is proposed, namely the global segmentation level and the local segmentation level. At the global segmentation level, the intent is to find the potential broken numerals and group them together. On the contrary, the local segmentation level seeks to split the touching digits.

Two classifiers are combined into the recognition module. The isolated digit classifier divides the input patterns into ten numerals ('0' - '9'), while the holistic double zeros classifier intends to recognize the cursive and touching double zeros as an atomic symbol.

The proposed courtesy amount recognition system has been trained on the database of CENPARMI checks and tested on the database of real checks. The system reads 66.5% real checks correctly at 0% misreading rate.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Dr. Ching Y. Suen and Dr. Tonis Kasvand for their guidance to my research work. Without their support and constant encouragement, this work could have been impossible to finish.

Specials thanks to Nick Strathy who provided the image processing code libraries and isolated digit classifier. Also, I would like to thank Christine Nadal who provided me with technical support. Their help has been of significant importance in making this research possible.

My great appreciation goes to my colleagues and graduate students: Dr. Jinho Kim, Dr. Kye Kyung Kim, Beverley Abramovitz, Jianxiong Dong, Qizhi Xu, Dr. Xuejing Wu, Dr. Xiaoping Chen, Dr. Yuntao Qian, Qizhi Xu, Jun Zhou, Zheng Li, Andrea Barretto de Souza and Guiling Guo who have given me their help and shared my emotions.

I am indebted to my parents who brought me to this world and raised me with endless love. To my brother, I am so grateful for his love and support. I am very thankful to have had such a warm family in my life.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Challenge 2

1.3 State of the art 3

1.3.1 A2iA 4

1.3.2 LIREC 4

1.3.3 ParaScript 5

1.3.4 Previous work at CENPARMI 5

2 System Overview **7**

2.1 Courtesy Amount (CA) Analysis 7

2.2 Principles of CAR system 9

2.3 The proposal 11

3 Punctuation Recognition **13**

3.1 Description of the Image 13

3.2 Punctuation Analysis 14

3.3 Feature Description 15

3.3.1 Shape Features 16

3.3.2 Spatial Features 19

3.4 Feature Combinations 22

4 Segmentation **23**

4.1 Global Segmentation Level 24

4.2 Local Segmentation Level 27

5 Recognition and Verification **32**

5.1 Isolated Digit Classifier (IDC) 32

5.2 Double Zeros Classifier (DZC) 33

5.3 Evaluation and Verification 36

6 Experimental Results **39**

6.1 Database 39

6.1.1 CENPARMI Checks 39

6.1.2 Real Checks 40

6.2 Working Environment 41

6.3 Result of Segmentation 42

6.3.1 Result on Training Set 42

6.3.2 Result on Test Set 42

6.3.3 Performance Analysis of Segmentation 43

6.4 System Integration 47

6.4.1 Numeral recognition 47

6.4.2 Recognition results on real checks 48

6.4.3 Performance Analysis 49

7 Conclusion **54**

7.1 Contributions 54

7.2 Strengths and Weaknesses 55

7.3 Future Work 56

Bibliography **58**

List of Tables

1 Global Recognition Performance of LIRECHéque System. 5
2 Performance of CENPARMI System. 6
3 Segmentation hypotheses on the training set, 645 samples. 42
4 Segmentation hypotheses on the test set, 400 samples. 43
5 Erroneous segmentation on CENPARMI checks and real checks. . . . 45
6 Performance of numeral recognition on CENPARMI checks and real checks 48
7 Recognition results on 400 real checks. 48
8 Courtesy Amount Recognition. 49

List of Figures

1 A sample of Candian bank check. 7
2 Illustration of Writing Styles. 8
3 System Architecture 11
4 Examples of corner detections 14
5 Samples of Courtesy Amount 15
6 Illustration of *Flat* feature 17
7 Illustration of *Slope* feature 18
8 Illustration of *Below_half* feature 20
9 Illustration of *Euclidean distance* between two components 21
10 Examples of overlap feature 21
11 Illustration of the hypotheses tree. 25
12 Exit SCPs by extending the contours through a stroke. 28
13 The horizontal pixel distance feature array of a character "8". 33
14 Touching Situation of Double Zeros 34
15 Diagram of One Hidden Layer Neural Network. 36
16 Hierarchical Structure of Recognition Module. 37
17 Illustration of Recognition and Verification. 37
18 Sample of a CENPARMI check. 40
19 Sample of a filled CENPARMI check. 41
20 Sample of poor image quality. 44
21 Failure to detect *Significant Contour Points*. 44
22 Sample of failure in punctuation detection. 45
23 Illustration of acceptable segmentations. 46

24 Illustration of under-segmentations. 46
25 Illustration of good recognition results. 50
26 Problem of numeral recognition. 51
27 Problem of '$' recognition. 52
28 An example of ambiguous digit strings. 53

Chapter 1

Introduction

Automatic check processing has become an active research field in document analysis and handwriting recognition due to its commercial potential and the nature of the academic challenge it presents. The objective of this thesis is to develop a system for the automatic reading of courtesy amounts on checks. To this end, several processing modules have been integrated in the system including the preprocessing module, the segmentation module, the isolated digit recognizer and the parser.

1.1 Motivation

Despite the rapid growth of E-commerce which seems to be taking us into a paperless future, personal checks remain America's favorite method of payment since they are considered to be secure, confidential and convenient. Investigations show that 83% of Americans prefer to make their non-grocery type payments by means of personal checks [1]. It is estimated that 69 billion checks are written and change hands annually and this already formidable volume presently increases by 2 billion checks in the United States every year in spite of the popularity of ATMs, credit cards and home banking [2].

The labor costs entailed by processing checks have imposed a huge financial burden on banks and utility companies, where an immense number of checks are treated every day. The protocol of processing checks in these institutions requires that one operator keys them in and another person verifies them. Check processing has consequently become a very labor intensive task. According to a survey, the US banks spend more than 10 billion dollars just to clear all the checks received annually [2].

On the other hand, advances in the technology of optical character recognition (OCR) and document analysis (DA) have led to successful commerical applications of the integrated check reading systems (ICRS). Compared with the labor costs, the investment required for an ICRS would be relatively inexpensive. If half of the received checks had been automatically processed by machine, the US banks would have saved about 5 billion dollars every year. In addition, it has been reported that ICRSs have attained a speed of processing of about 20,000 cheques per hour [3]. Consequently, current ICRSs have already met the two basic industrial requirements of inexpensive production costs and high processing speeds. In conclusion, machine based check reading systems have a wide application potential in banking institutions.

As Bartneck [4] pointed out, the general task of reading systems including ICRSs will be to bridge the gap between the world of paper and conventional writing and the world of computers and electronic processing. This will ensure that ICRSs have a promising future.

1.2 Challenge

The courtesy amount is one of the most important pieces of information on a check and it is mostly written in digits. Courtesy amount recognition (CAR) is a very complex process in which humans have proven more capable than machines.

To meet industrial requirements, a CAR system must solve two main problems:

- *High recognition rate.* The expectation of banks has been set at a 70% reading rate [2].

- *High reliability.* An error rate of less than 0.1% is required for a commercial system [5]. In real application instances, errors are very intolerable and the remedy is very costly since people have to detect them by sorting through a huge quantity of checks by hand.

However, building a successful CAR remains a difficult task due to the complexity and variability of checks, especially given:

- Various check forms, which differ in both the background and in the position of courtesy amount zones.
- Great variations in the writing style, which vary not only in terms of cursive handwriting, but also in terms of the different ways that information is represented.

Actually, human handwriting is affected by many factors such as characteristic, educational background, writing environment, down to the use of writing instrument, etc. The problems posed by courtesy amount recognition are difficult to solve since there are so few constraints permitting an adequate definition of the problem. Even one instance of the same courtesy amount written by a given person is different from another instance. In order to obtain a feasible solution, great efforts should be made to identify hidden contraints.

On the other hand, it is almost impossible for a CAR system to achieve no substitution error. It is necessary to combine several algorithms in the CAR system in order for them to reinforce each other.

1.3 State of the art

Courtesy amount recognition has been studied for more than ten years and many different algorithms and systems have been reported in the literature. Some of these systems have already been applied in banks. In this section, we briefly review these CAR systems as well as the legal amount recognition (LAR) systems. We also note that, since there is no standard test set for all these systems, this review is intended as an introduction the state of the art rather than as a comparision of different systems.

1.3.1 A2iA

A2iA system has been developed in France by Knerr et al. [6], which recognizes both the legal amount and courtesy amount, albeit for French checks only. The system contains two recognition chains. One analyzes the legal amount and the other analyzes the courtesy amount. Then, the results of the two chains are fused and a final decision is made whether to accept or reject the amount.

For both chains, the recognition process is divided into 5 steps: extraction of the image parts containing the amount, preprocessing of the extracted amount images, segmentation of the amounts, recognition of characters, words and amounts, and the final decision.

The courtesy amount recognition has attained a 70.6% recognition rate, the legal amount recognition has reached 43.2%, and the recognition rate of the combination is 64.5% with less than 0.1% substitution.

1.3.2 LIREC

LIRECheques system has been developed by MATRA SYSTEMS & INFORMATION [7]. The courtesy amount module is also divided into 5 steps: preprocessing, segmentation, non-numeral detection, numeral recognition and syntactic analysis.

At the point of recognition, two sets of features are extracted from each binary image: one set is based on concavity measurements, while the other combines exclusively statistical and structural features. Each pattern is thus represented by two feature vectors which form the basic input of a statistical recognition process based on linear discrimination. Two sets of character solutions are then derived from the two sets of features and a unique final decision is made for each digit.

The system has been tested on a test set of 3,374 real checks scanned at 240 dpi. The test set also contains 15% of typed checks. The global recognition performance of the LIRECheques system are list in Table 1, where R(n) and C(n) stand for the percentage of presence and absence respectively of the correct amount hypothesis among the best n rated.

R(1) (%)	R(2) (%)	R(4) (%)	R(8) (%)	R(16) (%)	R(M) (%)	C(M) (%)	Reject (%)
52.94	60.51	65.44	68.79	71.17	73.40	21.05	5.55

Table 1: Global Recognition Performance of LIREChéque System.

1.3.3 ParaScript

ParaScript is a check amount recognition system based on the cross validation of courtesy amounts and legal amounts [5].

In the courtesy amount recognition module, the image is cleaned and segmented. The output image is sent to a numeral recognizer based on a matching input subgraph to graphs of symbol prototypes. Similarly, after legal amount preprocessing and segmentation, words or phrases are sent to a holistic recognizer which processes them without segmentation into characters.

The cross validation procedure takes two ordered lists of answers as an input: one from the courtesy amount recognizer and another from the legal amount recognizer. Each answer in the list has a similarity score showing to what extent the inscription in the input image can be interpreted as such an answer. Then, the answer lists are merged and the similarity score of each answer is calculated using its courtesy amount and legal amount similarity score.

The system has been tested on a test set of 5,000 real checks. At a 1% error rate level, the recognition rates of courtesy amount, legal amount and final answer have achieved 47%, 18% and 67%, respectively.

1.3.4 Previous work at CENPARMI

Check recognition is one of the main fields of study at CENPARMI, where research has been conducted for almost a decade [8, 9]. A system has been designed to recognize the courtesy amount, legal amount and date of each check. This integrated system involves prepocessing, segmentation, classification and syntactical analysis module.

The handwritten items of data are extracted through locating and consequently removing the base lines. The segmentation of data is achieved through splitting the courtesy amount into digit candidates, the legal amount into word candidates and the date into month, day and year divisions. By using a priori knowledge about the relative locations of these items on checks, the corresponding classification module is applied to each item. Finally, the syntactical analysis module parses output data of the classification module.

Table 2 shows the performance of the system for recognizing courtesy amounts and legal amounts without rejection. The recognition rates for courtesy amounts and legal amounts have made this system one of the top performers ever reported.

Item	Train No.	Test No.	Rec. Rate
Courtesy amount recognition	-	400	72
Legal amount rec. (English)	5,223	2,482	92.7
Legal amount rec. (French)	4,513	1,622	86.7

Table 2: Performance of CENPARMI System.

Chapter 2

System Overview

2.1 Courtesy Amount (CA) Analysis

In North America, all bank checks possess a similar format. Figure 1 shows a typical Canadian bank check. The data zone of courtesy amounts is always located in the box on the right of each cheque, with a "$" symbol printed on the left of the box. The box is a restriction zone where courtesy amounts should be written.

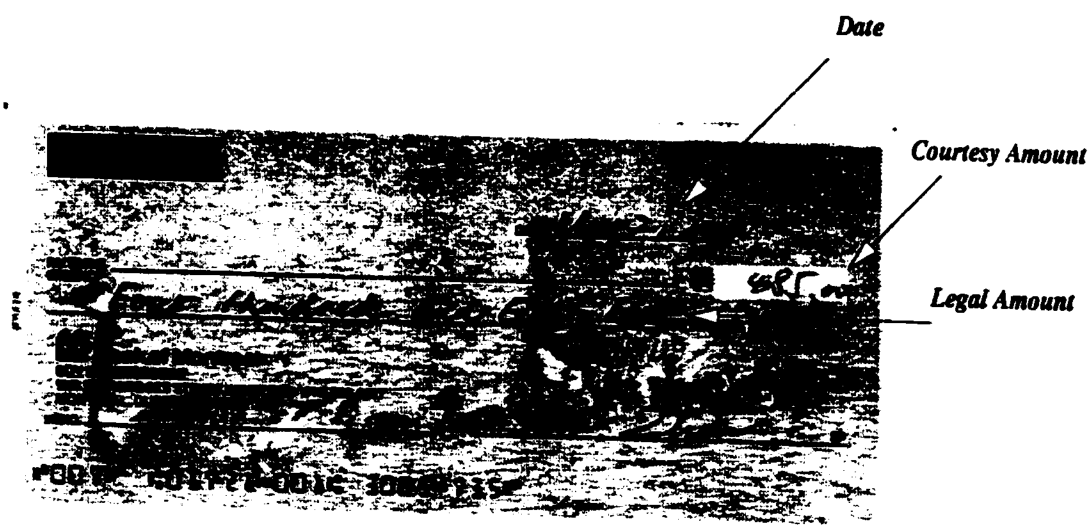


Figure 1: A sample of Candian bank check.

Nevertheless, there is no constraint on writing styles, so the appearance of a courtesy amount varies, depending on each person's writing habit. A courtesy amount usually contains two parts: a dollar part and a decimal part. Punctuation marks [period ('.'), comma (',') and hyphen ('-')] are used to identify the beginning and the end of the amount and to separate these two parts. Also, commas are often used to split every three digits in the dollar part. Some people often use auxilliary elements ("100" and "xx") as denominators in the decimal part. As there are more than 14 classes of symbols (10 digits, 3 punctuation marks and one denominator 'x'), the variations of courtesy amounts are extraordinarily great. There are more than a dozen different styles currently available in writing the courtesy amount in Canada, as shown in Figure 2. Some of them are extremely difficult to cope with, e.g., it is not so easy to distinguish the denominator "100" from the real amount. There is no general writing style, and we can only assume that writers intend to separate the dollar and decimal parts with a punctuation mark, an evident space or a denominator if there is a fractional part.

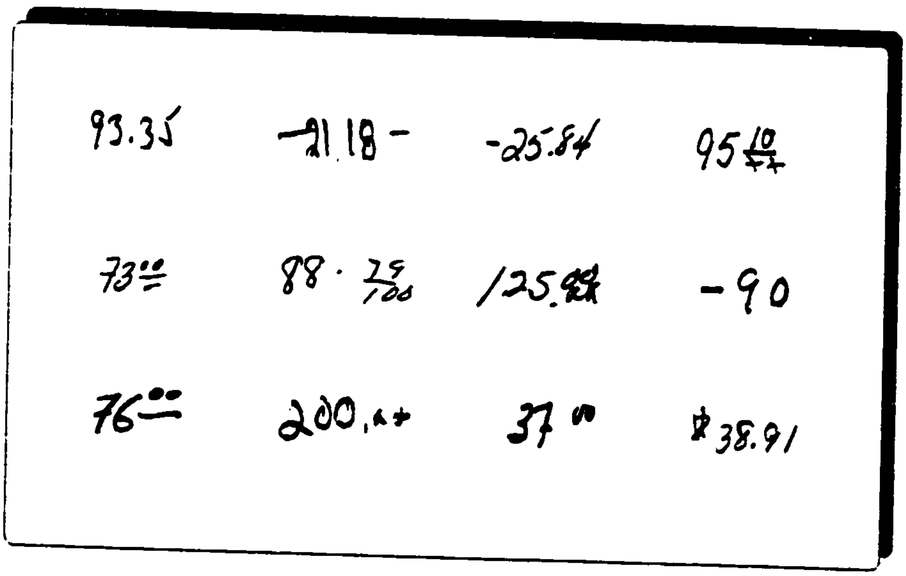


Figure 2: Illustration of Writing Styles.

The objective of courtesy amount recognition is to build a computer-based system to identify symbols in images and to parse them to an amount, so that human beings can be spared from this arduous daily manual labor. However, computer-based recognition is not as *smart* and *flexible* as that of human beings. Its performance totally depends on its algorithms, which provide an extension of human knowledge. In the observation of the real checks, our CAR system should be able to solve the following problems:

- Recognition of ten classes of digits ("0"-"9"). Courtesy amounts are mostly organized into 10 classes of numerals, so that a basic function of CAR is the digit classifier.
- Grouping the broken digits. Since there are a few of connected components in an image, how can the system find these components which are parts of a digit?
- Segmentation of touching digits. How to find a reasonable segmentation cut without destroying the morphological structures of the touching digits?
- Punctuation marks, broken pieces and garbage/noise detection. The appearances of these items are quite similar, relatively small and simple in shape, but how can the system distinguish them?
- Syntactic analysis. After recognition, a string of numerals and punctuation marks are obtained. How can the system distinguish the decimal point through identifying periods and commas and hence separate the dollar and decimal parts?
- Decision-making. How can the system accept a possible result and avoid costly errors?

2.2 Principles of CAR system

A courtesy amount recognition system is a complex system which involves many tasks such as preprocessing, segmentation, recognition and syntactic analysis. To build a structured CAR system, Knerr et al. [6] have proposed the following key elements:

1. *Hierarchical organization.* The data structure of CAR systems is organized as a hierarchy: starting from the pixel level, proceeding via components, and

characters to the amount level. At each hierarchical level, a set of relevant concepts is defined and all objects are described in terms of these concepts.

2. *Modularity* and standards for input/output of the modules. Each recognition task in the hierarchy is implemented through an isolated software module. The data flow between different modules is consequently standardized.

3. *Complementarity*. Since it is impossible to design an algorithm which works optimally, as expected, due to the complexity of courtesy amounts, it is necessary to combine several algorithms to achieve robustness and redundancy.

4. *Adaptivity*. Variations in the data are modeled according to a set of parameters which are learned from the data. For example, classifiers such as neural networks are trained on large databases of character images. The system is easily adaptable to new applications by simply adjusting this set of parameters.

5. *Soft decisions*. These are probability values. A sub-optimal decision made at an early stage can be recovered later by using soft decisions instead of hard decisions, which are used to reject or accept.

6. *Use of several sources of information*, funtioning independently if possible. Generally, the more information used, the better the decision.

As discussed in Chapter 1, the development of a courtesy amount recognition system presents a challenge. Certain problems are intrinsic to the design of the CAR systems: high recognition rate *vs* robustness, speed *vs* complexity of algorithms, and cost *vs* efficency. It is impossible to build a general CAR system for different types of application. To achieve a better performance, the design strategy must be ***domain-specific*** which means the CAR systems should be suitable for specific applications.

2.3 The proposal

A courtesy amount recognition system is proposed, which consists of preprocessing, segmentation, digit recognition and parsing. The function of the preprocessing module is to remove the noise. The segmentation module seeks to extract characters (digits, punctuation marks and so on) from numeral strings. The recognition module aims at classifying every pattern into one of the ten numerals, rejecting ambiguous patterns. Finally, the syntactical analysis module outputs an acceptable courtesy amount from the recognition results. The system architecture is shown in Figure 3.

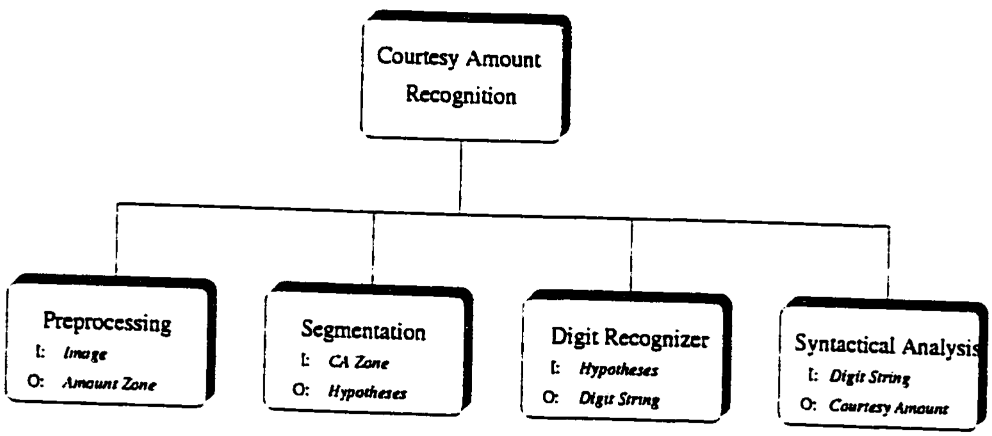


Figure 3: System Architecture

For the purpose of punctuation recognition, a set of features have been introduced. By means of the geometric and spatial aspect of a connected component, three kinds of punctuation marks (period, comma and dash) are identified.

A hypotheses-then-verification approach for the segmentation of courtesy amounts is proposed. The emphasis is on finding all possible segmentation hypotheses and then evaluate them. A two-level strategy is applied in the segmentation module, namely the global segmentation level and the local segmentation level. At the global segmentation level, segmentation hypotheses are generated by combining the connected components and a hypotheses tree is constructed. The hypotheses are verified by the recognition module and touching digits are then split at the local segmentation level.

A double zero classifier is also implemented in this research. The holistic classifier intends to recognize the cursive and touching double zeros as atomic symbols.

These proposed approaches have the following advantages:

- Punctuation marks have been detected at the beginning of the segmentation process by using certain spatial and shape features.
- It is a recognition based segmentation. A number of cuttings are tried but only those hypotheses with a high confidence value are accepted.
- By using the two-level strategy, isolated digits and touching digits are classified. The touching digits are subsequently separated at the local segmentation level.
- The split hypotheses are based on an analysis of the shape of images, so the results are rational.
- Double zeros classifier is able to recognize those cusive touching zeros which are difficult to separate.
- There is no need for any prior information concerning applications.

Chapter 3

Punctuation Recognition

3.1 Description of the Image

The input to our system is a binarized image, which contains a set of black and white pixels. Connected components have been extracted by tracing contour chains around the foreground objects [10]. Each contour consists of a linked list of nodes called a chain. Also, a set of important topological and structural information has been ascertained from the contours:

- *Chain code* of each connected component, which presents the size and shape of the object [11].

- *Bounding box* of each connected component, which presents the two-dimensional size of the object and the relative distances to other objects.

- *Euclidean distance* between two objects, which computes the distance between the closest points in each object.

- *Corners* of contour (see Figure 4, which are the points of high curvature. An N-code or Gallus-Neurath code [12] has been used to determine the corners: the higher the absolute N-code value, the sharper the corner.

- *Stroke tips* of the contour, which are the pixels between two corners. For simplicity, the straight lines between the corners have been applied when they are longer than 8 pixels.

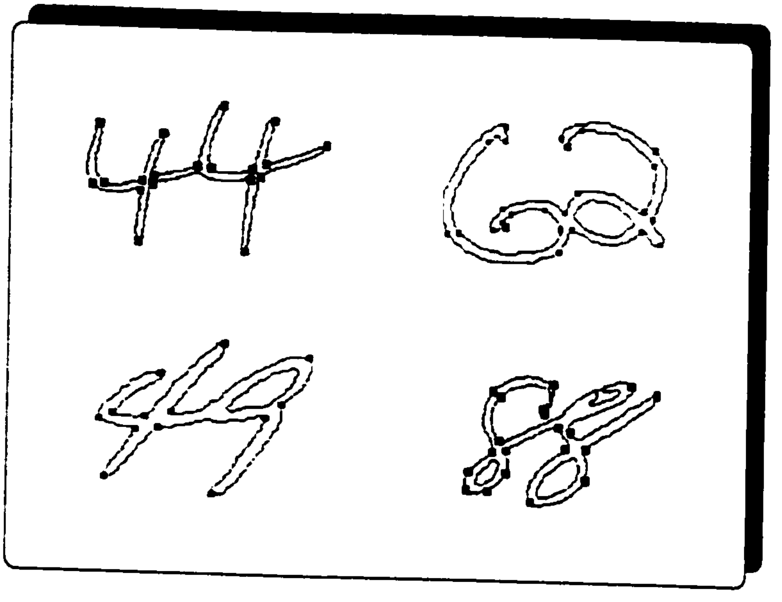


Figure 4: Examples of corner detections

3.2 Punctuation Analysis

There are three types of punctuation marks frequently seen on Canada bank checks: period/decimal point('.'), comma(',') and hyphen/dash('-'). The presence of a decimal point identifies the division between the dollar and decimal parts of the courtesy amounts, commas are often used to split every three digits in the dollar part and the positions of hyphens mark the beginning and end of these amounts. However, it is not necessarily the case that punctuation marks are used only in these given ways. For instance, people sometimes use a comma or obvious spacing instead of a decimal point to separate the dollar and decimal parts. Figure 5 presents some real samples extracted from real bank checks.

On the other hand, broken pieces of digits and garbage/noise are very similar in appearance to punctuation marks. They are small and have simple shapes. As shown in Figure 5, the separated upper horizontal bar of the digit '5' looks like a hyphen,

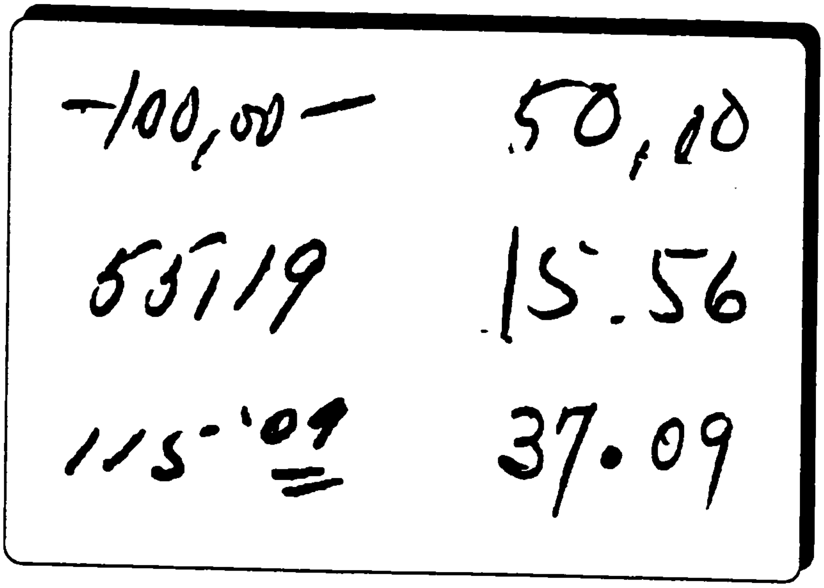


Figure 5: Samples of Courtesy Amount

and certain commas are very similar to "1" and so on. If these punctuation marks and broken pieces cannot be identified correctly, it is likely that worse decisions will be make in later processing, i.e., recognize commas as "1" and misrecognize "5" without the upper horizontal bar. That explains the importance of punctuation recognition and the detection of broken pieces. Fortunately, most people tend to distinguish between punctuation marks and digits, and specific features may be found to identify different types of punctuations and broken pieces.

3.3 Feature Description

Devijiver and Kittler define features as the information extracted from the raw data most relevant to classification purposes, in the sense of minimizing the within-class pattern variability while enhancing the between-class pattern variability [13]. For the purpose of classifying punctuation marks, two categories of features have been included by means of the description of the shape or spatial aspect of a connected component [14].

In this research work, feature algorithms return either a Boolean value or a confidence value. The Boolean value is either 1 or 0, where 1 means the presence of a certain feature, while 0 means its absence. A Boolean value means a binary decision, while conversely, a confidence value provides a fuzzy value. Confidence value is located between 0.0 to 1.0 (inclusive), where 1.0 indicates that the feature is most likely to be present and decreasing values suggest a decreasing probability that the feature is not present.

3.3.1 Shape Features

The shape features indicate the geometric aspects of connected components. Punctuation marks are usually written as simple curve, e.g., a stroke tip or a small loop. The features used in this system include *small*, *flat*, *straightness* and *slope*.

Small

Punctuation marks such as periods and commas are relatively smaller than the size of digits. The *small* feature compares the size of a given component with that of the others. The sizes of connected components are based on the lengths of the outer contours. The pepper-salt noise is normally very small, and it should consequently be removed first. Next, the average size (*avg_size*) of all the connected components is computed. A value is assigned to the component depending on the ratio of its size (*comp_size*) to the average size:

$$1 - comp\_size/avg\_size \tag{1}$$

It is assigned to 0.0 if *comp_size* is found to be larger than *avg_size*.

Flat

The dash has a certain distinguishable characteristic; namely, it is flat, or it has a relatively low ratio of height to component width. Therefore, R. Fan has proposed the use of a *flat* feature [15]. A confidence value is computed using the ratio of the maximum height of the component to the *width* of the bounding box, as shown in

Figure 6. It is assigned to 0.0 if the *max_height* is larger than the *width*. Otherwise, it returns:

$$1 - max\_height/width \tag{2}$$

, which indicates the probability that the component can be considered "flat".

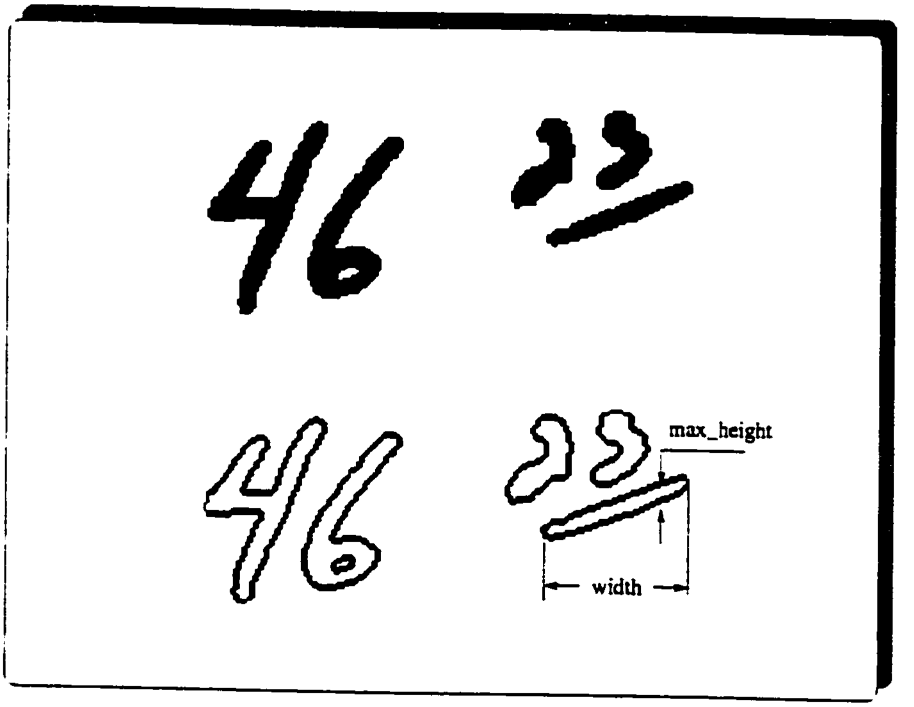


Figure 6: Illustration of *Flat* feature

Slope

The shapes of the dash and the digit "1" are similar: straight and long. Therefore, the *slope* feature has been introduced to identify them. This feature computes the cumulative slope of the connected components from the slope of each stroke tip. The slope (*ln_slope*) of each stroke tip is:

$$ln\_slope = \frac{y_1 - y_2}{x_1 - x_2} \tag{3}$$

where (x_1, y_1) and (x_2, y_2) are the end points of a stroke tip (see Figure 7).

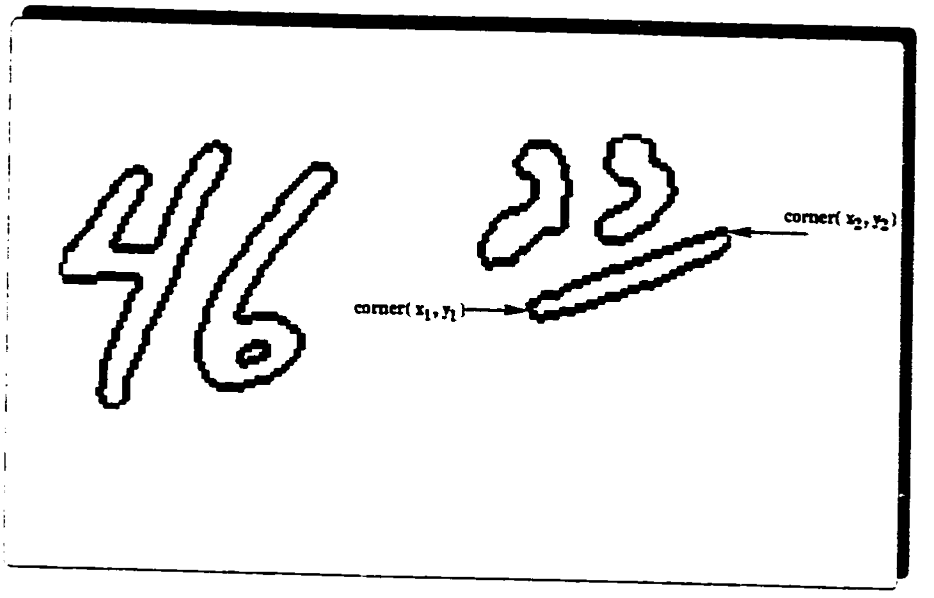


Figure 7: Illustration of *Slope* feature

Hence, the cumulative slope (*sum_slope*) of the connected components is computed by summarizing the slope (*ln_slope*) of all the stroke tips:

$$sum\_slope = \sum_{i=1}^{n} ln\_slope_i * ln\_length_i \tag{4}$$

where n is the number of the stroke tips. A confidence value is subsequently returned depending on the ratio of the overall slope to the length of the chain of the component:

$$sum\_slope/comp\_length \tag{5}$$

where 1.0 indicates the component lies on a horizontal line and the decreasing values indicate the increasing slope of the component.

Straightness

This feature computes the cumulative straightness of the stroke tips of the connected components. First, the length sum (*sum_length*) is computed by summarizing the lengths (*ln_length*) of all the stroke tips,

$$sum\_length = \sum_{i=1}^{n} ln\_length_i \tag{6}$$

where n is the number of the stroke tips. At this point, a confidence value is returned depending on the ratio of the length sum to the length (*comp_length*) of the chain of the component:

$$sum\_length/comp\_length \tag{7}$$

where 1.0 indicates that the component is located on a straight line and the decreasing values indicate the increasing curvature of the component. Among punctuation marks, periods have the highest curvature and dashes have the lowest.

3.3.2 Spatial Features

The spatial features indicate the context aspects of the punctuation marks. The features used in this system involve *below_half*, *distance* and *overlap rate*. These features indicate a connected component's spatial locations in the image and to the neighbor.

Below_half

Punctuations are usually found at the various different locations on the image. Normally, periods and commas are written at a relatively low position, while the hypen is usually situated around the middle of the image. A set of reference lines are therefore developed, which include upper bound, lower bound, *middle* line, as shown in Figure 8.

The *below_half* feature compares the position of the connected component to the *middle* line of the image. It returns 1.0 if the component falls completely below the *middle* line, and returns 0.0 if the component falls completely above the *middle* line. Otherwise, it returns:

$$(middle - comp\_xt)/(comp\_xb - comp\_xt) \tag{8}$$

If the returned value is larger than a threshold, the connencted component is considered as a punctuation candidate.

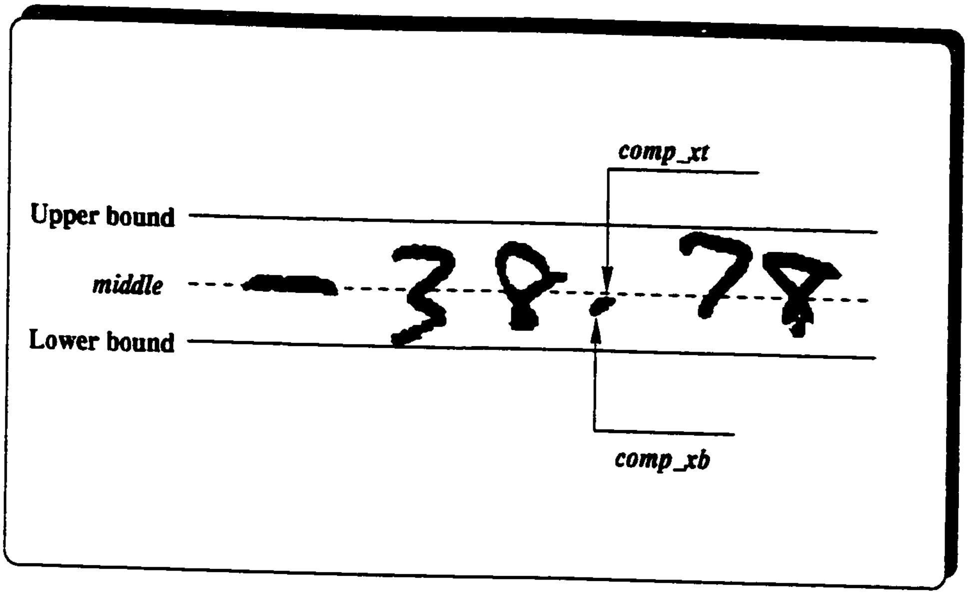


Figure 8: Illustration of *Below_half* feature

Distance

People tend to separate punctuation marks and digits. Consequently, certain spatial distances may be detected between the punctuation mark and the closest components to it. The *distance* feature is measured using the minimum *Euclidean distance* between two neigbored components (see Figure 9), and it is a Boolean value. If the distance is smaller than a threshold, 1 is returned. Otherwise, 0 is returned. This feature can distinguish the broken pieces from punctuation candidates.

Overlap Rate

The *overlap rate* feature is computed using the percentage of the vertically overlapped part of the smaller one of two near components. It returns 0.0 if the components do not overlap each other. If the smaller one is positioned left of the bigger one, it returns:

$$(small\_yr - big\_yl)/(small\_yr - small\_yl); \qquad (9)$$

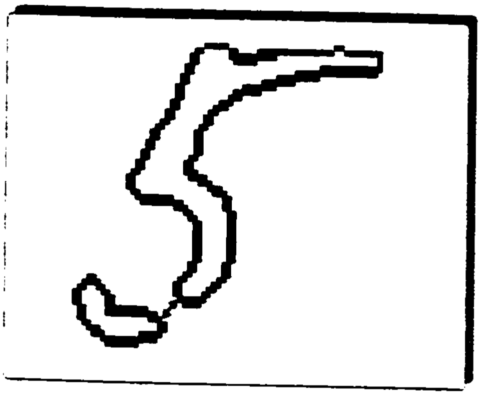


Figure 9: Illustration of *Euclidean distance* between two components

otherwise, it returns:

$$(big\_yr - small\_yl)/(small\_yr - small\_yl). \tag{10}$$

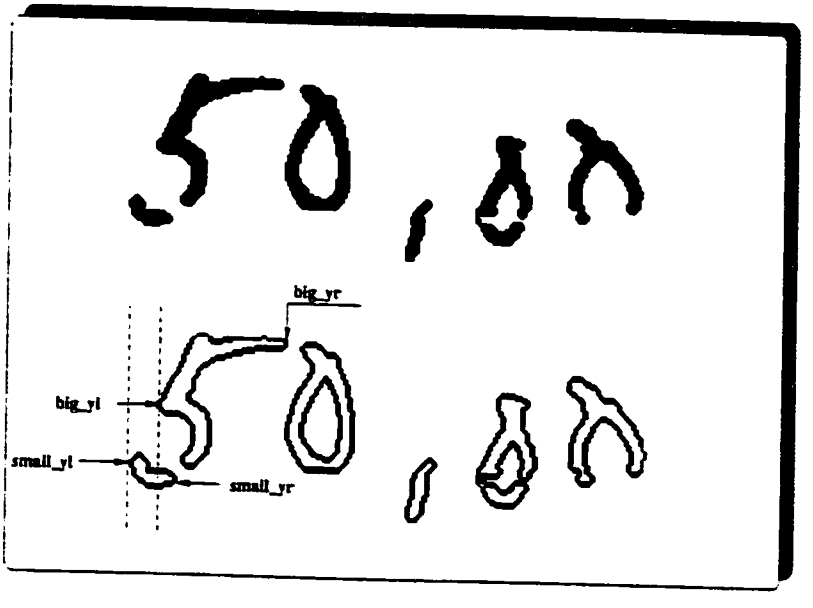


Figure 10: Examples of overlap feature

3.4 Feature Combinations

Different feature combinations are used for each connected component in order to verify if it belongs to one of the specified punctuation classes. The two kinds of measurements are applied; namely, Boolean and confidence values. Boolean values provide a 'hard' decision, such as excluding a connected component from a possible candidate of punctuation. Meanwhile, confidence values are computed to evaluate the likelihood of a connected component to be identified as a certain kind of punctuation marks. Each confidence value represents the percentage of the presence of a distinguishing feature, and the average values of all the different features tend to compensate for each other.

The following 4 sets of features were chosen in the puctuation recognition process and for the detection of broken pieces:

- Period: $small$ and $below\_half$
- Comma: $small$, $straightness$ and $below\_half$
- Dash: $flat$, $straightness$, *slope* and $below\_half$
- Broken pieces: $small$, *distance* and *overlap rate*

Chapter 4

Segmentation

Segmentation is an operation that decomposes an image of a sequence of characters into subimages of individual symbols. It is one of the decision processes within an optical character recognition (OCR) system. This decision reveals whether the decomposed pattern is a meaningful character or some identifiable unit. It is proven that the wrong decision often makes a major contribution to the error rate of the whole system. Casey and Lecolinet [16] have suggested three "pure" strategies for segmentation:

- Dissertion: the decomposition of the image into a sequence of subimages using general features [17, 18]. This process looks for the potential breaks to isolate characters.

- Recognition-based segmentation: a set of candidate breaks is provided using a mobile window of variable width and is validated by recognition results [19, 20]. The candidate breaks are chosen at the elementary level, combined with neighboring strokes, and subsequently checked as to whether they are meaningful.

- Holistic methods: Recognition of words as a whole without segmenting them [21, 22]. Recognition was based on the comparison of a collection of simple features extracted from the whole word against a lexicon of "codes" which represents the "theoretical" shape of the possible words.

Recently, a number of research papers have combined these methods in order to achieve accurate segmentation results [23, 24, 25]

In this research work, a hypotheses-then-verification approach for the segmentation of courtesy amounts is proposed. The emphasis has been on finding all possible segmentation cuttings and related component grouping hypotheses so as to evaluate them. Consequently, the recognition system rejects any courtesy amount that yields one or more rejected symbols or an inadmissible phrase of the courtesy amount.

The segmentation module uses a two level strategy, employing global and local segmentation levels. At the global segmentation level, connected components are detected and different grouping hypotheses are generated by the analysis of the whole image. Each hypothesis is verified by a digit recognizer. The rejected hypothesis is sequentially inputted to the local segmentation level, where it is divided into two sub-blocks or more by the analysis of contour points. Then, each sub-block is verified again.

4.1 Global Segmentation Level

At this level, connected components are found by tracing contours [10] and then sorted from left to right. These components could be isolated digits, broken pieces, punctuation marks, touching digits or garbage. Therefore, punctuation marks and garbage are first detected and then removed from the image.

An "undersegmentation" strategy is applied at this level. This strategy intends to find separated parts of a digit and group them back into a dynamic sub-block of variable width so that the correct segmentation boundary is included inside this sub-block.

Subsequently, a hypotheses tree is constructed by means of a combination of neighboring connected components, as illustrated in Figure 11. The basic strategy of combination is to exhaust all probable global segmentations. Inside this tree, the root is the original image and its children are the subimages extracted from left to right. Next, the subimages are removed from the original image and the children of these are the subimages extracted from the remaining image, and so on. Every node

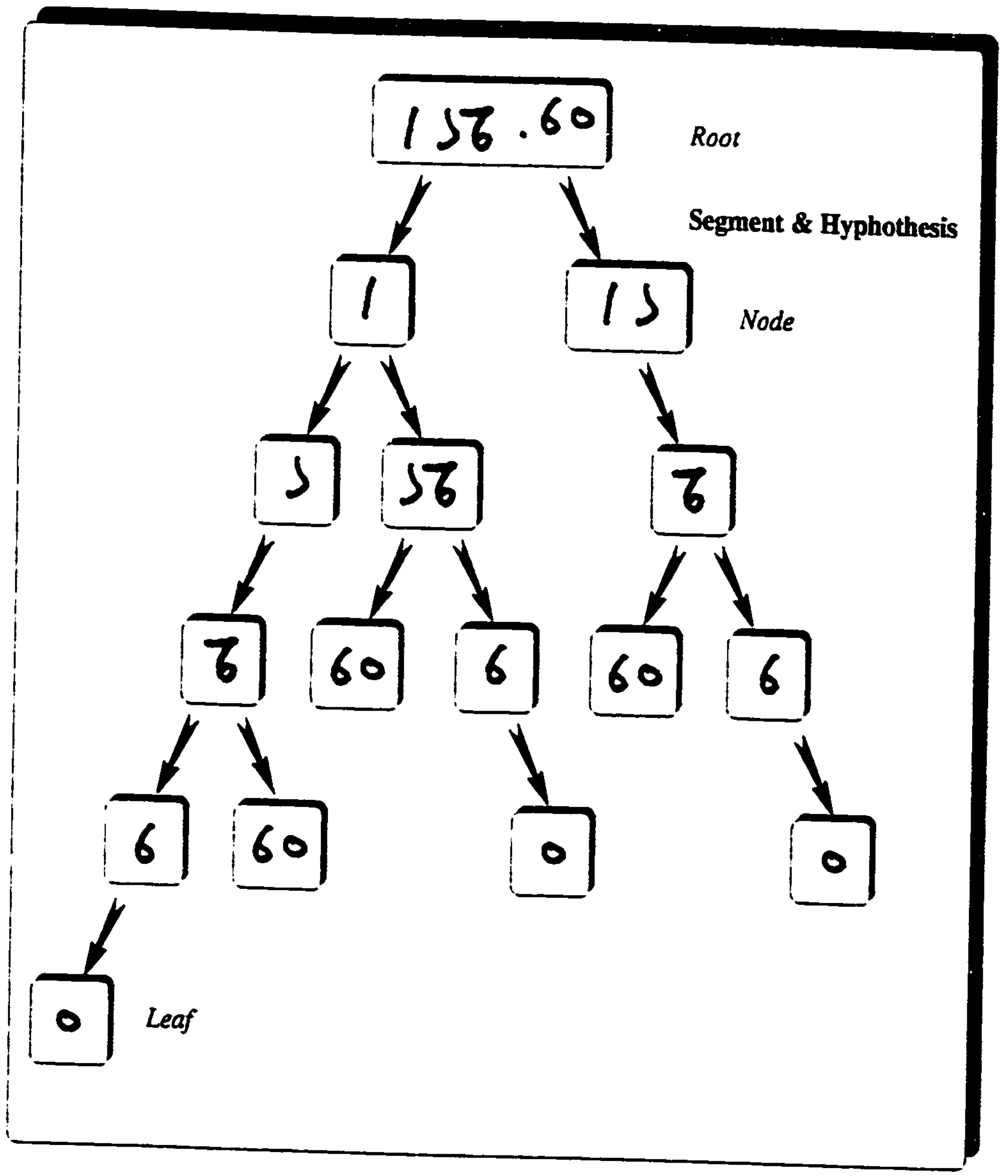


Figure 11: Illustration of the hypotheses tree.

in this tree is a sub-block of an image extracted from the original image and a path from the root to the leaf is a hypothesis. Therefore, a hypothesis is a list of sub-blocks which are, in turn, the combination of connected components of the original image. Two principles for extracting a sub-block of images may be derived from the:

1. *Width* of the block. It should be a reasonable width. Otherwise, the subimage might contain more than one character and the hypotheses tree then becomes too big.

2. *Distance* between two neighboring components. If it is larger than a threshold, the components do not likely belong to the same character.

A recursive algorithm for generating the hypotheses tree, which is named *GenerateHypotheseTree*, is summarized as follows. The initial call of the algorithm is *GenerateHypotheseTree(1, root)*, where *root* is the root of the hypotheses tree.

n is the total number of sorted connected components;
yl_i and yr_i are the y coordinates of the left and right bound of
the ith connected components, respectively;
th_1 is a fixed threshold of the width of the subblock;
th_2 is a fixed threshold of the distance between two neighboring components;

(1) **procedure** *GenerateHypotheseTree* (i, rt)

(2) **begin**

(3) **if** $i > n$ **then return**;

(4) $j \leftarrow i$;

(5) **while** $j \leq n$ **do**

(6) **if** $yr_j - yl_i \leq th_1$ **then**

(7) **if** $j > i$ **and** $yl_j - yr_{j-1} > th_2$ **then return**;

(8) group the connected components i to j into a subblock *node*;

(9) assign *node* as a child of *rt*;

(10) **call** *GenerateHypotheseTree* (j+*1*, *node*);

(11) **else return**;

(12) $j \leftarrow j + 1$;

(13) **end do**;

(14) **end.**

The complexity of all recusive algorithms increases exponentially. Therefore, two thresholds, th_1 and th_2, are set in *GenerateHypotheseTree*() in order to bound the exponential increase of the hypotheses tree. Otherwises, the complexity of the algorithm is $O(2^n)$!

After the hypotheses tree is generated, all global level segmentation hypotheses can be exported by using a simple pre-order traversal. Then, each sub-block insides this hypotheses tree is verified by a digit recognizer and this stage is called *early recognition.*

An obvious advantage of this hypotheses tree is that considerable meomery is saved. More importantly, in stead of verifying all hypotheses, it is simple and efficent to recognize the nodes of this hypothese tree since every hypothesis is a route from the root to the leaf. Hence, very considerable processing time is saved.

4.2 Local Segmentation Level

At the global segmentation level, each node of the generated hypotheses tree is verified by a recognizer. As shown in Figure 11, for each node, it may contain a complete character, a part of a character or more than a character. If it is well recognized, it is accepted as a digit. Otherwise, this block might contain touching digits and it is sent to the local segmentation level, where the segmentor attempts to split it into isolated digits.

On the constrary to the "undersegmentation" strategy at the global segmentation level, an "oversegmentation" strategy is used here. The image is considered as two or more touching digits and a few of cuts are tried in the potiential touching points. Finally, split parts are tested by the recognition module and the cut giving a high confidence value is accepted.

The segmentation algorithm is based on image contour analysis. The identification of *significant contour points* (SCPs) on the outer contour of touching digits is a key element of the segmentation. Three types of SCPs are recognized [10]:

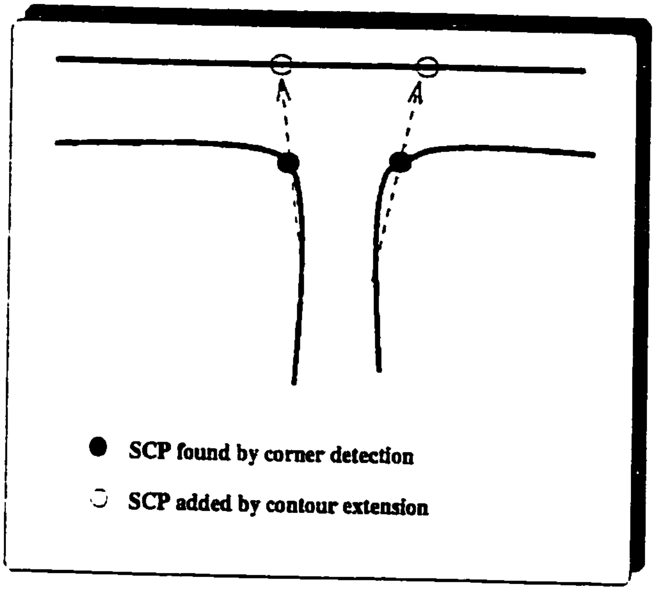


Figure 12: Exit SCPs by extending the contours through a stroke.

1. The corner points of the outer contour.
2. The maximum (minimum) of each mountain (valley) regions of outer contour, as shown in Figure 12.
3. The exit points of the imaginary straight lines by extending the contour through the stroke at the concave corners in mountains and valleys, as seen in Figure 12.

Then, the SCPs are sorted according to nine measures of each pair [10]:

1. A fixed number of credits for each mountain/valley pair.
2. A fixed number of credits for each SCP found by corner detection.
3. Credits for the proximity to each other of the points in the pair.
4. Credits for the sharpness of the concavity at each of the SCPs.
5. Credits for the proximity, where applicable, of an SCP's mountain (valley) to bottom (top) of the image.
6. Credits, where applicable, for the distance of an SCP from the bottom (top) of its mountain (valley).
7. Credits, where applicable, for the degree to which a valley corner is above a mountain corner in the image.
8. Credits for the degree to which stroke pixels outnumber background pixels in an imaginary straight line drawn between the pair.
9. Credits for the nearness of the pair to the left hand side of the image.

Credit values are normalized according to the height of the bounding box of the given string. The SCP pairs are sorted from the highest score to the lowest, so that the more favourable candidates appear towards the front of the list.

The cutting path is determined by a straight line which passes through the foreground pixels or by the contour in regions where the line passes through the background pixels. Once cutting paths have been determined, the single connected stroke component can be split into several separated components. The split algorithm is briefly described as follows:

image is the input block to be split;
N is the maximum number of sub-images to be split;

(1) **procedure** *SplitHypotheses* (*image*)

(2) **begin**

```
i ← 0;
while i ≤ N do
    split the image into two sub-images using an SCP pair;
    recognize the two sub-images;
    if the sub-images are both recognized then return;
    if one of the sub-images is recognized then
        i ← i + 1;
        remove the sub-images from image;
    end if;
    get another SCP pair;
    if all SCP pairs are tried then return;
end do;
end.
```

In general, the image is cut into left and right parts and then verified by the recognition module. If a cut produces a component too small to be considered a digit, it is not accepted. A sub-image is considered to be recognized when its confidence value is greater than a specific threshold. Step by step, the recognized sub-image is reomved from the whole image and the remaining image is split again. The following cases can happen:

1. Both sub-images are recognized. The recognition result of these two sub-images and separated sub-images in the previous step are accepted.

2. Only one sub-image is recognized. The recognized part is removed from the image and the remaining part is split again.

3. Both sub-images are rejected. The remaining image is split again using another pairs of SCP.

4. All SCP pairs are used. If the remaining image is rejected, the whole image, *image*, is rejected.

Chapter 5

Recognition and Verification

The output of the segmentation module is a hypotheses tree, which contains a list of hypotheses. Each hypothesis is a combination of sub-images of the original courtesy amount image. Each subimage is then sent to the digit recognition module which finally derives a list of possible digits and corresponding confidence values.

In our CAR system, two classifiers are applied in the recognition stage: one is an isolated digit classifier and the other is a touching double zeros classifier. The architectures of the two classifiers are very similar: a combination of three simpler neural networks and majority voting strategies are used. The performance of the classifiers is very critical since they form the base line of the system. Once misrecognition occurs, it can hardly be recovered. In fact, a good classifier is not only able to recognize digits but able to reject garbage as well.

5.1 Isolated Digit Classifier (IDC)

The isolated digit classifier applied in our CAR system was implemented by Strathy [8] and it is summarized as follows:

1. *Preprocessing.* Thinning and normalizing of the input image.

2. *Feature Extraction.* Two sets of features have been extracted: the pixel distance features and the size-normalized image pixels. Pixel distance features (PDF)

are measured in the horizontal and vertical directions for each pixel, shown in Figure 13. The horizontal pixel distance features give the direction and distance from that pixel to the nearest black pixel in the same horizontal scan line. The direction is indicated by the sign, positive if the nearest black pixel is to the left, negative if to the right. Vertical PDFs are analogous.

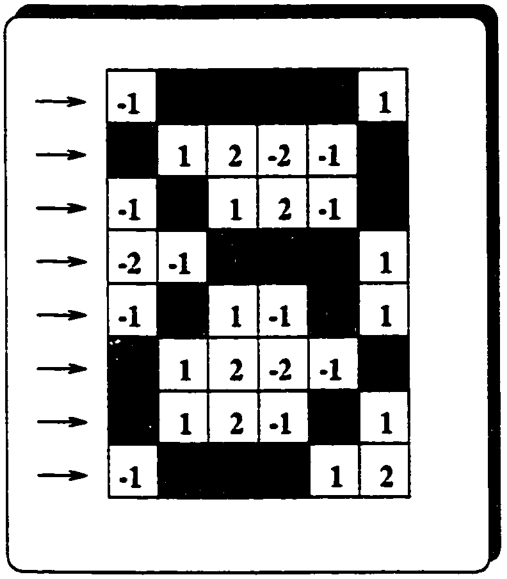


Figure 13: The horizontal pixel distance feature array of a character "8".

3. *Architecture.* Each network has three layers. One network is fully connected, having 336 input, 70 hidden and 10 output units. The other two networks are not fully connected, having 256 input, 560 hidden and 10 output units.

4. *Training and Results.* Each network was trained on about 450,000 isolated digit images and a recognition score of 99.07% was obtained [9].

5.2 Double Zeros Classifier (DZC)

Double zeros are frequently seen on bank checks, especially in the cent parts. Because the cent part is usually less important, double zeros are often written less carefully;

namely, touching and cursive. Seven categories of touching double zeros can be observed, as Figure 14 shows.

Touching Types	Examples
Single Point Touching Only one touching point between the contours of two zeros.	
Multiple Point Touching More than one touching point.	
Ligature Touching An extra ligature stroke connects two zeros.	
Ligature Overlap The ligature overlaps zeros as well as connects them.	
Overlap Two zeros overlap each other.	
Cursive The writing of zeros is sloppy.	
Noise Useless pattern extracted from the image or the pattern is not clear.	

Figure 14: Touching Situation of Double Zeros

It is extremely difficult to split some of those double zeros into two complete zeros. Ligatures, overlaps and cursivenesss cause serious difficulties in segmentation. To tackle this problem, a holistic classifier has been developed to recognize the touching double zeros as an atomic symbol. This classifier is composed of three backpropagation neural networks and majority voting is applied in an effort to attain a high degree of reliability. The architecture of the classifier is:

- Composed of three simpler non-fully connected neural networks with a single hidden layer.

- The first neural network has 512 input, 448 hidden and 11 output units. The dimension of the input feature vetors is 2*16*16. The input image is first normalized to 16*16 and then the horizontal and vertical pixel distance features (PDF) are extracted. The hidden layer is composed of two groups of 14*16 units and each group consists of 14 maps. All the units in a map take the same input from 3*16 feature units and use a set of 49 weights (including the bias). The receptive fields of in the same group overlap, but the *receptive fields* of the two groups do not overlap. The output units are fully connected to the hidden units. The architecture of this network is shown in Figure 15. For simpilicity, this figure only shows the horizontal pixel distance features (PDF) and their *receptive fields* in the hidden layer. Actually, the connections of the input vertical pixel distance features (PDF) and their *receptive fields* are the same.

- The last two networks have 513 input, 448 hidden and 11 output units. The architectures are very similar to the first one. The pixel features and the image density feature are extracted from the size-normalized image. The hidden layers are also composed of two groups of 14 maps of 16 units. All the units in a map takes the same input from 3*16+1 feature units and use the same set of 50 weights. The output units are fully connected to the hidden units.

- The outputs are 10 digits ("0"-"9") and "00".

- A *tanh* activation function is used in the hidden layer.

- A *log* function is used in the output layer.

The training set consists of about 50,000 isolated digits and 1,000 samples of "00". In order to make each class achieve the same prior probability, the "00" samples are duplicated five times. The standard backpropagation algorithmis applied and the training parameters are set as follows:

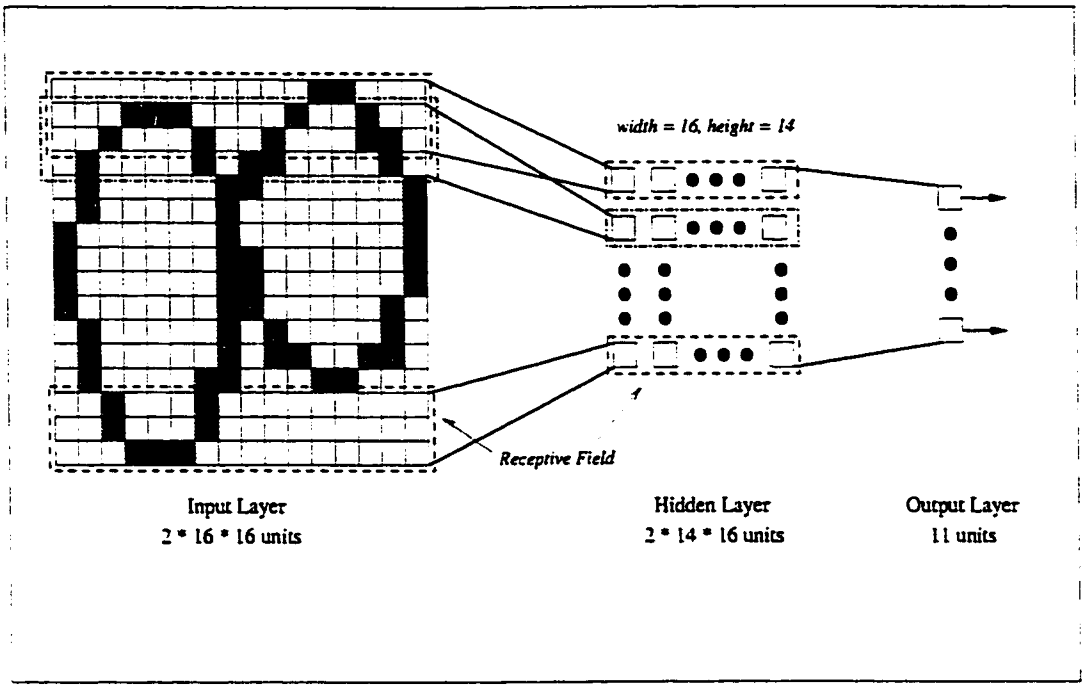


Figure 15: Diagram of One Hidden Layer Neural Network.

- The learning rate growth is set to 1.1.
- The learning momentum is set to 0.2.

The classifier is applied to the symbols rejected by the isolated digit classifier. The threshold of the classifier is set to 0.6 and only the result of "00" is accepted to prevent a misrecognition of other symbols. The hierarchical structure of the two classifiers is present in Figure 16.

5.3 Evaluation and Verification

For each subimage of a hypothesis, the top two possible digits are outputed by the digit recognition module (see Figure 17). A confidence value between 0.0 and 1.0 (inclusive) is also given to each digit.

Therefore, a confidence value is assigned to the hypothesis using the average confidence values of the top choices of all symbols:

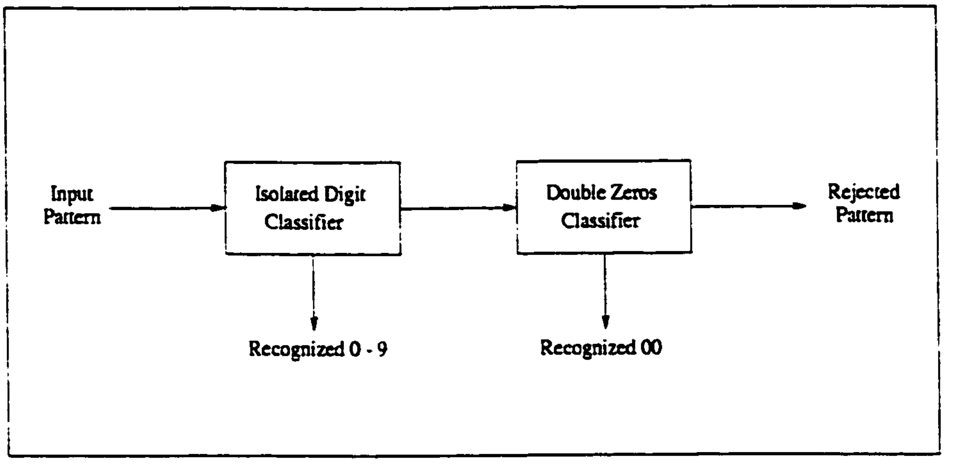


Figure 16: Hierarchical Structure of Recognition Module.

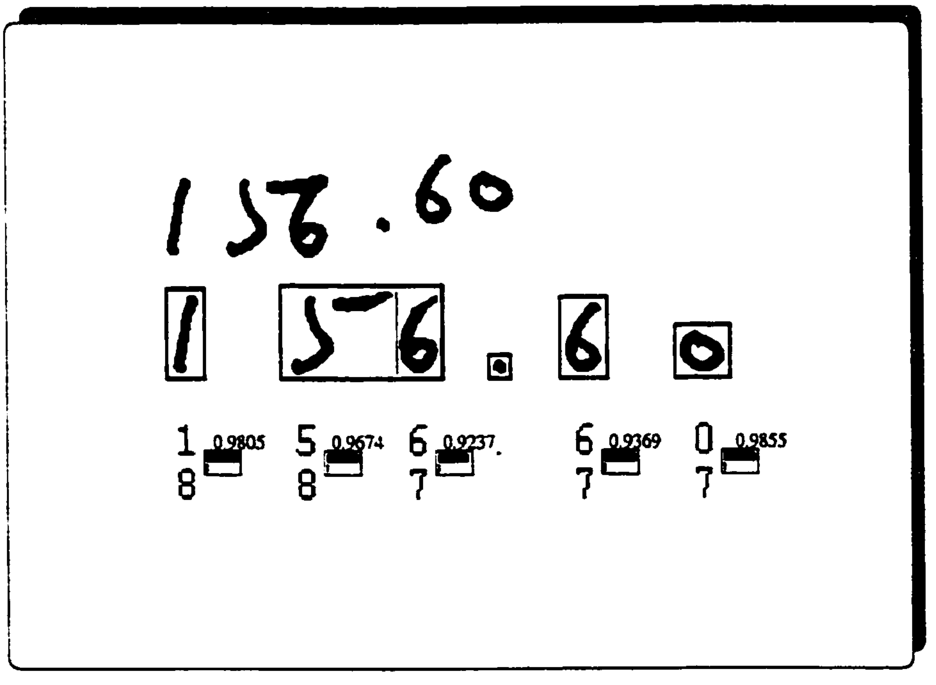


Figure 17: Illustration of Recognition and Verification.

$$conf = \frac{\sum_{i=1}^{n} conf_i^m}{n}, m = 1$$

,

where $conf$ is the confidence value of the hypothesis, $conf_i^m$ is the confidence value of the mth choice of the symbol i and n contained in the number of symbols of this hypothesis. In this CAR system, the top choice of symbols is only considered.

The result of the evaluation stage consists of a list of amount candidates along with various confidence values. This list has been sorted from the highest confidence values to the lowest, so the most probable candidate appears at the front of the list. The final decision in the verification stage is either to accept the top candidate in the list or to reject it. Due to a high reliability expectation, the candidate is rejected if one or more of the following conditions are satisfied:

1. $conf \leq th_1$.
2. $conf_i^1 \leq th_2, i = 1..n$.
3. $conf_i^1 - conf_i^2 \leq th_3$.

Here, $th_{1,2,3}$ are fixed thresholds, which are adjusted according to different applications to achieve the desired substitution rate.

Chapter 6

Experimental Results

6.1 Database

The database of bank check images used by the CENPARMI research group have been collected from two different sources, referred as CENPARMI checks and real bank checks.

6.1.1 CENPARMI Checks

When designing a check recognition system, a large quantity of bank checks written by various people are required to enable the research team to train and test the system adequately. However, for security reasons and protection of the privacy of the bank customers, it is very difficult for the CENPARMI research group to access the real checks from banking institutions.

Therefore, the 'Bank of Concordia' check was designed by the CENPARMI research group [26]. It is similar in appearance to those real checks from banks used in the Montreal area and has a similar layout also, see Figure 18. The background is white and invisible to the scanner with the baselines printed in a light blue color instead of black or dark ink. This facilitates the location of the handwritten information and the removal of the baselines.

After that, the research group visited different classes at Concordia University

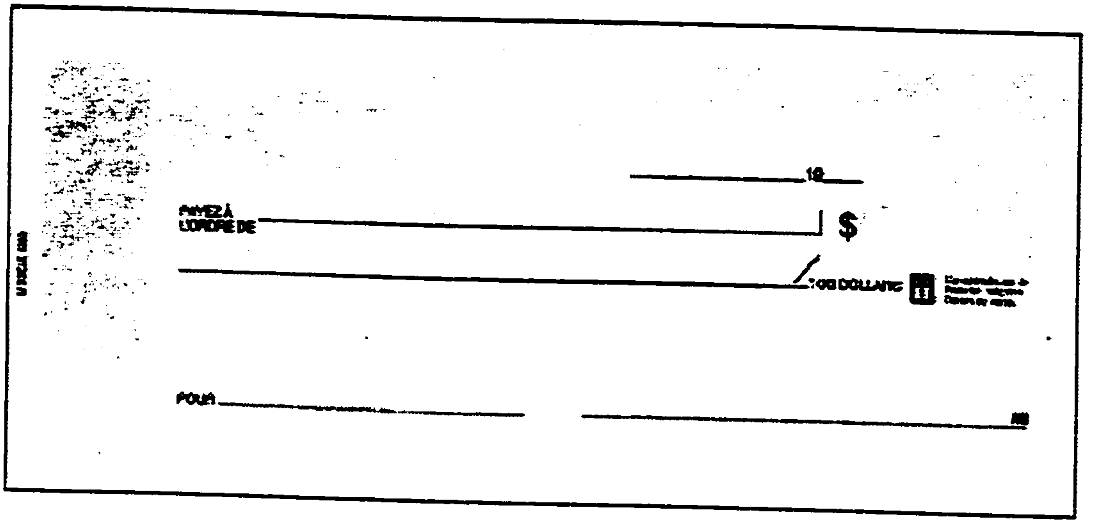


Figure 18: Sample of a CENPARMI check.

and Ecole Polytechnique Montreal [15]. Each student was asked to fill in three CENPARMI checks. They were also asked to fill in a predefined amount on every check but were free to fill in the fields 'date', 'Pay to the order of' and 'signature'. Although there are abnormally 'neat' and 'sloppy' samples, the marjority of the collected data is good. Figure 19 gives a sample of filled CENPARMI checks. Those samples have been scanned at 300 DPI (dots per inch) in grey scale. The written information has been extracted and binarized in a semi-automatic fashion.

A training set of 645 binarized courtesy amount images has been built for this research work. In order to train a robust system, there are not only good data but also some rather poor samples inside this database.

6.1.2 Real Checks

A limited number of filled bank checks were received by the CENPARMI researh group from a local telephone company [15]. These checks were orignally issued by different banks in the Montreal area. Therefore, their layout and geometric measurements are slightly different from each other. Moreover, the printed backgrounds, fonts, lines and boxes have various colors which are visible to the scanner. This makes the image processing much more complicated. Hence, automatic reading of amounts and date

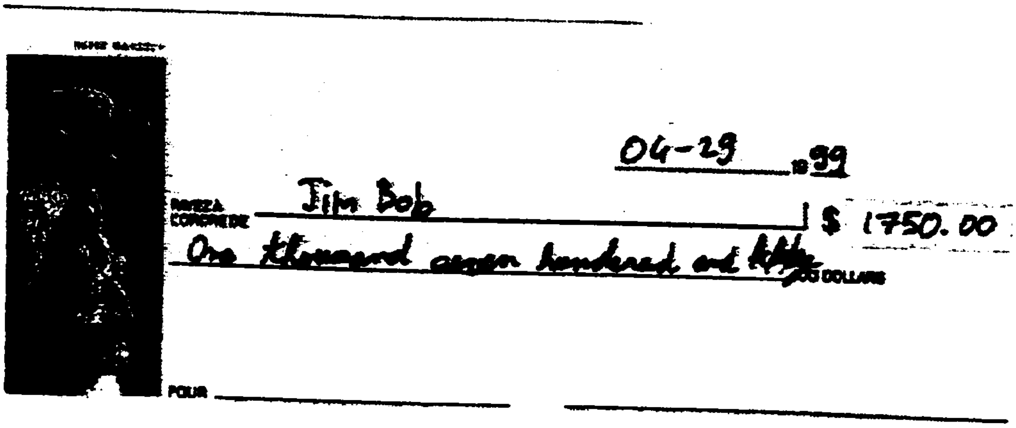


Figure 19: Sample of a filled CENPARMI check.

becomes much more challenging.

On the other hand, these checks were written by the general public in a real-life environment and have real financial background. Therefore, they are suitable for testing the robustness and performance of the integrated check reading system. They were also scanned into a testing database at 300 DPI in grey scale.

In this research work, 400 binarized courtesy amount images have been selected from the database as the test set. The quality of some samples is poor due to the difficulty of image processing.

6.2 Working Environment

The courtesy amount recognition system has been trained and tested on a PC with a Pentium III 500MHz processor and 128 Megabyte memory. The system was implemented using Microsoft Visual C++ 6.0 on Windows NT workstation 4.0 operating system.

6.3 Result of Segmentation

As discussed in Chapter 4, the segmentation is a decision process within this courtesy amount recognition system. Its decisions affect the final results of the system since there is no way for the recognition module to output a correct result on a wrong segmentation hypothesis. Therefore, this part of statistics is provided specifically on the performance of the segmentation module. It must be noted that the performances rely on the opinion of a human being who determined the good segmentation option. The good segmentation options are defined as the cases that the punctuation marks and noises are removed and the entire numeral string are correctly separated into isolated numerals. Hence, it is possible to provide a correct recognition result by the recognition module.

6.3.1 Result on Training Set

The training set consists of 645 CENPARMI check images. Experimental results of segmentation obtained on this set are presented in Table 3, where Rec(n) gives the percentage of cases for which the good semgentation is in the list of the top n hypotheses. As we can see, the top hypotheses cover the good segmentation in 88.1% of these 645 cases and the top 2 hypotheses cover the good segmentation at 91.5%. However, there are 7.3% of the cases that the good segmentation is not in the list of hypotheses.

n	1	2	5	> 5
Rec(n)	88.1%	91.5%	92.7%	7.3%

Table 3: Segmentation hypotheses on the training set, 645 samples.

6.3.2 Result on Test Set

Table 4 shows the experimental results of segmentation on the test set of 400 real check images. Also, a good segmentation performance is obtained. There are 86.0% of the cases that the good segmentation is the top option in the hypotheses list. In 89.3% of the cases, the top 2 hypotheses cover the good segmentation. Nevertheless, due to the high complexity of information extraction from the real checks, the quality of courtesy amount images extracted from real check is not as good as that of the CENPARMI checks. Therefore, there are 9.7% of the cases that the good segmentation is not provided.

n	1	2	5	> 5
Rec(n)	86.0%	89.3%	90.3%	9.7%

Table 4: Segmentation hypotheses on the test set, 400 samples.

6.3.3 Performance Analysis of Segmentation

As Tables 3 and 4 show, the proposed segmentation method provides a significant performance. On the training set, there are 88.1% of the cases that the good segmentation is the top 1 choice of the candidates list while 91.5% of the cases that the good segmentation falls within the top 2 choices of the hypotheses list. Meanwhile, on the test set, there are 86.0% and 89.3% of the cases that the good segmentation is the top 1 choice and the top 2 choices, respectively.

However, there are 7.3% of the cases on the training set that the good segmentation is not in the candidate lists while 9.7% of the cases on the test set do not provide a good segmentation. It can be due to one or more of the following reasons:

- *Case 1* : As mentioned before, due to the difficulty and complexity of data extraction and image preprocessing, the image quality of some samples are very poor in the test set (Figure 20). If an image contains too many connected components, the segmenation process will not only become more difficult, but tends to make errors as well. Therefore, rejection is the best decision.

- *Case 2* : *Significant Contour Points*, which are based on a contour analysis of image, have proven to be very promising in detecting the touching points of

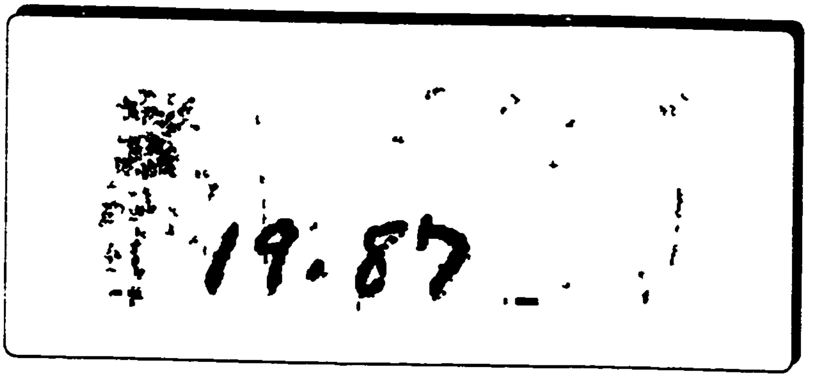


Figure 20: Sample of poor image quality.

digit pairs. However, if the *SCPs* are not detected in some special cases, it is likely that the samples are under-segmented. Refer to the example in Figure 21, the separated upper horizontal bar of the digit '5' connects to the top of the neighboring digit '3'. The system fails to identify the correct *Significant Contour Points* in the middle of the long stroke.

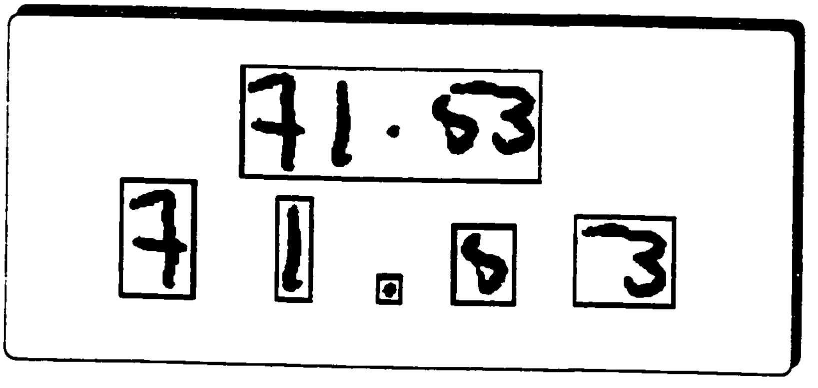


Figure 21: Failure to detect *Significant Contour Points*.

- *Case 3* : Failure in punctuation detection. There exist two kinds of situations: i) fail to detect the punctuation mark and it is recognized as a numeral or a part; ii) detect a numeral or a broken piece as a punctuation mark. Hence, these situations tend to provide an incorrect recognition results. As shown in Figure 22, the system fails to detect the dash since it touches to the numeral '9'.

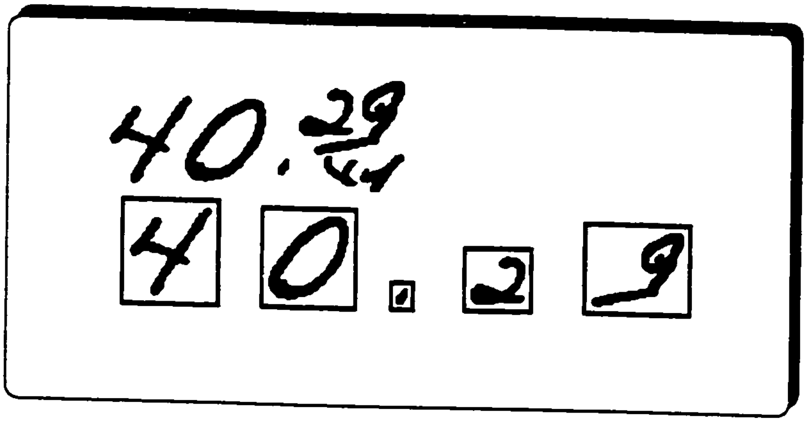


Figure 22: Sample of failure in punctuation detection.

- *Case 4* : On the other hand, a segmentation-then-recognition strategy is applied at the local segmentation level (see the algorithm *SplitHypotheses(image)* in Section 4.2). Once the split subimages are recognized, where the confidence values of subimage are greater than a fixed threshold, they are accepted as isolated digits. As an effort to save processing time, the segmentation module does not try the remaining detected *SCPs* if an acceptable segmentation solution is found. Nevertheless, it is not necessarily the case that this strategy provides the best solution. As Figure 23 shows, a reasonable segmentation is provided, where the segmentor says the touching digits are '2' and '5', but it is rejected by the human supervisor.

Detailed statistics of each case on CENPARMI checks and real checks is given in Table 5.

	Case 1	Case 2	Case 3	Case 4	Mixed
CENPARMI checks	3	17	14	8	5
Real checks	6	13	10	6	4

Table 5: Erroneous segmentation on CENPARMI checks and real checks.

Besides that, around 3% of the cases that the good segmentation is the second option in the candidates list. The first options in all these cases are under-segmentations

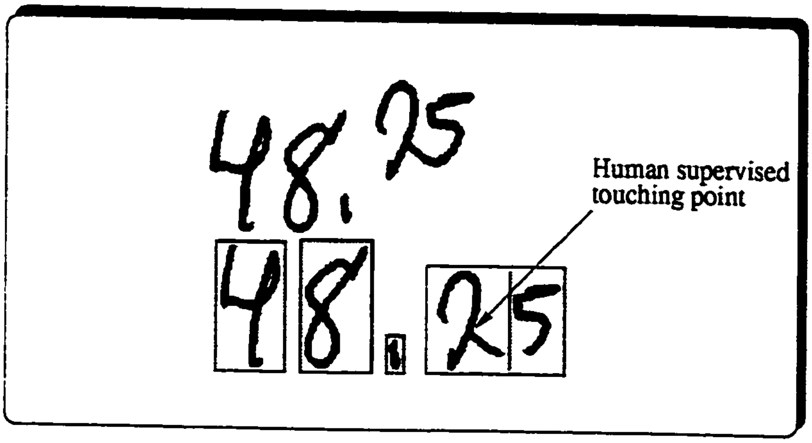


Figure 23: Illustration of acceptable segmentations.

(Figure 24). Their confidence values are slightly higher than those of the good segmentations. This situation is also due to the segmentation-then-recognition strategy. The recognition module gives a high confidence value to the under-segmented sub-blocks which should have been rejected. To solve this problem, the garbage-rejecting ability of the existing recognition module should be improved and also verification modules should be integrated to achieve a more robust system.

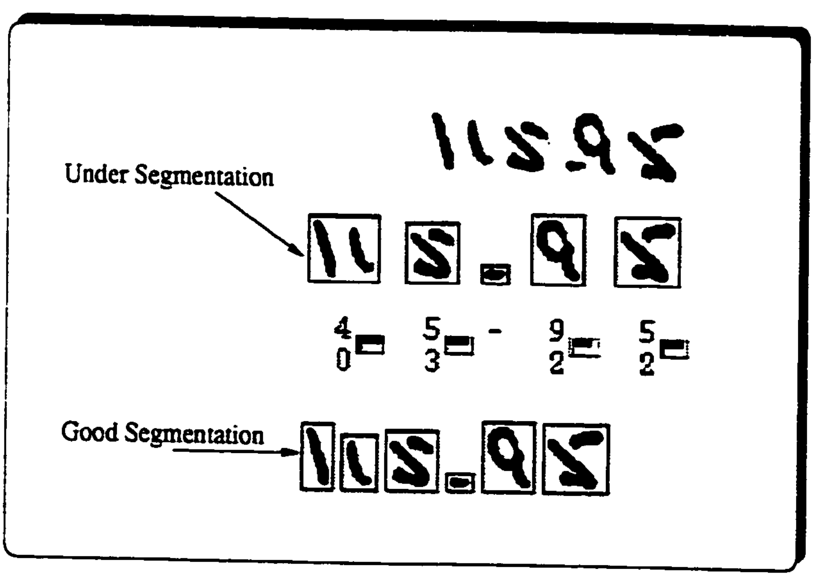


Figure 24: Illustration of under-segmentations.

6.4 System Integration

When a courtesy amount is read from the database, noise is first cleaned and the punctuation marks are detected. The clean numeral string image is subsequently input to the segmentation module, where segmentation hypotheses are generated. These hypotheses are in turn sent to the recognition module. For each hypothesis, the recognition module gives a confidence vaule between 0.0 and 1.0. In this research work, only the top candidate with the highest confidence value is output as a digit string. Finally, this digit string is parsed to a courtesy amount by the syntactic analysis module and verified with the truth amount.

Within this process, segmentation and recognition are the key modules, which make a major contribution to the final results. For this reason, this part of statistics focuses on numerals recognition. It will assist detailed analysis of the performance of the recognition module, while the performance analysis of the segmentation module is discussed in Section 6.3. At the end, final recognition results of courtesy amount are given, which present the overall performance of the entired system.

6.4.1 Numeral recognition

The recognition module consists of isolated digit classifier and touching double zeros classifier. The first one was trained on a database of 450,000 isolated digits and the last one is trained on a database of 55,000 samples. Table 6 shows the recognition results of isolated numerals which are separated from the clean numerals strings among the CENPARMI check database and the real check database. This can illustrate not only the performance of the recognition module on the "real application", but also its contribution to the final recognition results.

Among 645 CENPARMI check images, 2721 sub-images are split in the segmentation stage and sent into the recognition module. Meanwhile, 1604 sub-images are separated among 400 real bank checks. In Table 6, the results are given at two substitution levels, namely 0% and 1%, where the reliability rate is defined by

$$\mathit{Reliability} = \frac{\mathit{Recognition\ rate}}{\mathit{Recognition\ rate} + \mathit{Substitution\ rate}}$$

.

	Recognition (%)	Rejection (%)	Substitution (%)	Reliability (%)
CENPARMI Checks	93.0	7.0	0.0	100.0
2721 sub-images	94.7	4.3	1.0	99.0
Real Checks	90.4	9.6	0.0	100.0
1604 sub-images	92.1	6.9	1.0	98.9

Table 6: Performance of numeral recognition on CENPARMI checks and real checks

6.4.2 Recognition results on real checks

Table 7 illustrates the recognition results of the proposed courtesy amount recognition system on 400 real checks at 0%, 0.5% and 1% substitution levels. Also, Table 7 gives the recognition results of a previous CENPARMI Check Recognition System tested on the same test set. The comparison shows that this newly proposed system provides a significant increase in the recognition rate and the reliability rate. At the 0% substitution level, the proposed system increases the recognition rate by 20.7%.

	Recognition (%)	Rejection (%)	Substitution (%)	Reliability (%)
Proposed CAR system	66.5	33.5	0	100
	68.5	31.0	0.5	99.2
	69.8	29.2	1.0	98.6
Previous CENPARMI	45.8	54.2	0	100
Check Rec. System	56.0	43.5	0.5	99.1
	62.0	37.0	1.0	98.4

Table 7: Recognition results on 400 real checks.

Figure 25 shows some good recognition results. These courtesy amounts are correctly segmented into isolated digits or touching "00"s. Then, these digits including touching "00"s are recognized and a correct amount is given.

Table 8 shows the comparative performances of the proposed system and two state-of-the-art CAR systems, A2iA Check Reader and ParaScript Check Reading System. Among them, A2iA Check Reader had already been operating in several French banks and financial institutions successfully [27]. ParaScript recognizes legal amounts and courtesy amounts on American personal checks [5]. The recognition results are given at 0.5% and 1% error rate levels. It must be noted that there is no standard database in this research area.

	Recognition (%)	Substitution (%)	Reliability (%)
Proposed CAR System	68.5	0.5	99.2
400 samples	69.8	1.0	98.6
A2iA Check Reader	40	0.5	98.8
35,000 samples	49	1.0	98
ParaScript	34	0.5	97.9
5,000 samples	47	1.0	96.6

Table 8: Courtesy Amount Recognition.

6.4.3 Performance Analysis

Table 7 shows that a significant performance has been obtained by the proposed courtesy amount recognition system, compared to the other CAR systems reported in the literature [5, 9, 27]. 66.5% of the 400 real checks are identified at 0% substitution rate level. This performance has almost the industrial requirements: a 70% reading rate and high reliability, so the system has a promising commercial potential. It must be noted that the CAR system is trained and tested on a relative small database. For further research, larger training sets and test sets are needed.

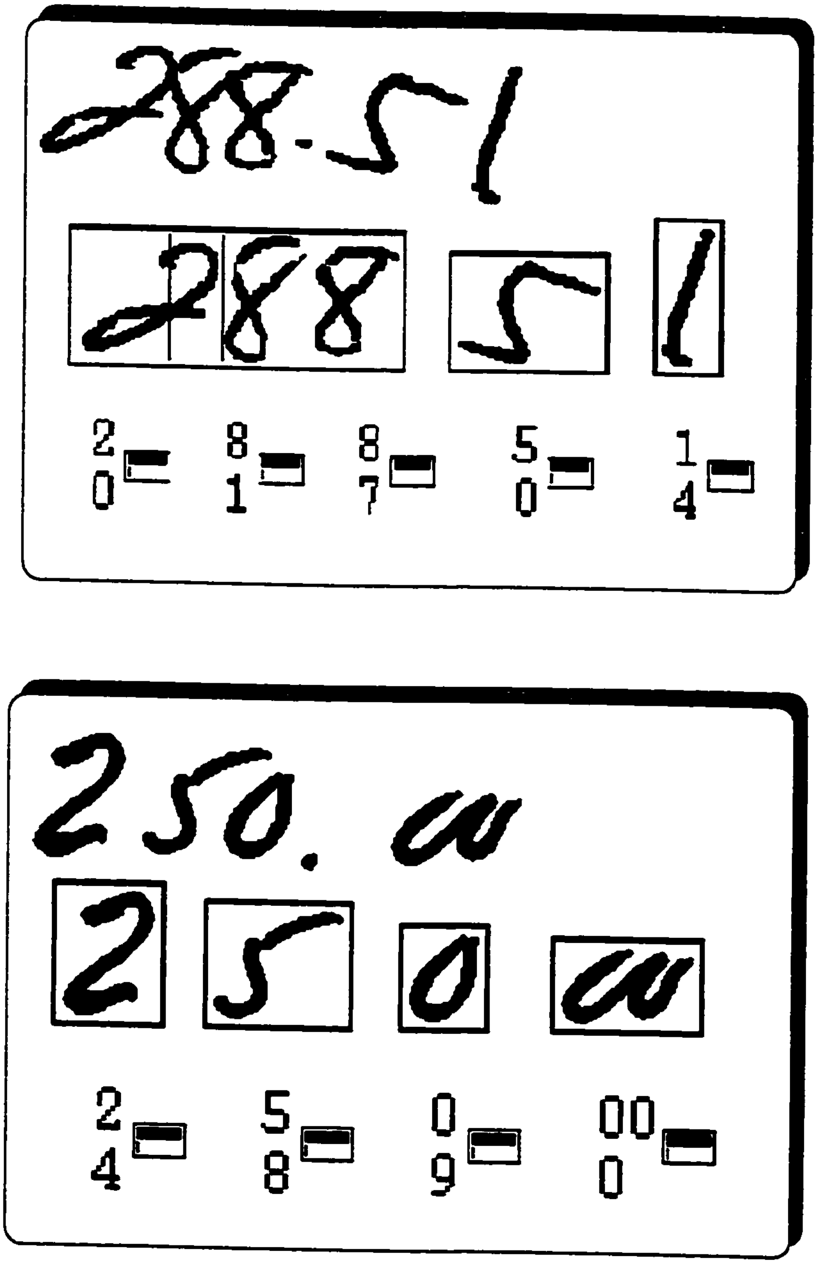


Figure 25: Illustration of good recognition results.

Notably, a significant amount of good segmentation hypotheses have been rejected. A rejection occurs if the sample yields one or more of the following reasons:

1. *Case 1* : Recognition of numerals. Great variations in handwritten numerals provide difficulty in recognition. As Figure 26 shows, the courtesy amount is rejected because the confidence value of '7' given by the recognition module is less than a fixed threshold.

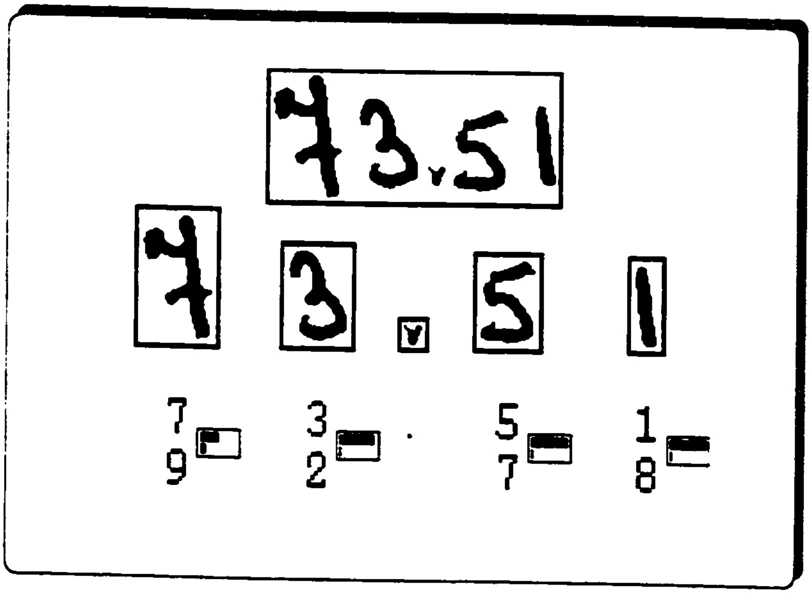
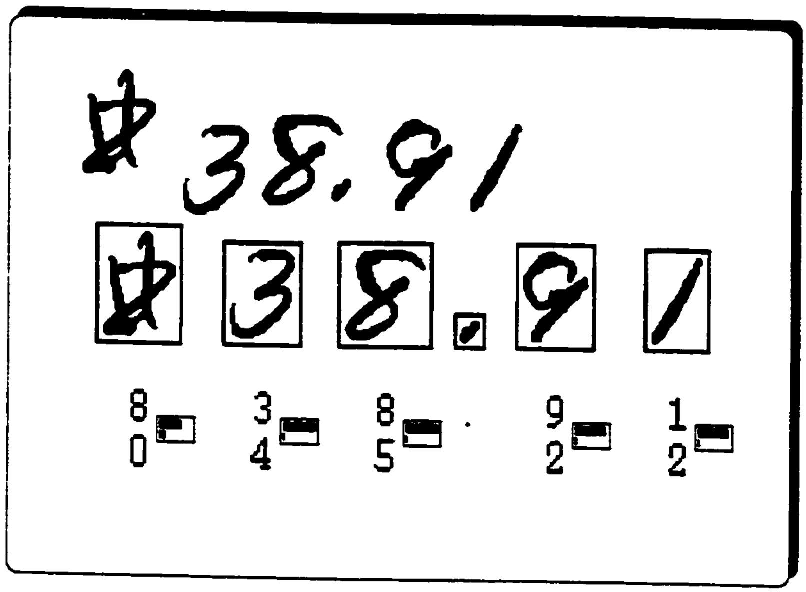
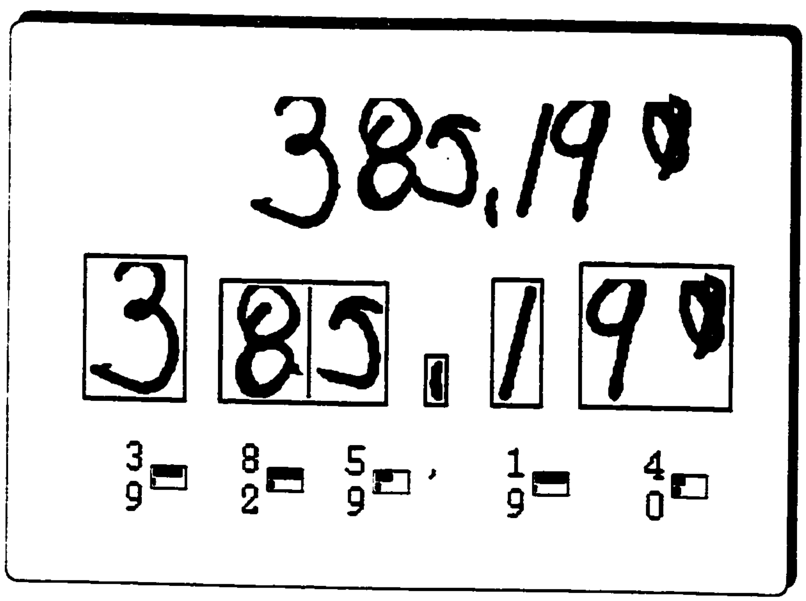


Figure 26: Problem of numeral recognition.

2. *Case 2*: Recognition of non-numerals. As discussed in Chapter 3, three kinds of punctuation marks (periods, comma and dashes) are frequently seen in courtesy amounts. Actually, more non-numerals can be observed besides those punctuation marks, especially the '$' symbol. It is observed that there is a '$' symbol in about 2% of the courtesy amount samples. The '$' symbol is either in the leftmost or in the rightmost of the amount (see in Figure 27). Normally, it is recognized as an isolated digit or combined with related connected components, depending on the confidence values of segmentation hypotheses given by the recognition module. As the confidence values are usually not high enough, this kind of courtesy amount samples are rejected.



(a) '$' is in the leftmost and recognized as an isolated digit



(b) '$' is in the rightmost and recongized as a part of a digit

Figure 27: Problem of '$' recognition.

3. *Case 3* : Ambiguous digit string. Although an amount image is well separated and recognized, it is still rejected by the syntactical analysis module if an ambiguous string is obtained. As Figure 28 shows, since there are only two digits in the amount and no decimal point is detected, the amount will be ambiguous: it can be 16.00 or 0.16. The contextual analysis module rejects this kind of confusing digit strings in an effort to obtain a reliable result.

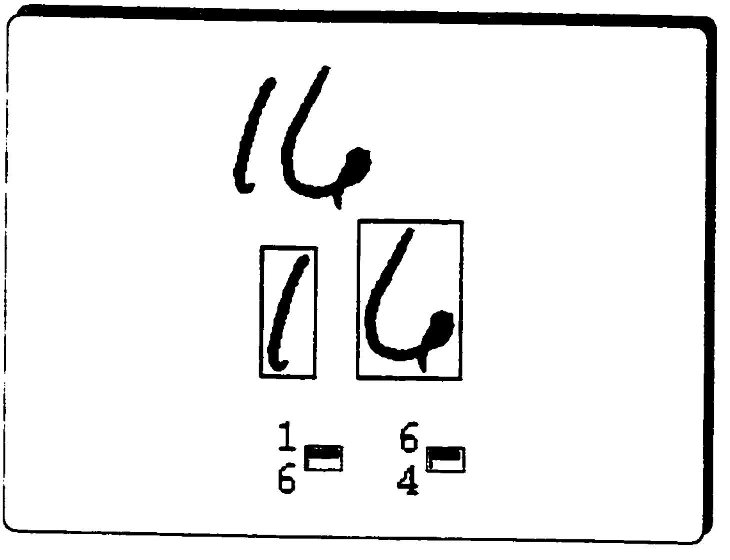


Figure 28: An example of ambiguous digit strings.

A decision of rejection is usually given by two modules in the CAR system, the recognition module and the contextual analysis module. *Case 1* and *Case 2* are generally classified as recognition problems, while *Case 3* belongs to syntactical problems.

It is also worth noting that the proposed CAR system presently processes a courtesy amount in around 2 seconds. The speed is slower than the reported commercial system [3] and also behind the industrial expectation. The bottleneck of this system is the segmentation module. A few of segmentation hypotheses are generated and then verified at the global and local segmentation levels, thus this iterative process is time consuming. For further improvement, speedy algorithms should be reviewed without loss of the system performance.

Chapter 7

Conclusion

7.1 Contributions

A segmentation based courtesy amount recognition system is proposed in this thesis. The system is implemented with into four major modules, referred as preprocessing module, segmentation module, recognition module and parsing module. Every module conducts one hierarchical task in the procedure of courtesy amount recognition.

Three types of punctuation marks (period, comma and dashes) are studied in this research work. These punctuation marks are most frequently observed on Canada bank checks. In detecting them, a set of features are introduced by means of the shape and spatial aspects of the connected component. These features are proven successful in identifying broken pieces from punctuation marks.

A two-level segmentation module is developed using the segmentation-then-recognition strategy. At the first level, namely global segmentation level, the module aims at grouping potential broken digits. The generated hypotheses are verified by the recognition module and the rejected hypotheses are subsquently sent to the local segmentation level. On the contrary, the segmentator seeks to decompose the potential touching digits at the local segmentation level. The split subblocks are also verified by the recognition module.

Two context-free classifiers are integrated into the recognition module for the different purposes. The isolated digit classifier divides input samples into 10 numeral classes ('0'-'9'). The rejected samples are then sent to the double zeros classifier. The holistic double zeros classifier intends to recognize the cursive touching or sloppy double zeros which are difficult to be split into two complete zeros.

Experimental results show the proposed CAR system achieves a 66.5% recognition rate on the test set at 0% substitution rate. This significant performance makes the system one of best courtesy amount recognition systems ever reported.

7.2 Strengths and Weaknesses

The proposed CAR system is segmentation based and aims at cursive handwriting courtesy amount recognition. Several strategies have been combined in the segmentation system, therefore a very good segmentation performance has been achieved. This system has the following strengths:

- Two level strategy is applied. At each segmentation level, the segmention module has different tasks and uses corresponding strategies. At the global segmentation level, the intent is to "undersegment", i.e., to group the neighboring connected components so that the good segmentation boundaries of one digit are included in one sub-block. On the contrary, at the local segmentation level, it seeks to "oversegment", i.e., to split the sub-blocks into several pieces so that the potential touching digits can be separated.

- Multi-hypotheses strategy is applied. Multiple separating points are generated and different combinations of these points are tried.

- It is a recognition based segmentation and the feedback from the recognition module affects the segmentation process. At the global segmentation level, sub-blockes are recognized by recognition system and those rejected sub-blocks belong to the next level. Similarly, an acceptable segmentation at the local segmentation level is the one in which all split parts are well recognized, i.e., their confidence values are larger than a fixed threshold.

Mixed strategies have provided a significant segmentation performance. However, this system still has its weaknesses which are summaried below:

- Relatively slow. As discussed in Chapter 6, the system reads a courtesy amount in about two seconds due to the complexity of the algorithm. The time is consumed mainly by two sources: the generation and then recognition of multi-hypotheses and the feedback interation between segmentation and recognition. In general, the more hypotheses are generated, the more processing time it takes. Also, every hypothesis is verified and the segmentation cuts are adjusted according to the feedback from the recognition module . This iterations contributes quite a lot of processing time.

- High degree coupling between the segmentation and recognition modules. Since it is a recognition based segmentation, segmentation hypotheses are passed to the recognition module and feedbacks are sent reversely to the segmentation module. These frequent interations are against module intergrity. An individual classification sub-module should be integrated into the segmentation module.

- Single source information. The present system highly depends on one context-free recognition module. This design is against the principles of CAR system proposed by Knerr [6], *Use several sources of information.* If the recognition module makes a mistake, it is hardly recovered.

7.3 Future Work

Courtesy amount recognition has been proven challenging due to its complexity. The difficulty of the task is that a high reliability is required as well as a high reading rate. To achieve this goal, generic approaches should be studied and exposed. First of all, the performance of segmentation should be improved, i.e., the good segmentation is bound to be generated among the hypotheses list. On the other hand, more reliable evaluation methods should be investigated, i.e., the good segmentation is sure to have the highest confidence value so that it is in the top of the hypotheses list. Some detailed future research directions are:

- Improvement of recognition methods. The recognition module should not only be able to classify numerals, but also able to reject garbage.
- Simplification of algorithms and procedures. A commerical system is required to process 10,000 checks per hour.
- Recognition of '$'. A suggested solution is to design an efficent and simple nerual network which takes this kind of '$' symbols into account.

Bibliography

[1] S. Casey. Checks still America's favorite payment method. *Item Processing Report*, December 1994.

[2] G.F. Houle, D.B. Aragon, R.W. Smith, M. Shridhar, and D. Kimura. A multi-layered corroboration-based check reader. In *Proc. of the Int. Asso. for Pattern Recognition Workshops on Document Analysis Systems*, pages 495–546, Malvern, Pennsylvania, October 1996.

[3] S. Kelland and S. Wesolkowski. Comparison of research and production architectures for check reading systems. In *Proc. of the 5th Int. Conf. on Document Analysis and Recognition*, pages 99–102, Bangalore, India, September 1999.

[4] N. Bartneck. The role of handwriting recognition in future reading system. In *Proc. of the 5th Int. Work. on Frontiers in Handwriting Recognition*, pages 147–163, Essex, England, September 1996.

[5] G. Dzuba, A. Filatou, D. Gershuny, I. Kil, and V. Nikitin. Check amount recognition based on the cross validation of courtesy and legal amount fields. *Automatic Bankcheck Processing*, pages 177–194, World Scientific, Singapore, 1997.

[6] S. Knerr, V. Anisimov, O. Baret, N. Gorski, D. Price, and J.C. Simon. The Ai2A intercheque system: courtesy amount and legal amount recognition for French checks. *Automatic Bankcheck Processing*, pages 43–86, World Scientific, Singapore, 1997.

[7] L. Huette, 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. *Automatic Bankcheck Processing*, pages 133–156, World Scientific, Singapore, 1997.

[8] C.Y. Suen, L. Lam, D. Guillevic, N.W. Strathy, M. Cheriet, J.N. Said, and R. Fan. Bank check processing system. *Int. Journal of Imaging Systems and Technology*, 7:392–403, 1996.

[9] N.W. Strathy. Handwriting recognition for cheque processing. In *Proc. of the 2nd Int. Conf. on Multimodal Interface*, pages III:47–50, Hong Kong, 1999.

[10] N.W. Strathy. A method for segmentation of touching handwritten numerals. Master's thesis, Dept. of Computer Science, Concordia University, Montreal, Quebec, 1993.

[11] H. Freeman. Boundary encoding and processing. *Picture Processing and Psychoptictorics*, pages 241–266, Academic, New York, 1970.

[12] G. Gallus and P.W. Neurath. Improved computer chromosome analysis incorporating preprocessing and boundary analysis. *Phys. Med. Biol.*, 15(3):435–445, 1970.

[13] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice-Hall, London, 1982.

[14] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27(1):41–52, 1994.

[15] R. Fan. Recognition of dates handwritten on cheques. Master's thesis, Dept. of Computer Science, Concordia University, Montreal, Quebec, 1998.

[16] R. G. Casey and E. Lecolinet. Strategies in character segmentation: a survey. In *Proc. of the 3rd Int. Conf. on Document Analysis and Recognition*, volume II, pages 1028–1033, Montreal, Canada, August 1995.

[17] R. Fenrich. Segmenting of automatically located handwritten numeric strings. *From Pixels to Features III - Frontiers in Handwriting Recognition*, pages 47–60, North-Holland, Amsterdam, 1992.

[18] G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo. Automatic bankcheck processing: a new engineered system. *Automatic Bankcheck Processing*, pages 5–41, World Scientific, Singapore, 1997.

[19] C.J. Burges, J.I. Be, and C.R. Nohl. Recognition of handwritten cursive postal words using neural networks. In *Proc. of USPS 5th Advance Technology Conf.*, pages A–117, Washington, DC, USA, Nov./Dec. 1992.

[20] M. Chen and A. Kundu. An alternative to variable duration HMM in handwritten word recognition. In *Proc. of the 3rd Int. Workshop on Frontiers in Handwriting Recognition*, pages 82–91, Buffalo, New York, May 1993.

[21] M. Gilloux, J.M. Bertille, and M. Leroux. Recognition of handwritten words in a limited dynamic vocabulary. In *Proc. of the 3rd Int. Workshop on Frontiers in Handwriting Recognition*, pages 417–422, Buffalo, New York, May 1993.

[22] X. Wang, V. Govindaraju, and S. Srihari. Holistic recognition of handwritten character pairs. *Pattern Recognition*, pages 1967–1973, 2000.

[23] B. Plessis, A. Sicsu, L. Heutte, E. Menu, E. Leconlinet, O. Debon, and J.V. Moreau. A multi-classifier combination strategy for the recognition of handwritten cursive words. In *Proc. of the 2nd Int. Conf. on Document Analysis and Recognition*, pages 642–645, Tsukuba, Japan, Oct. 1993.

[24] T. M. Ha and H. Bunke. Off-line handwritten numeral recognition by perturbation method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):535–539, 1997.

[25] H. Su, B. Zhao, F. Ma, S. Wang, and S. Xia. A fault-tolerant Chinese check recognition system. *Automatic Bankcheck Processing*, pages 109–131, World Scientific, Singapore, 1997.

[26] D. Guillevic. *Unconstrained Handwritting Recognition Applied to the Processing of Bank Cheques*. PhD thesis, Dept. of Computer Science, Concordia University, Montreal, Quebec, 1995.

[27] N. Gorski, V. Anisimov, E. Augustin, O. Baret, D. Proce, and J. C. Simon. A2iA Check Reader: a family of bank check recognition systems. In *Proc. of the 2nd Int. Conf. on Document Analysis and Recognition*, pages 523–526, Bangalore, India, Sep. 1999.