SEGMENTATION OF LEGAL AMOUNTS ON BANK CHEQUES

JUN ZHOU

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

NOVEMBER 2001

0-612-68494-6

Abstract

Segmentation of Legal Amounts on Bank Cheques

Jun Zhou

Automatic processing of bank cheques has been investigated for ten years at CENter for PAttern Recognition and Machine Intelligence (CENPARMI). Topics studied include the recognition of legal amount, courtesy amount and date. In the CENPARMI Cheque Processing System, sentence-to-word segmentation has to be done before the recognition of legal amount. Segmentation serves as the basis of recognition. However, the previous segmentation module is hindered by both under-segmentation and over-segmentation. To overcome this problem, a feedback-based approach is proposed. It is implemented in two steps. In the first step, segmentation is done according to the structural features found between the connected components in the legal amount. In the second step, a feedback process is introduced to re-segment the parts that could not be identified in the first step. Subsequently, the classifier is used to verify the re-segmentation result. The confidence value produced by the classifier is used to determine the best segmentation points. Two neural network classifiers have been introduced into the segmentation process. Their performances are compared. This approach is tested on a CENPARMI database with 680 images and the result indicates that the new approach has increased the correct segmentation rate by 15.0% from the previous approach.

After segmentation, the legal amounts are recognized by a neural network classifier and a Hidden Markov Model (HMM) and a Multi-Layer Perceptron (MLP) hybrid classifier. With the new segmentation method, the performances of the two classifiers have increased by 9.7% and 8.9% respectively.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Dr. Ching Y. Suen for providing me the wonderful research and study environment. I can't finish this work and my thesis without his guidance, encouragement and patience. From him, I learn not only how to be a researcher, but also how to be a person.

I am also grateful to my co-supervisor Dr. Ke Liu for his guidance, encouragement and help throughout my research in CENPARMI. The time and effort he dedicated to this work is valuable.

Special thanks to Christine Nadal for providing initial research idea and technical support. Thanks to Guiling Guo who helped me to tag the samples in the legal amount database, and to Nick Strathy and Dr. Didier Guillevic whose research and codes have directly contributed to this work.

My thanks go to Qizhi Xu, Jianxiong Dong, Dr. Xiaoping Chen, Liqiang Zhang, Dr. Xuejing Wu and Dr. Yuntao Qian for their fruitful discussion with me and their friendship.

I would like to thank all my friends at CENPARMI for their help and friendship in the past two years: Zheng Li, Dr. Xiangyun Ye, Beverley Abramovitz, Boulos Waked and Yousef Al-Ohali.

I can not express my gratitude with words for the love and support from my parents, my wife Zhichun Fu and my sister. Their care and encouragement are the power to push me forward and will accompany me forever.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The recognition of handwritten text has gained significant attention as an important research and commercial topic since the 1960's. Handwriting research can be divided into on-line and off-line approaches. On-line handwriting recognition means that the machine recognizes the writing while the user writes [34]. On the contrary, off-line handwriting recognition is performed after the writing is completed. In on-line cases, temporal or dynamic information of the writing, such as the number of strokes, the order of the strokes and the direction of the writing for each stroke, and the speed of the writing within each stroke, can be captured by a digitizer or a stylus. The position, velocity and acceleration of the pentip can be calculated as a function of time [25]. However, such information is not available in the off-line case, making the off-line handwriting recognition a more difficult task.

Owing to the dedication of researchers, many off-line systems have been developed for processing bank cheques, postal addresses and commercial forms. In CENPARMI, automatic processing of bank cheques has been studied for ten years. Topics include the recognition of legal amount (i.e. cursive English or French amounts) [5, 30], the recognition of courtesy amount (i.e. handwritten numeral amounts) [37, 16], the recognition of date [32], isolation and identification of data area [18, 36], and so on. Among them, automatic processing of the legal amount is a challenging task because of the high variability of characters, words and writing styles. Before the recognition of legal amount, reliable sentence-to-word segmentation is a difficult task. This thesis proposes a feedback-based approach to improve the segmentation process.

1.1 The Challenge and The Motivation

Even though many studies have been reported on the segmentation of a word into characters [2], the literature on separating handwritten lines into words is rather limited. This is due to the difficulty of detecting and classifying the inter-word and inter-character gaps.

In a machine-printed text, normally there are consistent spaces between characters and between words. The inter-word gaps are bigger than the inter-character gaps. This makes the distinction of two kinds of gaps easy. However, in handwritten text lines, it is not the case. The gaps between words and characters may vary a lot. Sometimes, the distances between words and characters may not differ much. Overlaps often happen between the adjacent words (Figure 1). The lines may undulate up and down due to irregular vertical positions, different sizes, or ascenders and descenders of the words (Figure 2).

In many application situations, such as segmentation of postal addresses or date, some additional information can be used to help the segmentation process. For example, the transition between lower case characters and upper case characters, and the presentation of punctuation marks, are all the cues that could be utilized [3, 9, 32]. Nevertheless, the segmentation of legal amounts in bank cheques is more complicated. The distance between words in a line vary considerably because the users often write big characters at the beginning of a line and later on find that the space is not large enough at the end. As usual, the transition between upper and lower characters may not exist. And often, there is no punctuation in the legal amount at all. In such cases, the selection of methods to measure the distance between gaps and then classify them become a troublesome task. The most popular approach is to find the connected components, compute and sort the distances between them, and then select a threshold to determine the inter-word and inter-character gaps. However, it is difficult to find an ideal method to select the proper threshold. In a specific approach, by training of the segmentation algorithm, a good threshold can be reached while the incorrect

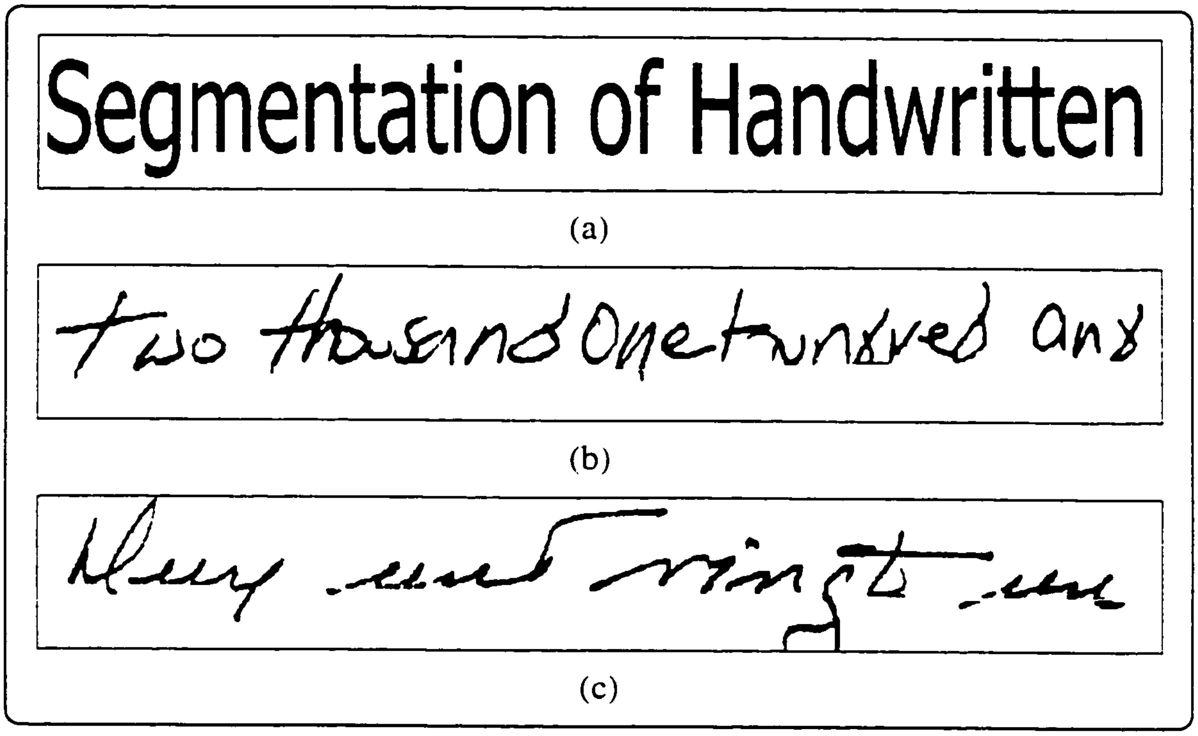


Figure 1: Machine printed text versus handwritten text. (a) Spaces are evenly distributed in machine printed text; (b) Spaces between words and characters vary a lot in handwritten text; (c) Overlaps often happen.

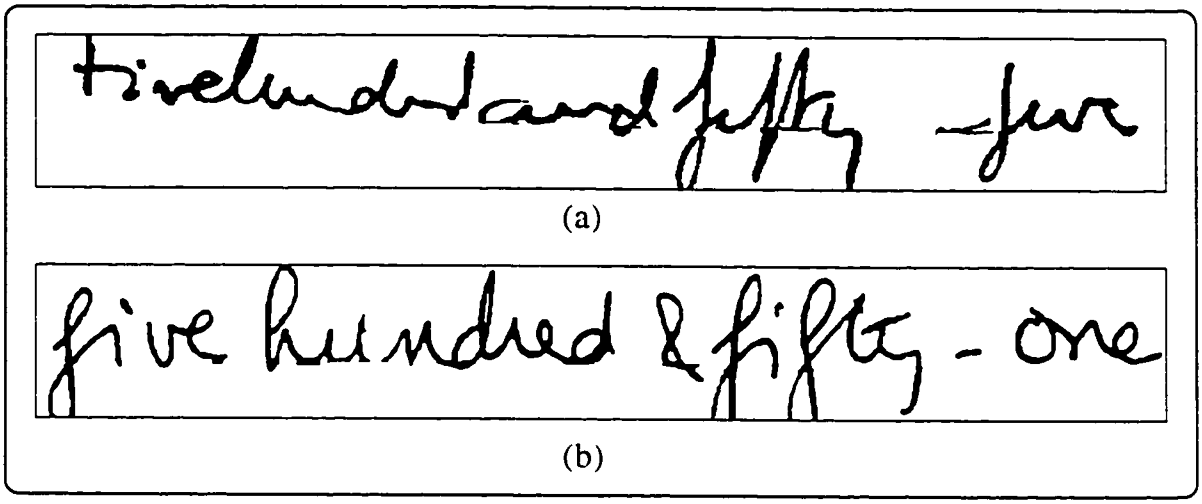


Figure 2: Undulation of handwritten text lines: (a) Vertical positions of the words are irregular; (b) Ascenders and descenders.

segmentation rate may still be very high.

In many systems, sentence-to-word segmentation is bypassed or ignored [23, 26]. However, in the CENPARMI Cheque Processing System, sentence-to-word segmentation serves as the basis of recognition. Segmentation must be done to produce the list of words for the classifier to recognize. The performance of the segmentation module will directly affect the performance of the whole system. Thus, an effective, efficient and robust segmentation approach is quite important.

1.2 Previous Work

A typical sentence-to-word segmentation approach may encompass the following steps:

1. find the connected components in a given text line;

2. compute the distances between adjacent connected components;

3. sort the gaps in ascending or descending order of magnitude;

4. select a method to compute the threshold to determine the inter-word and inter-character gaps.

Most of the segmentation approaches assume that gaps between the words are bigger than those between the characters. This makes the choice of a method to compute and classify the gaps a critical task.

Seni and Cohen [27] presented a comprehensive discussion on separating a line of handwritten text into words by determining the location of inter-word gaps. Eight distance calculation algorithms are presented which combine three basic distance measures as well as some heuristics. The three basic measures are listed below and shown in Figure 3 (a), (b) and (c).

- The *bounding box method* computes the minimum horizontal distance between the bounding boxes of the connected components. The distance between bounding boxes that overlap horizontally is considered to be zero.
- The *Euclidean method* computes the minimum Euclidean distance between the connected components.
- The *minimum run-length method* computes the horizontal run-length between portions of the connected components that overlap vertically. The minimum of the run-length is used as the distance measure.

Researchers in CEDAR proposed another method to estimate the gaps between the connected components [21]. As shown in Figure 3 (d), the *convex hull*, which is the smallest convex polygon enclosing the component, is found. Then, a line is joined between the centers of gravity of the connected components. The distance of the intersections of the line and the hull forms an approximation of the inter-component distance.

The above gap estimation methods have been used by many researchers [28, 35]. Knerr et al. [15] used the horizontal distance between the bounding boxes of the adjacent graphemes (*d1*) and the minimum horizontal run-length between the adjacent graphemes (*d2*). The joint empirical distributions of *d1* and *d2* is computed.

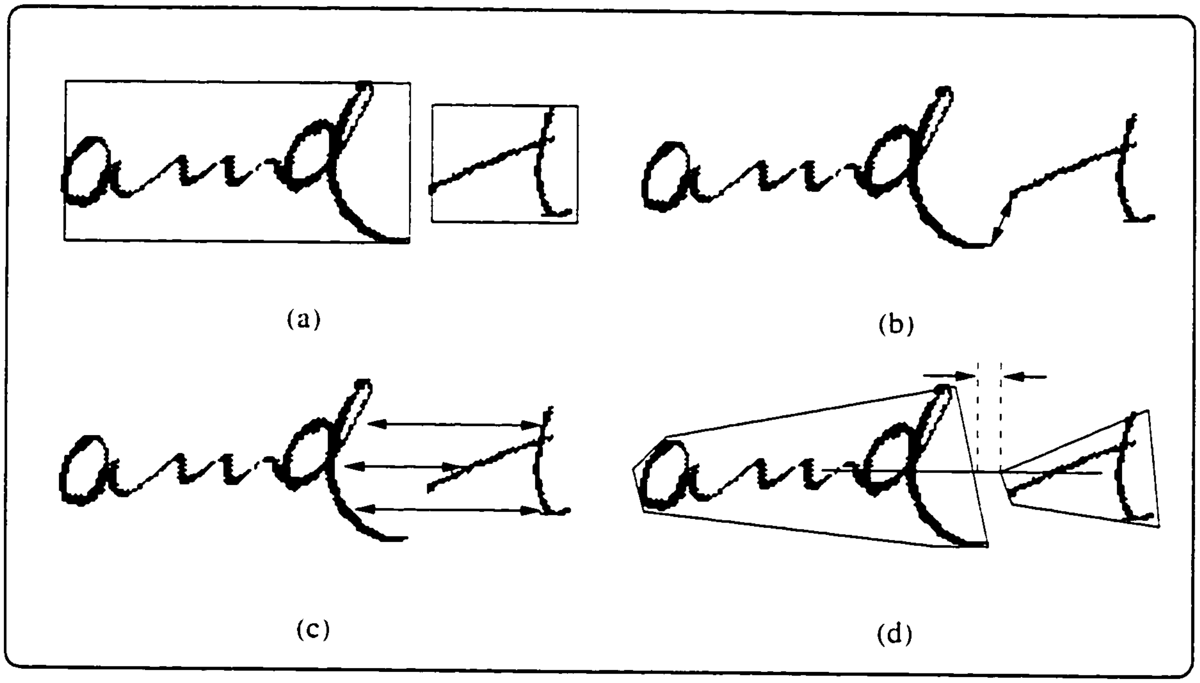


Figure 3: Four types of basic measures for computing the distance between two connected components. The bounding boxes are shown in (a), and the gap is measured by the horizontal distance between two bounding boxes. The Euclidean distance is shown in (b) where the arrows indicate the distance between the closest points in each component. The run-length distance is shown in (c) as indicated by the arrows whereas (d) presents the distance measured by the convex hull.

Then probabilities for all word segmentation options to be correct segmentation are evaluated according to the conditions on the number of words and number of connected graphemes. Finally, the most probable segmentation options form the word segmentation candidate list.

The method of Leroux et al. is also based on the gaps [17]. The lengths of the gaps are compared with the mean length of letter segments. Segmentation-by-recognition strategy, which is often used in the segmentation of word [14], is then applied to remove the ambiguities between inter-word, inter-character and intra-character gaps.

Hussein et al. used a three-phase segmentation approach in their system [7]. The first phase involves a linear density measurement of the legal amount. The density is the horizontal projection of the words in the legal amount. Segmentation points are chosen based on the region with zero linear density in the image. In the second phase, connected components are extracted from the binary image. A post-processing step is used to eliminate discontinuity in a single word. The final phase searches for peaks in the horizontal projection of the binary image, and the minima within the vertical projection. Segmentation is performed at these minima based on the average word length.

Madhvanath et al, Mahadevan et al, and Park et al. proposed some other methods to find the segmentation hypotheses after detecting the gaps [20, 22, 24], which include the nearly exhaustive method, MAX method, hierarchical clustering, and confidence computation by introducing features such as ascenders and ligatures.

All the methods mentioned above are based on identifying the physical gaps between the connected components, while Kim proposed another approach to characterize the variation of spacing between adjacent characters as a function of the corresponding characters themselves [10]. The segmentation points are determined using features like ligatures and concavities. The words are over-segmented, which means that no two characters remain connected, and a given character is expected to be split into at most three to four segments. Finally, a neural network is used to determine the segmentation points as well as to recognize the words.

1.3 Proposed Method

In the CENPARMI Cheque Processing System, a K-Nearest Neighbour (KNN) classifier based on global features [6] and an HMM-MLP hybrid model classifier [30] have been implemented to recognize the legal amounts after preprocessing and segmentation. The performance of both recognizers depends heavily on the result from the sentence-to-word segmentation process. However, the under- and over-segmentation problems are not well solved in the previous approach, which makes the segmentation process a bottleneck of the system.

In this thesis, a feedback-based approach is proposed to improve the performance of the segmentation [38]. In our approach, segmentation is divided into two steps. In the first step, a structural feature based segmentation module is called to calculate the Euclidean distances between the connected components in a sample. An initial segmentation is made while the over-segmentation is controlled to a low level. In the second step, the feedback information on the lengths of the segments is collected from the result of the first step. By comparing this information with the global information of the legal amount sample, re-segmentation is called to split, if needed, the selected segments. Several segmentation points of the selected segments are produced. A multiple neural network classifier is then introduced to generate confidence values that can be used by the feedback system to select the best segmentation points. By doing so, the under-segmentation problem is well solved. The test result proves the approach to be effective and efficient. The correct segmentation rate has been increased to 81.8%, which is a 15% improvement comparing with the previous approach.

Chapter 2

Legal Amount Processing

2.1 The CENPARMI Cheque Processing System

Large quantities of cheques are being used in daily business transactions in the world. Most of the cheques are processed manually. To reduce the labour cost, an automatic cheque processing system may be used to process the handwritten date, courtesy amount, legal amount and signature (Figure 4). The design of a commercial automatic cheque processing system is not easy due to the high reliability requirement of the system. Difficulty arises because the cheques from different sources may have different layouts and backgrounds. Problems are compounded when processing the cursive script. The CENPARMI Cheque Processing System is being designed to process the cheques for large companies and banks [31]. It is an integration of the results of several researchers [4, 12, 19, 29, 32, 36].

In this system, a grey-scale image is obtained by scanning the cheque at a resolution of 300 dots per inch (DPI). The image is smoothed and then its background is removed iteratively by applying a thresholding. After that, the baselines under the items are located and removed. When doing this, some parts of the handwriting that intersect with the baseline would also be lost. Thus, a morphologic closing operation, an image enhancement by median filtering and binarization and a topologic process is applied subsequently to restore the lost information. Finally, the segmentation of required information and recognition are done.

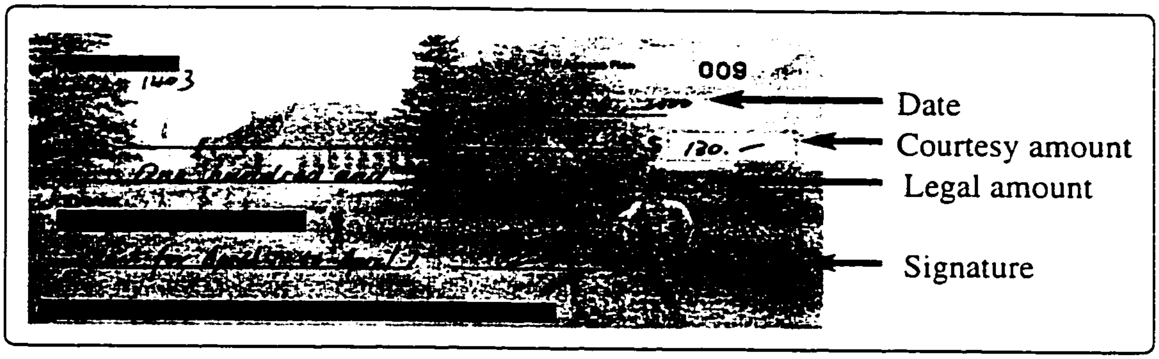


Figure 4: A typical cheque and handwritten regions to be processed

The recognition results of the courtesy amount and legal amount are compared. Since the performance of the courtesy amount classifier is better than that of the legal amount classifier, the result of the former is more accurate and reliable. The result of the legal amount classifier could be used as a verifier for the former. The structure of the whole system is displayed in Figure 5.

2.2 Legal Amount Processing

2.2.1 Overview

Legal amount processing is an important sub-system in the CENPARMI Cheque Processing System. It is also one of the most difficult tasks in the system. Currently, several recognition approaches have been implemented.

A typical recognition engine is proposed by Guillevic and Suen [6] that combines a global feature scheme with an HMM module. The global features consist of the encoding of the relative position of the ascenders, descenders and loops within a word. The HMM uses one feature set based on the orientation of contour points as well as their distances to the baselines. The engine is fully trainable, reducing to a strict minimum the number of hand-set parameters. The system is also modular and independent of specific languages such as English and French. Figure 6 shows the

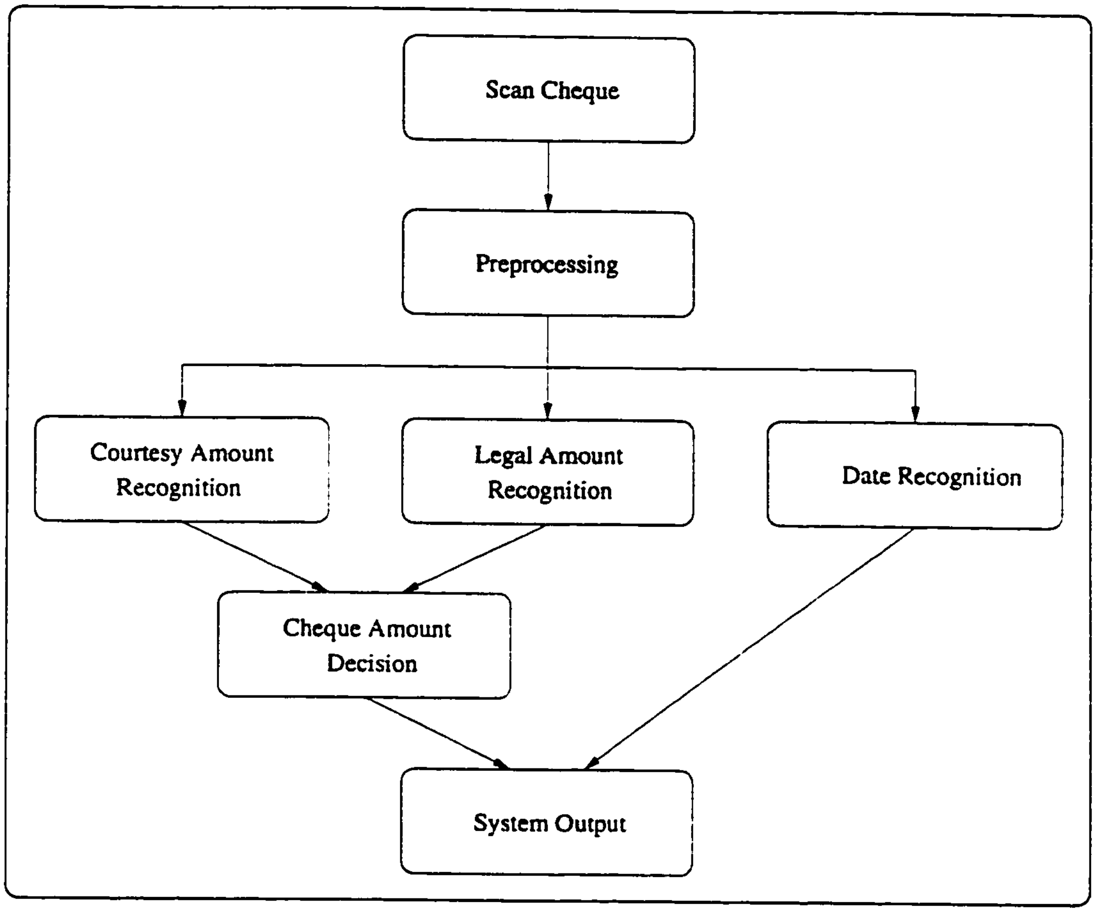


Figure 5: Structure of the CENPARMI Cheque Processing System

structure of the legal amount recognition sub-system.

2.2.2 Description of The Word Segmentation Module

The segmentation module (SEG1) in the CENPARMI Cheque Processing System adopts the structural features to calculate the Euclidean distance between the connected components [6]. It generates segmentation paths and produces word candidates to be sent to the recognizer. In the module, it is assumed that the space between words is bigger than that between characters.

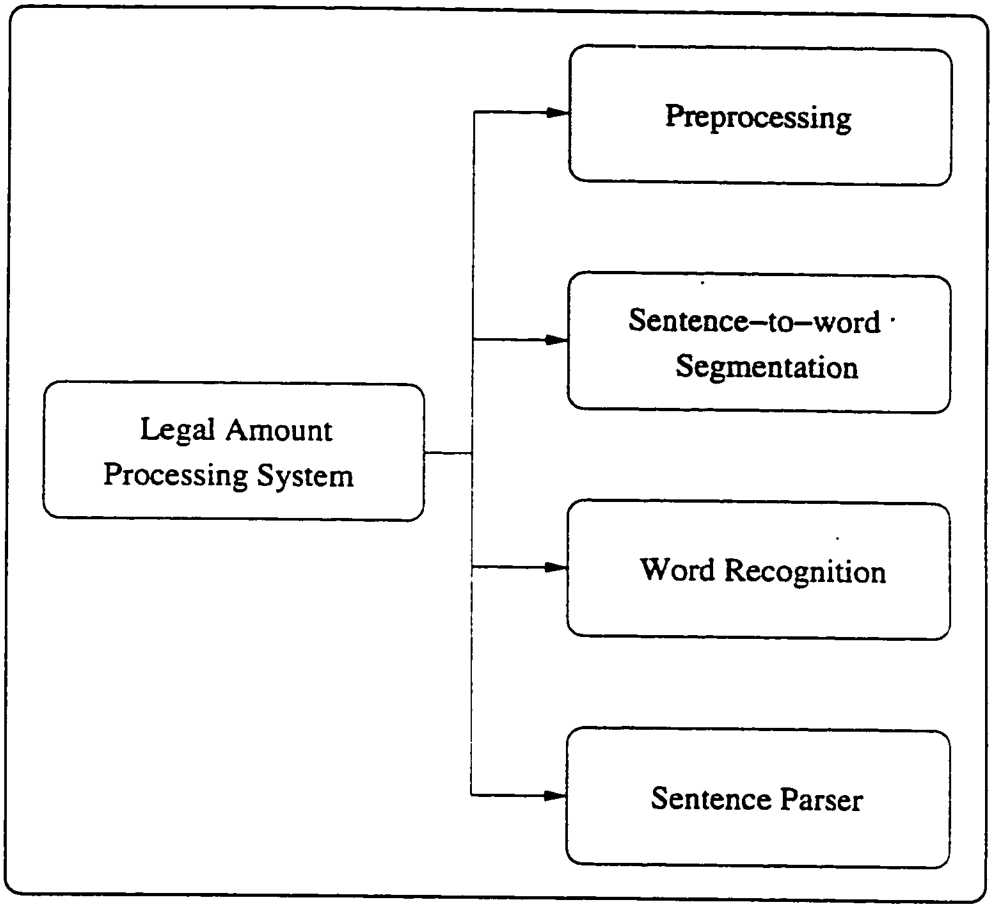


Figure 6: Processing of legal amount

SEG1 first represents a binary legal amount image as a list of contours of the connected components. Then, the components that are recognized as line or dash and those that meet some requirements, such as punctuation, are removed from further consideration. In this step, several classifiers have been developed and trained for the recognition and removal [4]. For example, a Bayesian classifier based on simple features (i.e. aspect ratio and average vertical density) is designed and trained to recognize and remove the lines that people have the tendency to write at the beginning or end of the legal amount. Then the system groups some connected components

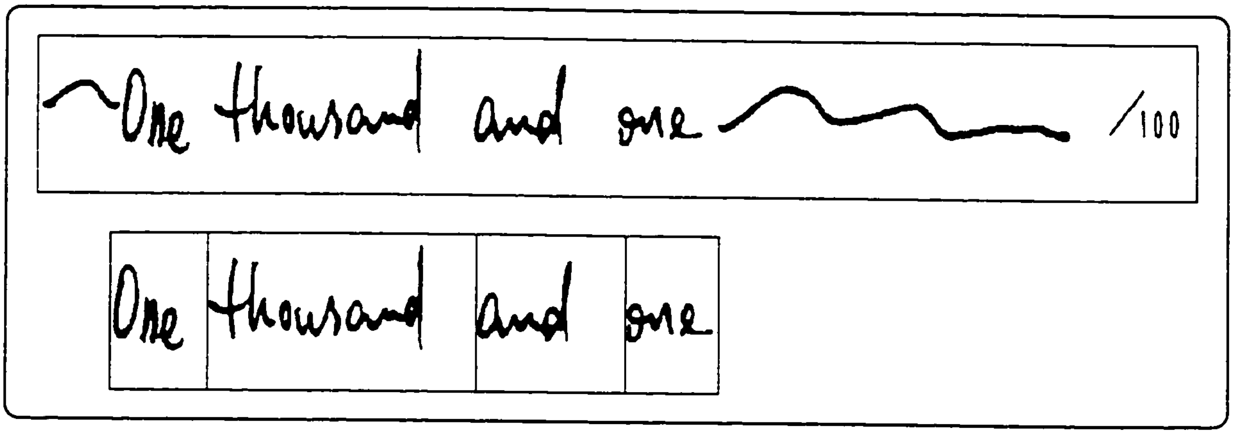


Figure 7: Word segmentation example. Non-word components are removed from the image. The vertical lines indicate the positions of segmentation.

together. For example, fully overlapped components, such as the small dot over "i" or the vertical stroke in the "T". This process is also useful for those words composed of broken strokes due to the weakness of the binarization process. The components that are very "close" to each other are also grouped together. After finishing the grouping step, the system computes the distance from one group to the next. The distances are sorted by their sizes and the largest inter-word gap is found. All those gaps that are larger than a given fraction of the largest gap are classified as inter-word gaps. The word candidates are then sent to the recognition module. Figure 7 is an example of word segmentation.

The word segmentation module has been tested on test database 1, which consists of a set of 300 cheques provided by a local phone company. The results have been analyzed manually for each sample. The overall performance is shown in Table 1.

	Correct	Under	Over	Noise	No
Result	74(%)	4(%)	9(%)	4(%)	9(%)

Table 1: Test result of SEG1 on database 1.

Here. the "Correct" word segmentation indicates that all the words in the amount could be isolated successfully. The samples in the "Under" and "Over" categories are under-segmented or over-segmented, where there exist fewer or more words in the segmentation results than the actual number of words. The samples in the "noise" category come from the errors in binarization step and item extraction module. A typical example is that the background is not completely removed and is connected with some words. This will prevent the segmentation module from finding and calculating the gaps between the connected components. In the "no" category, legal amounts can not be separated by SEG1 because some words are physically connected. or words are connected with long lines. or words are so close to each other that the inter-word gaps is not comparable with the inter-character gaps. The segmentation approach based on structural features can not easily solve the problem caused by a physical connection between words.

2.2.3 Description of Problem

By analyzing the above test result in detail. we find that two kinds of segmentation errors could be avoided or reduced. One kind of errors is due to under-segmentation and over-segmentation. The errors happen because after the gaps between connected components have been found and sorted. a threshold is selected and used to distinguish inter-word gaps and inter-character gaps. If we adjust the threshold. the segmentation result could be changed. Another kind of errors belongs to the "no" category. Some inter-word gaps could not be found because the words are very close to each other. By changing the definition of "close", these errors could be reduced.

Another test is done on SEG1 [39]. This time, we use database 2 which consists of 408 samples. The samples are cheques collected from the IRIS-PRECARN project in May 2000. the Concordia University Open House in January 2000. and the demos done at CENPARMI. Among the 408 cheques. the item extraction module can not extract the legal amounts from 47 cheques and the preprocessing module can not remove noises from 43 cheques. Thus. the final test database contains 318 samples of extracted and cleaned legal amount images. The test result is shown in Table 2.

Figure 8 shows some under-segmented samples. Figure 9 shows some over-segmented samples.

	Correct	Under	Over
samples	216	89	13
percent	67.9%	28.0%	4.1%

Table 2: Test result of SEG1 on database 2

In the segmentation result, we combine under-segmentation and no segmentation into one single category. From the above result, we find that under-segmentation is the main problem of the SEG1 and it is caused by the selection of a threshold to distinguish inter-word gaps from inter-character gaps. However, we could not simply adjust the threshold, because a decrease in under-segmentation may increase the over-segmentation. And the adjustment can not solve the problem if the inter-word gaps and inter-character gaps are very close to each other.

A method that could solve the problem stems from the fact that one could generate multiple segmentation paths based on the distances between connected components. In these paths, the probability of finding the correct word segments could be increased and maximized. Also, it has been proved that recognition-based segmentation can find an optimal path from a list of word candidates.

In the CENPARMI Cheque Processing System, SEG1 acts as the basis of the recognition module. Since the performance of a classifier is highly dependent on the word segmentation result, we should solve the problem of word segmentation before recognition. Meanwhile, the recognition module may include a combination of several classifiers because such configuration can improve the recognition rate. Then if we give several segment paths to the recognizer, the computational complexity of the system will be increased which may reduce the system efficiency. One direct consequence is that the processing time of the system may be multiplied according to the number of the paths used. Because of this, we have to develop an approach that would generate only one segmentation hypothesis.

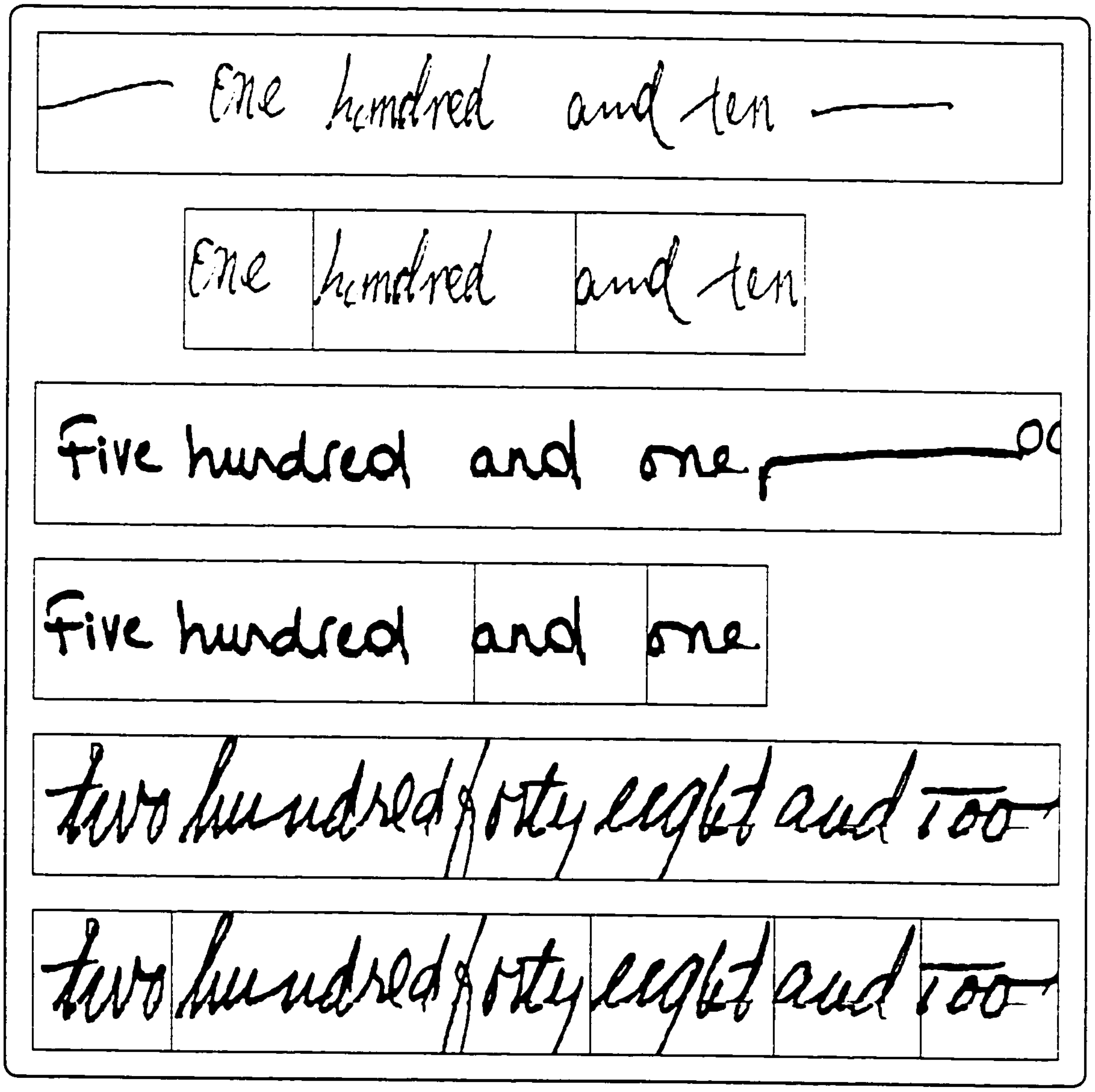


Figure 8: Some under-segmented samples of SEG1.

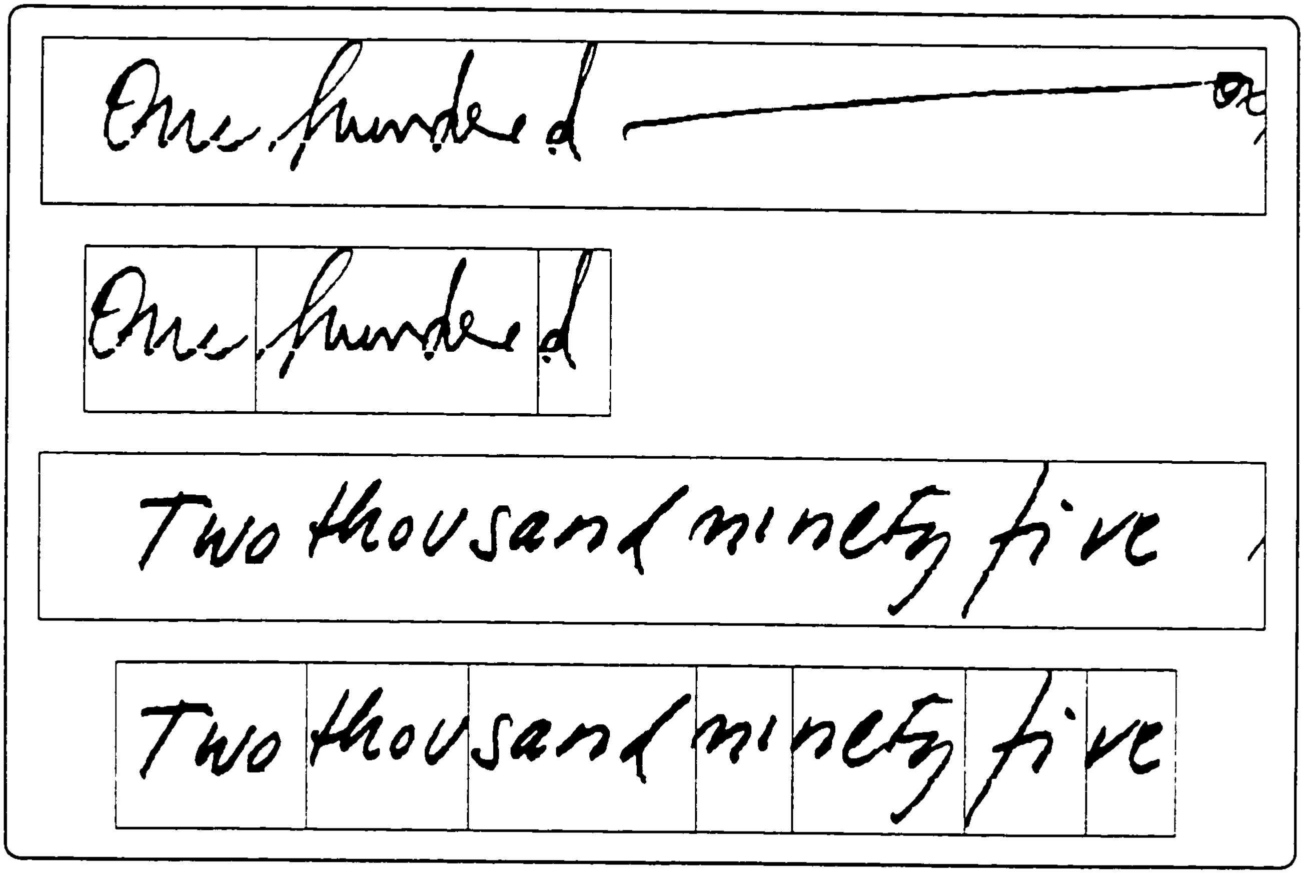


Figure 9: Some over-segmented samples of SEG1.

Chapter 3

Preprocessing and First Step Segmentation

3.1 Overview

The proposed feedback-based segmentation approach is developed to improve the performance of sentence-to-word segmentation module in the CENPARMI Cheque Processing System. The output of the approach is a list of legal word images. Before the development of the approach, we assume that the legal amount has been extracted from the cheque image.

Based on the above assumption, the approach is implemented in several modular steps. The legal amount image is first read into the system. The connected components are detected. Overlapped components are grouped together. The preprocessing, such as slant correction and line removal function, is applied. Then, a structural feature based segmentation module is used to do the first step of segmentation, or called pre-segmentation. The possible under-segmented segments are then found according to the feedback from the pre-segmentation. After that, these segments are split by the second step of segmentation to generate several hypothetical segmentation points. A neural network classifier is then introduced to generate confidence values. A dynamic-length sliding window method is used by the feedback system to select the best hypothesis path from the points. This path is then combined with the other segmentation candidates got from pre-segmentation to produce the final result. The structure of the approach is shown in Figure 10.

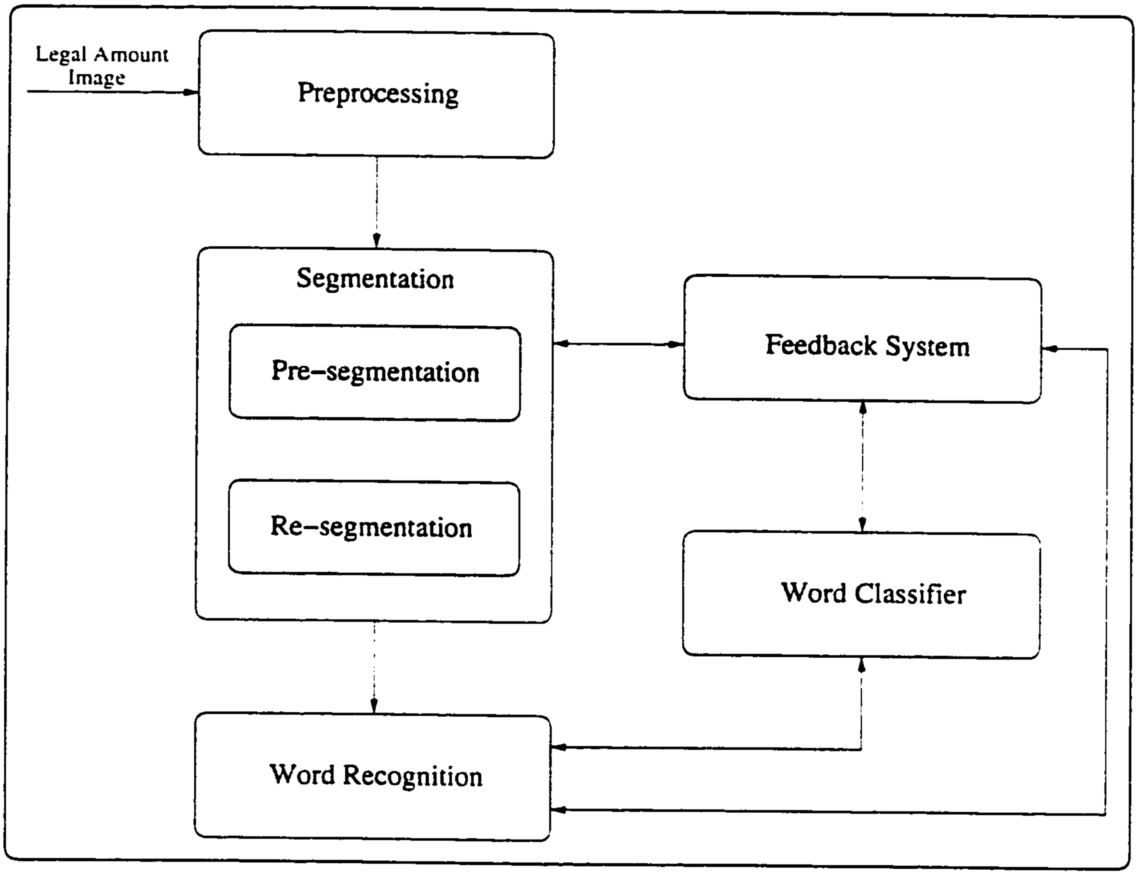


Figure 10: A block diagram for feedback-based segmentation approach

3.2 Preprocessing

3.2.1 Slant Correction

Some people tend to write text with a slant of certain degree to the right or left of the vertical. Slant correction normalizes the slant to the zero degree. This step is important because it will affect the calculation between the distance of connected components as well as the extraction of features in word image. It is the basic requirement of a writer-independent system. For this reason, slant correction is used in almost all the segmentation and recognition approaches [1, 8].

The slant correction algorithm we are using consists of two steps. First the slant of a given word is computed based on image projections at vaious degrees. Then, we correct the slant by a shear transformation [4].

Slant calculation

The histogram is computed at various axes by steps of (*delta_theta*) degrees. In a single scan of the image, all of the slanted histograms are computed. The algorithm used is as follows:

Algorithm: SLANTCALCULATION(*image*)

for each pixel of the image (i, j)
  for each of the slant value $(k * delta\_theta)$
    1. compute the new value v of the original coordinate j
      in the slanted histogram:
      $v = j - (height - i) * \tan(k * delta\_theta)$
      where *height* represent the height of the image.
    2. increment the count in the v^{th} column of the k^{th} histogram:
      $histogram_k[v] = histogram_k[v] + 1$;
  end **for**
end **for**

The algorithm described here assumes that the origin of the image is at the top left corner with the i and j coordinates increasing downwards and to the right respectively. In the algorithm, the *delta_theta* is fixed at 5 degrees. For both left and right directions, the number of histograms computed is set at 15. Thus, k ranges from 0 to 15 in both directions. As a result, a total number of 29 histograms for angles varying from $-70°$ to $+70°$ are computed at steps of $5°$. We then look into all the histograms derived and find the one with the greatest derivative. The $k * delta\_theta$ for this histogram is set as the average slant.

Slant correction

After we have found the average slant, we correct the slant with shear transformation [33]. During the shear operation, for each pixel (i, j) in the original image, the new coordinates of the slant-corrected image are calculated as follows:

Algorithm: SLANTCORRECTION(*image*)

> **for** each pixel of the image (i, j)
> $y = i$
> $x = j - (height - i) * \tan(\theta)$
> end **for**

where θ is the angle obtained from slant calculation. We noticed that after the transformation, the height of the word is not changed while the width of the words will probably change. The points on the bottom row of the image are not modified by the transformation. As a result of this transformation, the greater the value of the slant, the greater the variation of the aspect ratio of the image.

The slant correction algorithm provided here is proved to be effective. Figure 11 shows some image examples before and after the operation.

3.2.2 Line and Digit Removal

Line removal

People tend to write lines at the beginning and the end of the legal amounts. These lines are useless information for the recognition. To remove them, we developed a simple classifier. In our method, the vertical histogram of each connected component is calculated. Then the maximum value in the histogram is obtained. This value, together with the widths of the connected components, the left and right boundaries of the connected components and the length of the legal amount image, is used for the classification. The algorithm is shown below:

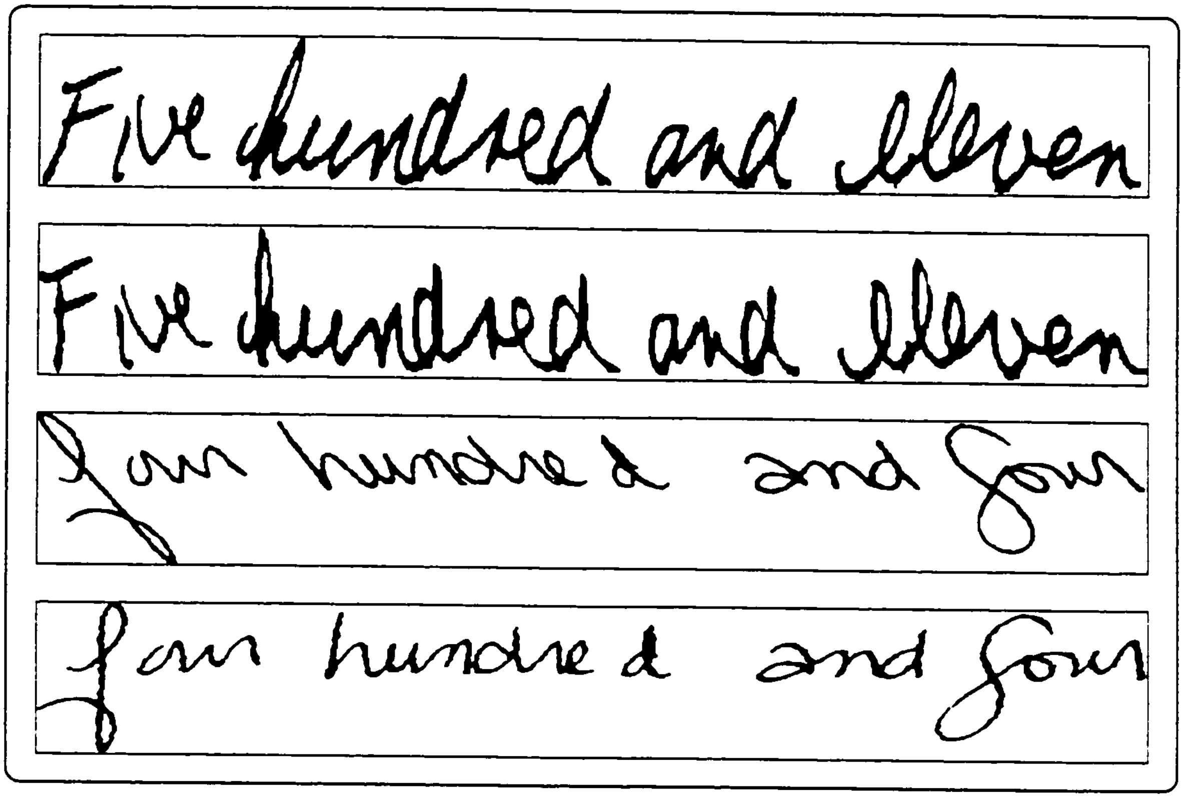


Figure 11: Slant correction

Algorithm: LINECLASSIFICATION($x_1, \cdots, x_n, image$)

$width_{image} \leftarrow$ width of the image
for $i \leftarrow 1$ to n
  compute the horizontal histogram of x_i
  $max_{x_i} \leftarrow$ maximum value in histogram
  $(width_{x_i}, left_{x_i}, right_{x_i}) \leftarrow$ width, left and right boundary of x_i
  if $max_{x_i} < t_1$ and $width_{x_i} > t_2$ and $right_{x_i} < t_3 * width_{image}$ **or**
  if $max_{x_i} < t_4$ and $width_{x_i} > t_5$ and $left_{x_i} > t_6 * width_{image}$
    remove x_i from the legal amount
end **for**

where x_i is the i^{th} connected component, n is the number of the connected components, t_1 to t_6 are empirical values. Figure 12 shows some examples of line removal.

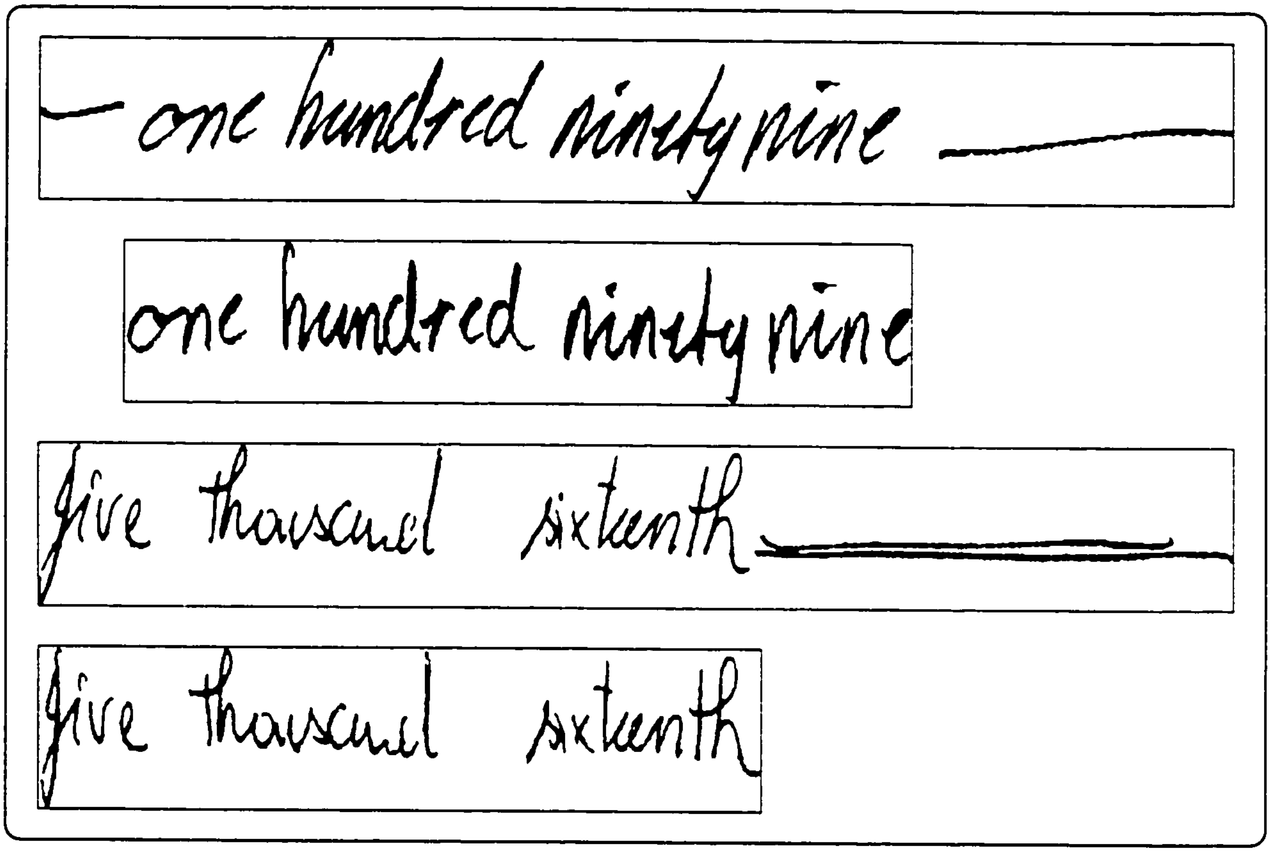


Figure 12: Line removal

Digit removal

When writing cheques, people often write the dollar part in legal words while the cent part in digits. The digital cent part is always written at the end of the legal amount. The recognition of the cent part could be fulfilled by a digit classifier. Thus, to effectively remove the digital part, we would need to develop a classifier to distinguish digit from word, which is beyond the scope of this thesis. Here, we developed a simple classifier based on the position information of the cent part digit in the legal amount. We tried to isolate this part for further digit processing.

By analyzing the legal amount samples, we noticed that the digital part is often written far away from the legal part. A long line may exist to separate them. Thus, we detect the digit part as follows:

1. Find the largest Euclidean distance (gap) between the connected components.

2. If the left boundary of the last connected components is within the last certain portion (t_7) of the legal amount image, and the largest gap is bigger than a certain value (t_8), remove the last connected components from further consideration. t_7 and t_8 are empirical values.

Figure 13 shows some examples of digit removal. Figure 14 shows some examples of both line and digit removal.

3.3 First Step Segmentation

Pre-segmentation, which is the first step of segmentation, is implemented to do the initial cut of legal amounts after preprocessing. Here, we used the previous segmentation method in the CENPARMI Cheque Processing System. The gaps between connected components are calculated and sorted according to the Euclidean distance. A threshold is selected as a portion of the largest gap. Those gaps, which are larger than the threshold, are classified as inter-word gaps. The others are considered as inter-character gaps.

Different from the previous method, the selected threshold is obtained by training the approach on a database (refer to section 6.3 for training method). However, in the previous method, it is an empirical value. This threshold is also interconnected with the one that is used in the second step of segmentation as we noticed that the steps in the whole segmentation approach are interactive and integrated. After the pre-segmentation, the legal amount image is divided into several segments. In each segment, there may be one or more words. It is also possible that some words are split into several segments.

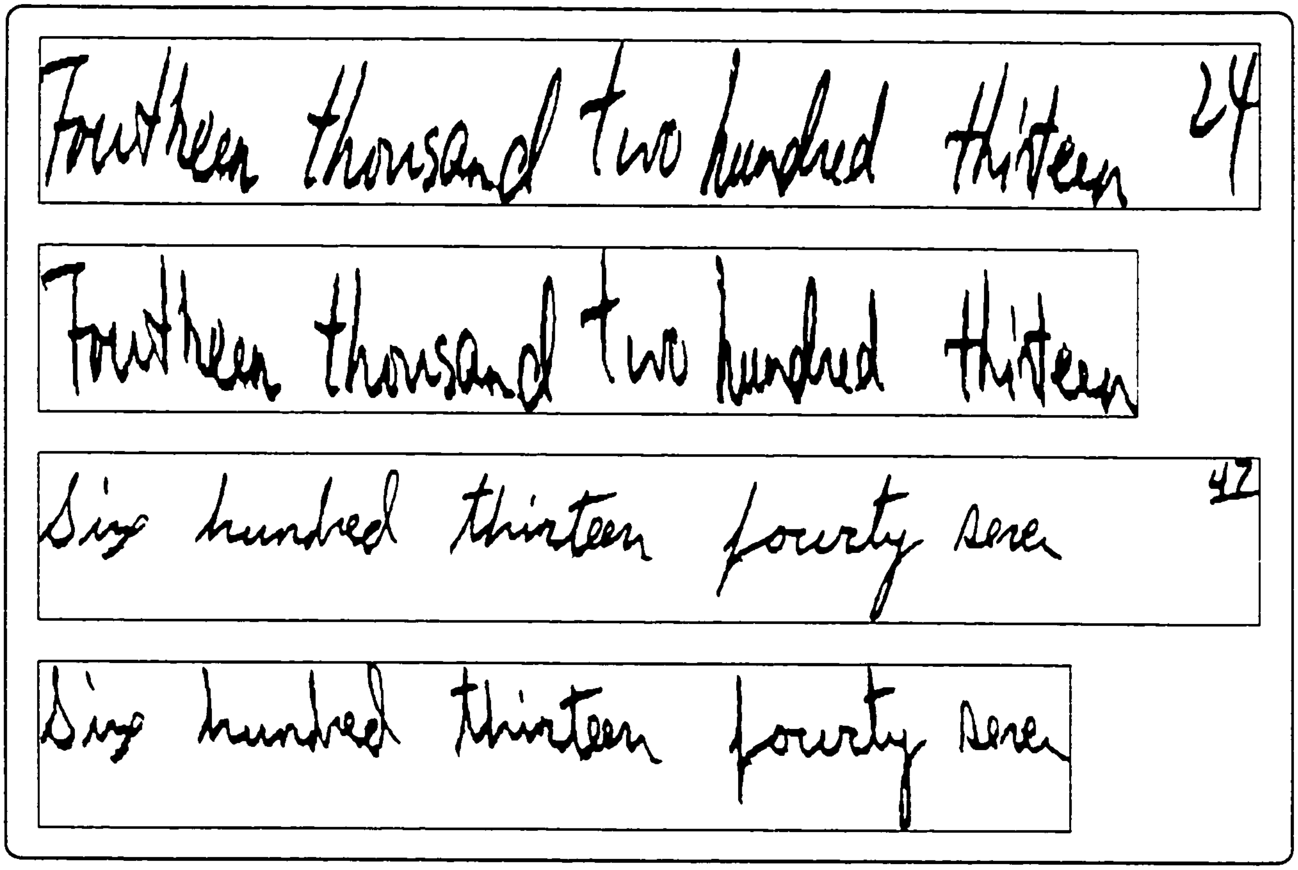


Figure 13: Digit removal

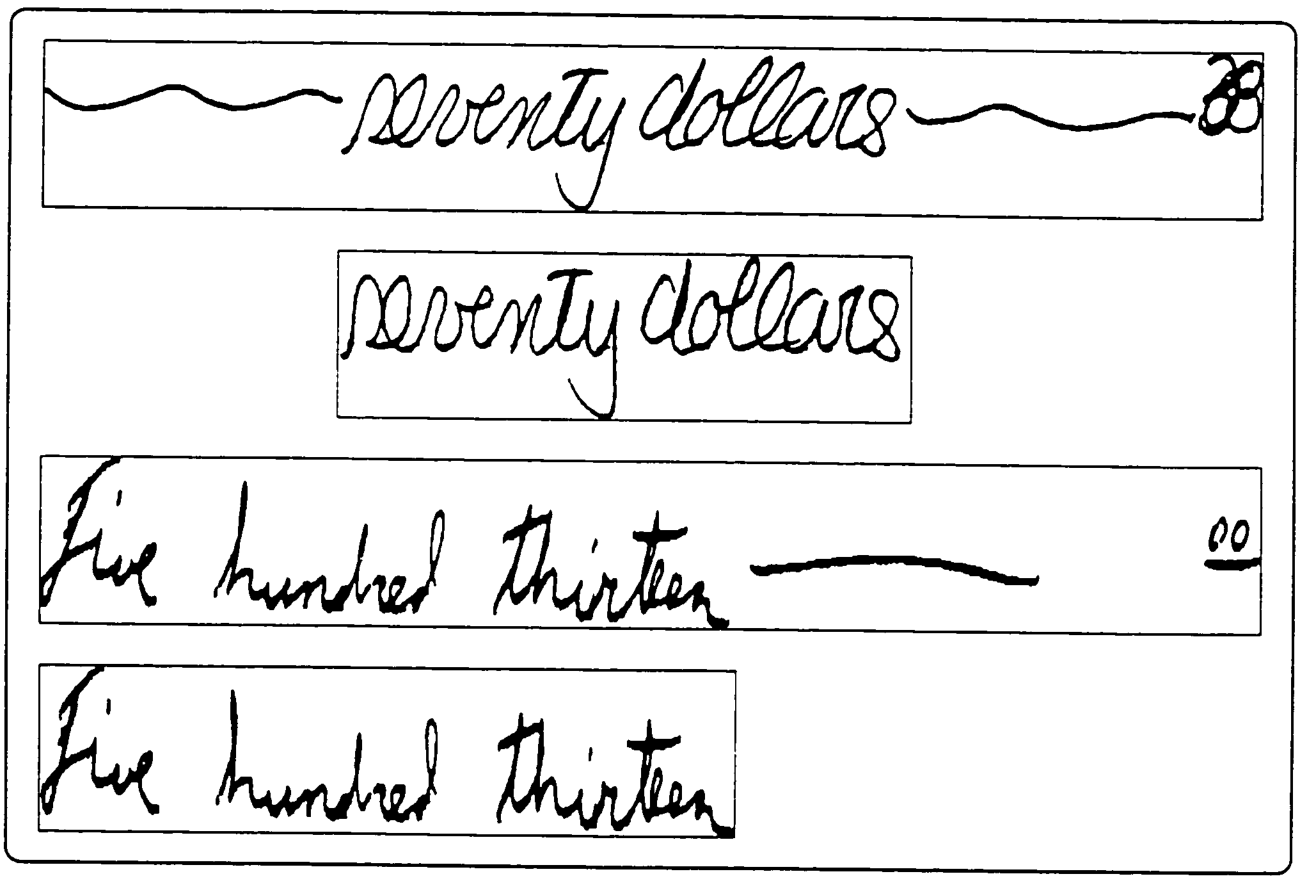


Figure 14: Both line and digit removal

Chapter 4

Feedback System

Feedback system is a critical part in our approach. It analyzes the outputs from the two segmentation steps and the classifier. Then, it decides the next task to be done and the direction of the data flow.

4.1 Feedback from First Step Segmentation

After pre-segmentation, the legal amount is split into a list of segments called word candidates. If a segment contains more than one word, it is an under-segmented case. If a word is split into several segments, it is an over-segmented case. Both cases are not segmented correctly. As we mentioned before, the effort to reduce one case will possibly increase the other. However, it is possible that we control the threshold for gap classification to a certain range so that either under- or over-segmentation is reduced to a very low level. Then we can deal with only one case of them. In our approach, we trained the threshold (t_9) in a certain range so that the over-segmentation cases are minimized after pre-segmentation. Then the structural information is collected by the feedback system to find the possible under-segmented segments. Before this process, by analyzing the lexicon, we obtained the criterion to find the possible under-segmented word candidates and the structural information to be collected.

4.1.1 Analysis of Lexicon

There are 32 words in the lexicon of the CENPARMI Cheque Processing System as shown in table 3.

one	eleven	ten	hundred
two	twelve	twenty	thousand
three	thirteen	thirty	and
four	fourteen	forty	dollar
five	fifteen	fifty	only
six	sixteen	sixty	
seven	seventeen	seventy	
eight	eighteen	eighty	
nine	nineteen	ninety	

Table 3: Lexicon in the CENPARMI Cheque Processing System

The longest word "seventeen" has nine characters. The next five longest words "thirteen. fourteen. eighteen. nineteen. thousand". have eight characters each. The average length of all the words in the lexicon is 5.66 characters. Thus. the ratio of the length of the longest and the second longest words to the average length of all the words in the lexicon are 1.59 and 1.41 respectively. If we define $t_{10} = 1.50$. which is the mean value of 1.59 and 1.41. then t_{10} can be considered as a theoretical value to identify the possible under-segmented word candidates. Once the ratio of the length of a segment to the mean length of all the segments in a legal amount exceeds t_{10}. under-segmentation may occur.

4.1.2 Analysis of Experimental Result

On analyzing the experiment result described in section 2.2.3, we find that in 85.4% of the under-segmented cases, the longest segments contain more than one word. Thus, to any legal amount image sample. among all the segments generated by the pre-segmentation. the longest one is the most probably under-segmented part.

In 13.4% of the under-segmented cases. the second longest segments are also not correctly divided. In these cases, the lengths of the longest and the second longest segments are very close to each other. Sometimes, there are only two segments in the output from pre-segmentation and their lengths are similar. We also find that in 15.5% of the under-segmented samples, the pre-segmentation result contains only one segment.

4.1.3 Selection of Possible UNder-segmented Segments (PUNS)

We developed a PUNS selection algorithm based on the above analyses. First, the output segments from the pre-processing are sorted according to their lengths. The mean length of all segments is calculated. Then, the ratio of the length of the longest segments to the mean length of all the segments (t_{11}) is calculated as:

$$t_{11} = \frac{max\{l_i | i = 1, \cdots, n\}}{\frac{1}{n}\sum_{i=1}^{n}(l_i)} \tag{1}$$

where l_i is the length of the i^{th} segment, n is the number of segments. If $t_{11} > t_{10}$, we consider the segment as a PUNS. If there is only one segment in the pre-segmentation result and it is longer than an empirical threshold t_{12}, it is also a PUNS. If the length of the longest and the second longest segments are close, we consider both of them as PUNS. The algorithm is shown below:

Algorithm: PUNSSELECTION($x_1, \cdots, x_n$)

$S \leftarrow \emptyset$
($x_{l_{max}}, l_{max}$) $\leftarrow$ longest segment and its length
if $n > 1$
  then ($x_{l_{max2}}, l_{max2}$) $\leftarrow$ second longest segment and its length
$l_{mean} \leftarrow$ mean length of all the segments
if $n = 1$ and $l_{max} > t_{12}$
  then $S = \{x_{l_{max}}\} \cup S$
if $n = 2$

$$\begin{cases} \textbf{if } l_{max} > t_{13} * l_{max2} \\ \quad S = \{x_{l_{max}}\} \cup S \\ \textbf{else} \\ \quad S = \{x_{l_{max}}, x_{l_{max2}}\} \cup S \end{cases}$$

if $n > 2$

$$\begin{cases} t_{11} \leftarrow \frac{l_{max}}{l_{mean}} \\ t_{14} \leftarrow \frac{l_{max2}}{l_{mean}} \\ \textbf{if } t_{11} \geq t_{10} \text{ and } (t_{11} - t_{14}) > t_{15} \\ \quad S = \{x_{l_{max}}\} \cup S \\ \textbf{if } t_{11} < t_{10} \text{ and } (t_{11} - t_{14}) < t_{15} \\ \quad S = \{x_{l_{max}}, x_{l_{max2}}\} \cup S \end{cases}$$

return (S)

where x_i is the i^{th} segment, t_{12}, t_{13}, t_{15} are empirical values.

4.2 Second Step Segmentation

The second segmentation step makes a re-segmentation on the PUNS that we obtained from the previous feedback step. Several hypothetical segmentation points are given out and each PUNS maybe split into several pieces. In this step, we use the same method as we used in the pre-segmentation. However, the threshold for word gap classification (t_{16}) is smaller than that (t_9) in the pre-segmentation (refer to section 4.1) because we want to split the PUNS that are not segmented successfully. The training process also shows that when $t_{16} < t_9$, the best training result can be achieved.

When each PUNS is segmented into several pieces, these pieces are combined together into hypothesis paths. Then, a classifier is used to recognize them and give out confidence values. The values are used by the feedback system to generate the best combination.

4.3 Feedback from Classifier

The purpose of introducing a classifier after re-segmentation is for the feedback system to acquire useful information in the form of confidence values. Here, we assume the confidence value of a word is higher than the value of part of a word. We are not trying to get final recognition result at this stage. However, the classifier should be effective in providing useful confidence value in a fairly short time. Thus, there are three requirements for the selection of classifier. First is that the recognition rate of the classifier should be as high as possible. A low recognition rate classifier may ruin the performance of the segmentation module. Second is that the confidence value should be useful. The value should reflect the distance of the input to the class it might be. The values of different hypothesis paths should be comparable. At last, the classifier should have high efficiency. The execution of the segmentation module should not take too much time.

When incorporating the classifier after re-segmentation, not all the PUNS need the recognition from the classifier because some PUNS remain intact. They are not split into pieces. For these cases, we send them directly to the recognition module. Only those that are split into more than one piece in re-segmentation are recognized. It is very likely that some PUNS have been over-segmented. Thus, combination needs to be done. For different combinations, the classifier generates different confidence values. However we do not want to try all the possible combinations by the classifier because the computation complexity would be too high. Hence, we adopt a simplified algorithm for the generation of the best segmentation hypothesis as illustrated below:

Algorithm: RESEGGENERATION($x_1, \cdots, x_n$)

if $n = 1$
  then $S = \{x_1\} \cup S$
else

$$
\left\{
\begin{array}{l}
S \leftarrow \emptyset \\
\textbf{for } i \leftarrow 1 \text{ to } n \\
\quad \textbf{for } j \leftarrow i \text{ to } n \\
\quad\quad \text{Try selected combinations from } x_i \text{ to } x_j \\
\quad\quad \text{Get confidence value } c_{ij} \text{ from classifier} \\
\quad \text{end } \textbf{for} \\
c_k = max(c_{ij}, \cdots, c_{in}), \quad k \in [j, n] \\
S = \{x_k\} \cup S \\
i = k + 1 \\
\text{end } \textbf{for}
\end{array}
\right.
$$

return(S)

We developed a Dynamic-length Window Sliding (DWS) method to select the combinations of the pieces from the PUNS. As indicated by its name, the method uses a window with dynamic length to slide on the pieces. The pieces inside the window are combined and recognized. Begining from the leftmost piece, the window extends to the right and finds a combination with the highest confidence value. Then the window slides to the piece next to the rightmost piece in this combination and goes through the same procedure again until all the segmentation points are generated. Figure 15 shows how the method works.

Leroux et al. proposed a similar segmentation-by-recognition strategy in the sentence-to-word segmentation of legal amounts on bank cheques [17]. In their approach, the recognizer is used from the beginning of the segmentation. But in our study, we did not incorporate the classifier from the beginning because there may be too many segments after the pre-segmentation. When using a feedback system and a classifier to select the best hypothesis path, the time cost will be very high. A direct consequence is that the processing time of the system may be multiplied according to the number of pieces from the PUNS. In our system, each time the classifier is called, the pieces need to be combined come from only one segment after pre-segmentation. Thus, the number of pieces is fairly small compared with the number coming from direct segmentation of the original legal amount image. It makes the implementation of our combination approach practical and effective.

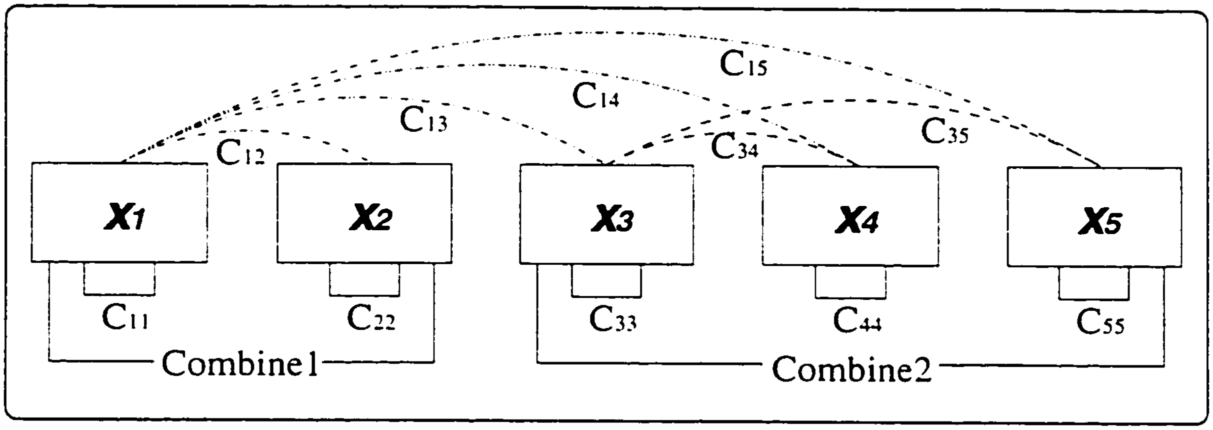


Figure 15: Dynamic-length Window Sliding method

The feedback system generates several new segmentation points. These points are merged with the hypothetical segmentation points from pre-segmentation to form the final segmentation result. A list of word images can be obtained from the final segmentation result. These word images are then recognized by classifiers.

Figure 16 shows the complete segmentation procedure of an original sample image. to the results of the pre-segmentation. re-segmentation and feedback system. It is a typical example that shows the effectiveness of our segmentation approach.

The lines and digits in the image are removed. The connected components are comnined into 16 groups. The three largest gaps between the groups are: G_3 (59 pixels). G_2 (25 pixels) and G_1 (23 pixels). Then the pre-segmentation step splits the image into S_{11} (550 pixels in width) and S_{12} (255 pixels in widths) at the place of the largest gap. However. because G_1 and G_2 are close to each other and much smaller than G_3. words "eight" and "hundred" are not separated successfully. The feedback system finds that there are only two segments after pre-segmentation and the width of S_{11} is almost twice that of S_{12}. it considers S_{11} as a PUNS. Then the re-segmentation step uses a smaller threshold to re-segment the PUNS. but it splits the word "hundred" into two parts. To combine the word pieces into words. the classifier gives out the following recognition results and confidence values:

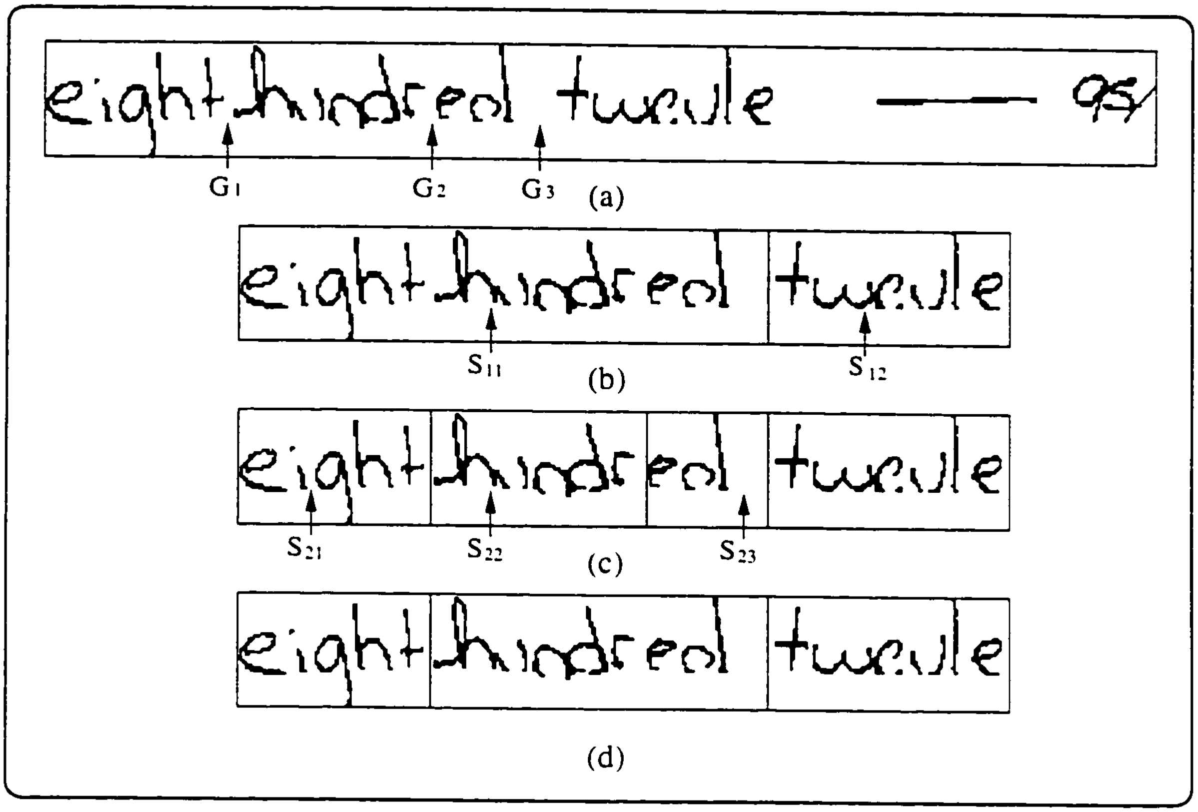


Figure 16: A legal amount sample and segmentation. (a) Original image: (b) result from pre-segmentation: (c) result from re-segmentation; (d) final segmentation result.

- "eight": eight. 0.9214.
- "eighthundr": eighteen. 0.4386.
- "eighthundred": thousand, 0.2999,
- "hundr": thousand. 0.1928,
- "hundred": hundred. 0.9177.

Since the combinations of "eight" and "hundred" have the highest confidence values. the feedback system outputs them as well as the rest of the legal amount as the final segmentation result.

Chapter 5

Recognition of Word and Legal Amount

The output of the segmentation module is a list of word images. The recognition module is then called to identify the words. For a given input, the word classifiers give a ranked list of top five classes according to their confidence values. Figure 17 shows an example of word recognition results. An English legal amount parser is then called to parse the recognition results. It produces a ranked list of syntactically and semantically correct legal amounts.

5.1 Recognition

How to develop a word classifier is beyond the scope of this thesis. However a discussion on the classifiers can not be omitted because they are used in both segmentation and recognition. In the following subsections, we introduce two word classifiers that are currently available in CENPARMI.

5.1.1 An MLP Classifier

Strathy designed a classifier that uses whole word recognition with classification by multiple neural networks trained by back propagation (MLPA) [29]. The classifier is

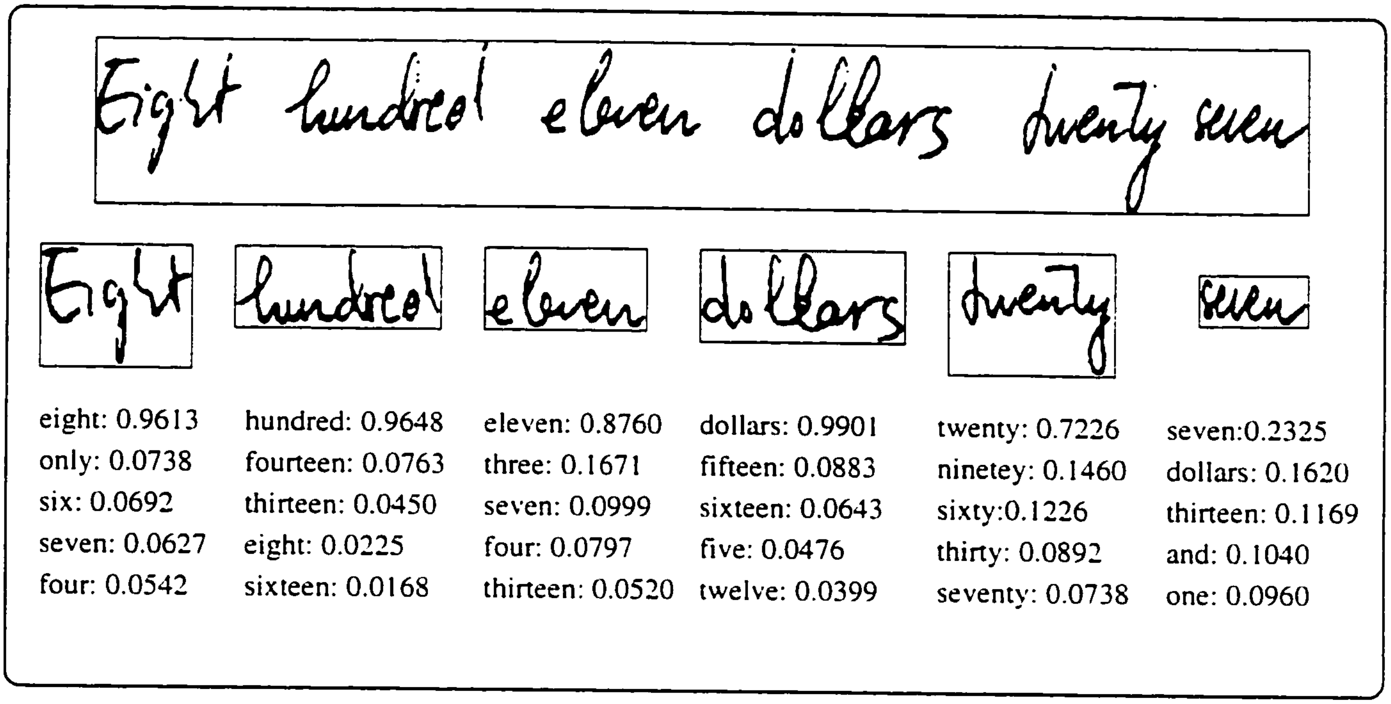


Figure 17: Example of word recognition results and their confidence values

composed of three simpler neural networks each with a hidden layer. MLPA adopts a loosely coupled structure as shown in Figure 18. It uses a majority voting strategy. The confidence value is obtained from the sum of the outputs from the neural networks. The classifier is trained on 5317 words from 32 classes. The test recognition result on 2514 words is 86.4%.

5.1.2 An HMM-MLP Classifier

Kim et al. developed an HMM-MLP hybrid method [13]. This method uses a multiplication scheme with weighting factor for the heterogeneous classifiers and a fusion scheme for the homogeneous neural networks. At first, two homogeneous MLPs are implemented separately and combined into a new single MLP (MLPB) classifier at

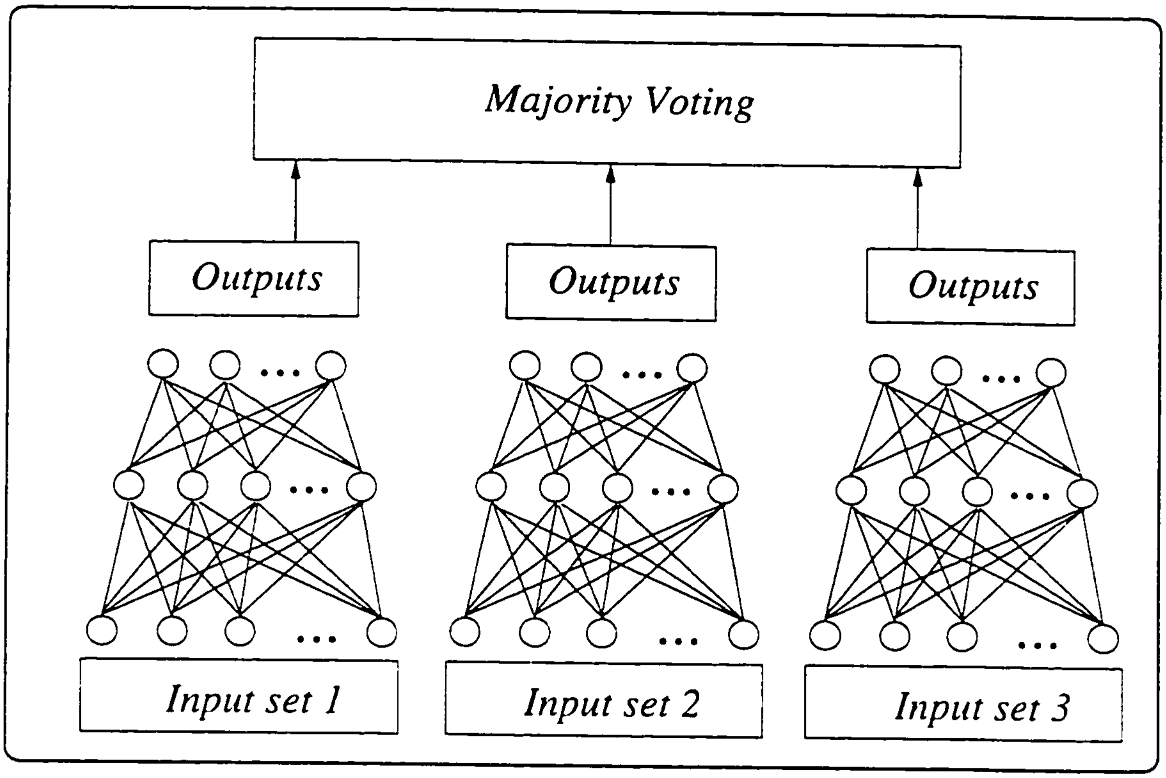


Figure 18: Implementation scheme of MLPA

the architectural level. Next, HMM is combined with the new MLP as a heterogeneous one. The classifier is trained on 5,223 words from CENPARMI's real cheque database. It was tested on a legal word database with 2,482 words and with a 92.7% recognition rate. The performances of single HMM and MLP on the same database are 82.0% and 87.9% respectively.

Implementation of MLPB Classifier

The performance of MLP is largely dependent on the input features as well as its topology. Five basic features are extracted from the word image:

1. Mesh feature is the number of black pixels in each of the sub-divided local regions of the binary images.

2. Chain feature is the number of same directional pixels in each of the sub-divided local regions of the contour image.

3. Crossing feature is the number of times the strokes are crossed by projection lines which are defined as equally spaced horizontal and vertical lines.

4. Distance feature is the distance from the minimum boundary rectangle of the word to the first black pixel of the word image.

5. Gradient feature is the number of pixels that have the same gradient angle in each of sub-divided local region of the binary image.

After extracting the above features, two separate MLPs are implemented by using each input feature set. Then a new MLP is implemented by using the outputs of the neurons in two hidden layers as a new input feature, as shown in Figure 19. Thus, MLPB is composed of three MLPs.

Implementation of HMM Classifier

The HMM classifier uses an explicit segmentation scheme for cursive words to obtain a pseudo temporal sequence. The pseudo temporal sequence is extracted from a sequence of segmented parts of words that are called graphemes. The graphemes are obtained by segmenting words into characters according to the local down or up hills included in the ligatures.

The segmentation process produces several cascaded graphemes that may include over- or under-segmentations of the cursive word. The implementation of HMM classifier is based on the segmentation. Its topology is shown in Figure 20 where the single state transition a_{ij} represents a single grapheme and three cascaded states represent a single character. During the state transitions, we can observe the output probability $b_{ij}(o_t)$ of the given observation symbol sequence **O**. Each character has three states s_i and three different state transition paths to represent over-, exact- and under-segmentations.

During the null transition a_{ij}, the output symbol probability is not observed since a null transition represents "letter skip" due to under-segmentation. So on the way of

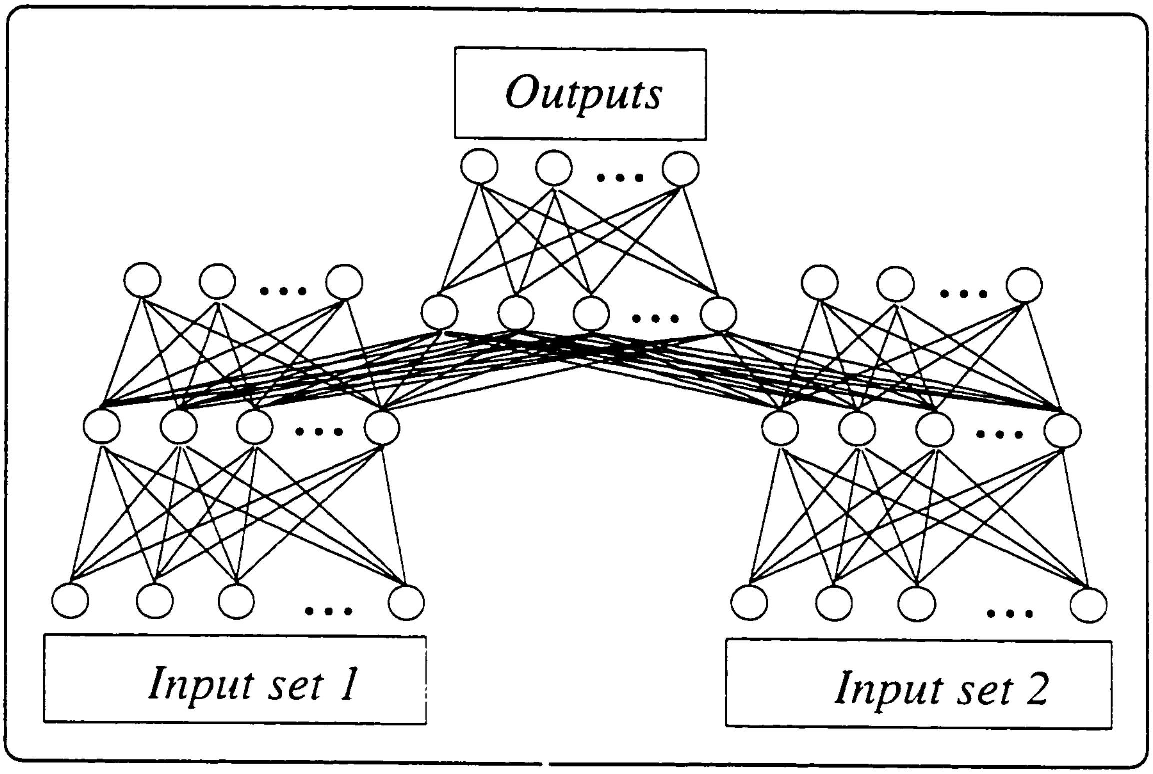


Figure 19: Implementation scheme of MLPB

state transition along the fixed single transition path. if the null transition is included. the observation time is not increased while the state is changed to the next one.

Combination of MLP and HMM Classifiers

Before the combination, each output of HMM and MLP is normalized to obtain equal level measurement values. Then the classifiers are combined by a weighted multiplication method. The probability of each class is defined based on the multiplication of the probabilities with weighting factors which are defined according to the performance of each classifier.

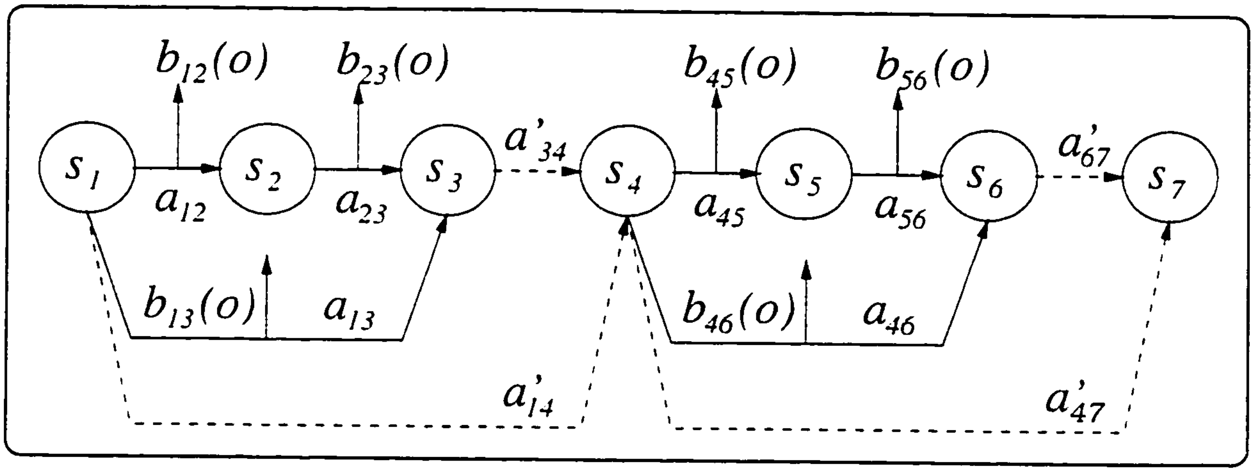


Figure 20: Topology of segmentation based HMM classifier

5.1.3 The Relationship of Classifiers in Segmentation and Recognition

Someone may have doubt on the relationship of the classifiers used in segmentation and recognition. The development of our segmentation method shows great flexibility. Thus, we can use different classifiers in the segmentation and the recognition. For example, we may use MLPA in segmentation while using MLPB in recognition. We can also use the same classifiers in both steps. If so, the structure of the segmentation is simplified as shown in Figure 21.

In our research, we trained and tested several combinations of classifiers in segmentation and recognition. For example, using MLPA in both segmentation and recognition, using MLPA in segmentation and HMM-MLP in recognition, using MLPB in segmentation and MLPA in recognition, and so on. Their performance will be reported in chapter 6.

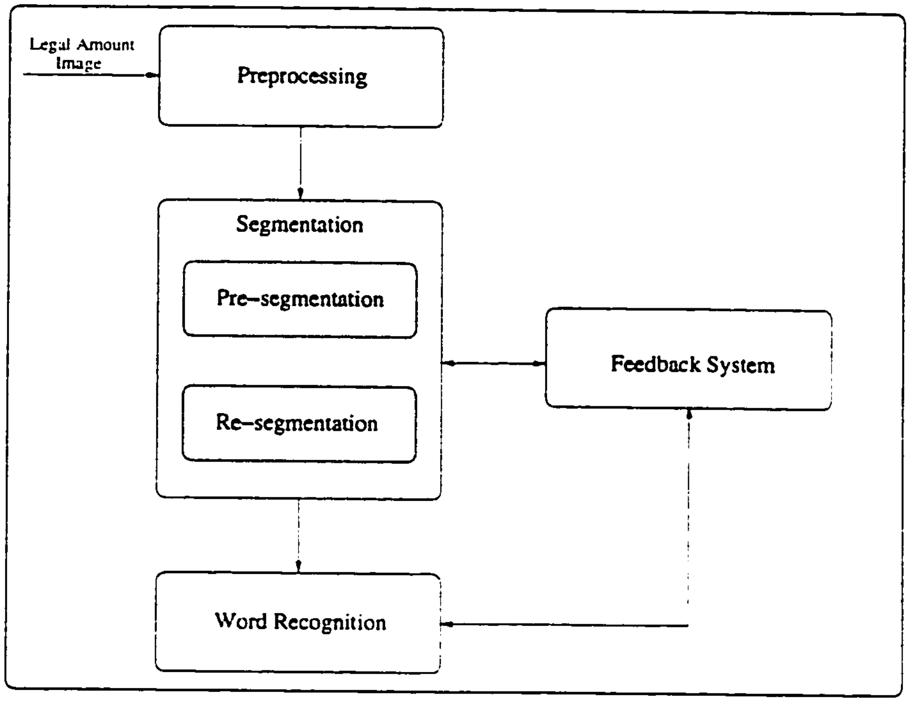


Figure 21: Structure of segmentation module containing the same word classifier as recognition

5.2 English Legal Amount Parser

An English legal amount parser has been designed by Guillevic [4]. The grammar can parse the cheque amount up to a value corresponding to $99,999.99. This limit is set for the security of the cheque processing system. Cheques with bigger amounts should be processed by human beings. The parser can parse both the dollar part and the cent part. However, it can not be used directly in our system.

In the lexicon of our cheque processing system, there is no word "cent". Thus, the system can not parse the amount as shown in Figure 22. Besides that, some legal amount cases are not considered by the parser provided by Guillevic. An example is shown in Figure 23 (a), whose legal amount ends with the word "and", with the cent part written in digit. Another example is shown in Figure 23 (b), in which the legal amount begins with the word "only".

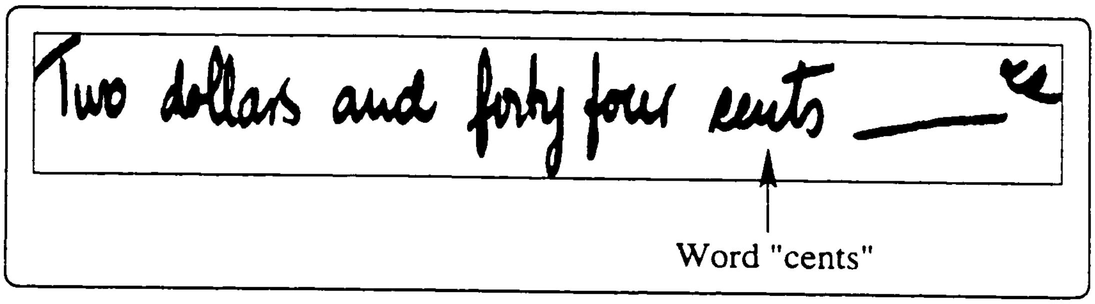


Figure 22: A legal amount with word "cents"

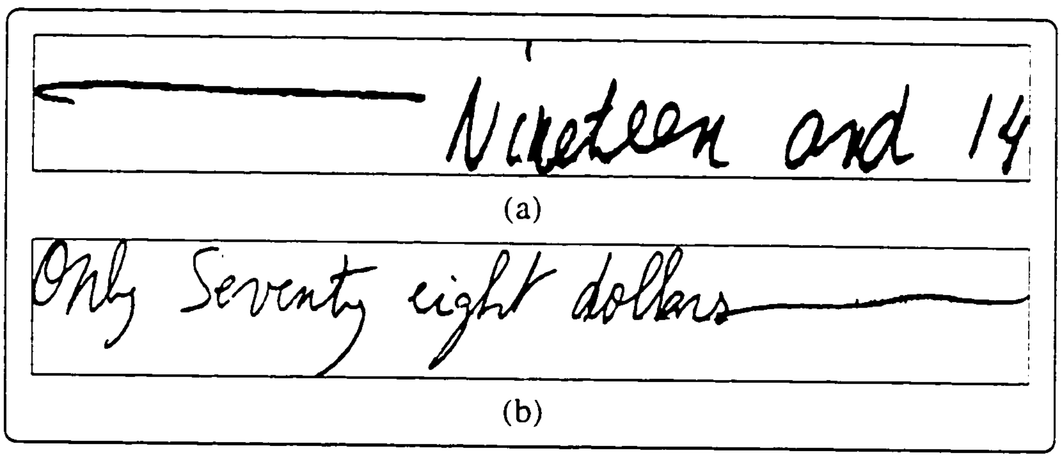


Figure 23: Legal amount examples. (a) An amount ends with the word "and" ; (b) An amount begins with the word "only".

To fit the database and the classifiers, we modified some part of the grammar in Guillevic's parser. The new grammar is shown in Table 4.

In the grammar, identifiers beginning with a capital letter are non-terminal symbols. 'S' represents the starting symbol. The keywords "numbers", "teens" and "tys" represent the classes of terminal symbols described in Table 5. The string "teensminusten" represents the set of terminal symbols in "teens" and "ten". The terminal

S	$\rightarrow$	(numbers | teensminusten) S1 | tys S2 | ten S3
S1	$\rightarrow$	hundred H | thousand T | E
S2	$\rightarrow$	(numbers | e) S1
S3	$\rightarrow$	thousand T | E
H	$\rightarrow$	(and | e) H1
H1	$\rightarrow$	(numbers | teens | e) E | tys D
T	$\rightarrow$	numbers T1 | (teens | e) E | and T2
T1	$\rightarrow$	hundred H | E
T2	$\rightarrow$	numbers T1 | (teens | e) E | tys D
D	$\rightarrow$	(numbers | e) E
E	$\rightarrow$	dollars G1 | and G2 | (only | e) F
F	$\rightarrow$	(only | e) F
G1	$\rightarrow$	and G2 | tys G3 | (only | numbers | teens | e) F
G2	$\rightarrow$	(numbers | teens) F | tys G3 | F
G3	$\rightarrow$	numbers F | F

Table 4: Grammar for the English legal amount

symbol 'e' represents an empty input.

numbers	=	{one, two, three, four, five, six, seven, eight, nine}
teens	=	{eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen }
tys	=	{twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety}

Table 5: Classes of terminal symbols for the English parser

When a sentence formed of legal words is fed into parser, the parsing process begins from the first word. At each node in the parsing tree, the parser judges the next input word based on the expected classes. If the word does not belong to the expected class, the sentence is not a grammatically correct legal amount.

5.3 Legal Amount Recognition Result

After word recognition, the system produces lists with top five choices for each word in the legal amount. Then the system finds all the paths that are grammatically correct by a recursive calling of the parser through the lists. At the same time, the confidence values of each path is calculated by computing the mean confidence value of the words in the path. These paths are then ranked in descendent according to their confidence value. The top ten paths are the output of the legal amount recognition system.

Chapter 6

Experimental Results

6.1 Database

A CENPARMI database has been described by Guillevic in his Ph.D. thesis [4]. To set up the database, a special cheque was designed with a white background and printed information in drop-out color, to facilitate the preprocessing of the cheque. The cheques were written by students in the departments of Computer Science and Accounting. The amounts to be written had been predefined in order to collect sufficient data on different words and numerals. Altogether, about 2,500 cheques written in English have been collected from nearly 800 different writers. A sample of the cheque is shown in Figure 24.

These cheques were scanned at 300 DPI with 8-bit greyscale. The legal amounts were then extracted and binarized by a user interface and stored in Tag Image File Format (TIFF) files to form the legal amount database. For each image file, a tag file was made to store the position and content information of words, characters, lines, digits and so on. Figure 25 shows a legal amount image and its tag information. Before using the database, we carefully removed all the images with word "cent" in the legal amount because this word is not included in the lexicon of our word classifier.

The database is then divided into a training set which contains 1431 images and a testing set which contains 680 images.

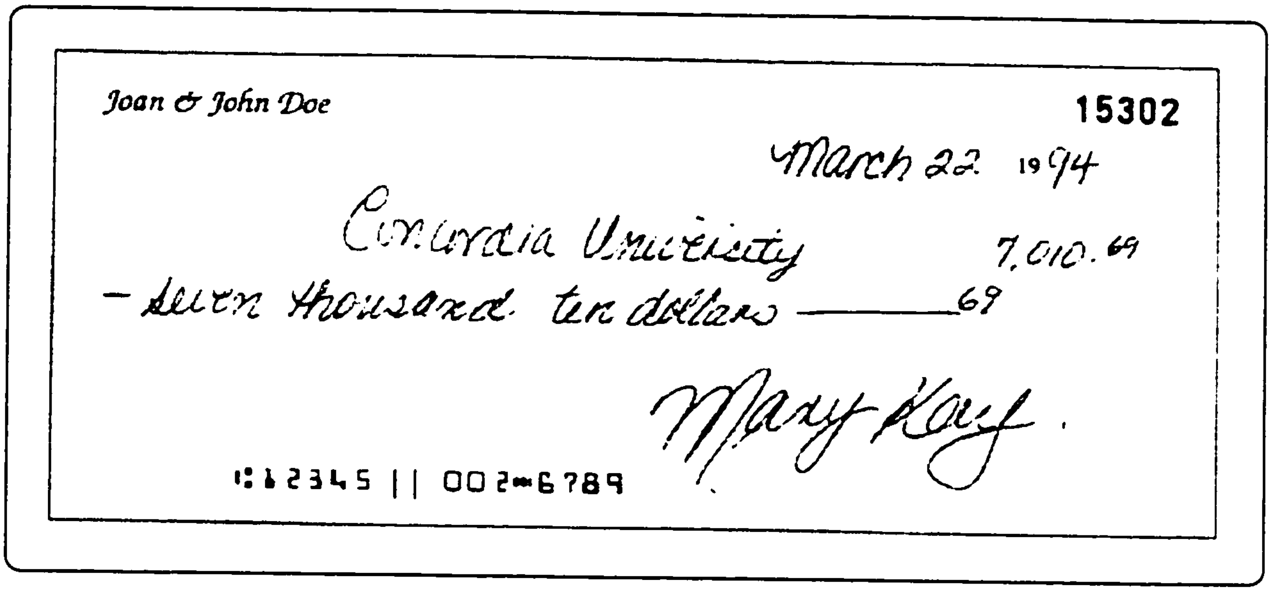


Figure 24: A cheque sample for the CENPARMI database

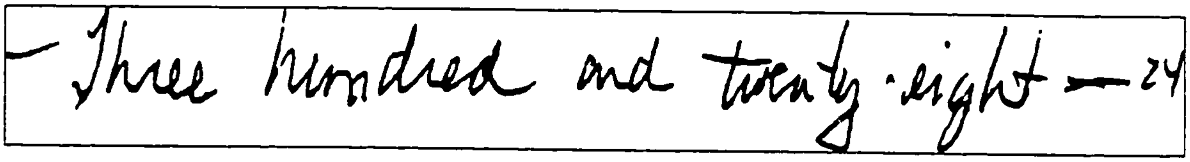
6.2 Environment

We have implemented the proposed approach using a C programming language on a SUN Ultra 60 workstation. The workstation has dual UltraSPARC-II 296M processors and 768M main memory. The operation system is Solaris2.6.

6.3 Training

In our approach, the thresholds used to distinguish the inter-word and inter-character gaps in pre-segmentation (t_9) and re-segmentation (t_{16}) are trained separately so as to achieve the best segmentation performance (refer to section 4.2). Both thresholds measure the smallest gap that can be considered as a possible inter-word gap in terms of pixels between two connected components. The thresholds are trained on training set under the following conditions.

- Previous segmentation approach (SEG1)



Tag information:

–: 60, 29; #: 80, 61; T: 125, 67; h: 168, 58; r: 194, 65; e: 221, 68; e: 260, 68;
#: 315, 68; h: 362, 58; u: 412, 55; n: 447, 75; ~: 461, 64; d: 515, 48; r: 536, 58;
e: 570, 51; d: 658, 57; #: 708, 53; a: 738, 57; n: 773, 59; d: 828, 56; #: 861, 34;
t: 991, 64: w: 944, 52; e: 971, 61; n: 1004, 66; t: 1028, 63; y: 1061, 59;
#: 1088, 55; –: 1103, 53; #: 1111, 69; e: 1145, 56; i: 1166, 64; g: 1201, 74;
~: 1208, 75; h: 1244, 64; t: 1294, 50; #: 1318, 51; –: 1397, 51; #: 1409, 57;
2: 1434, 41; 4: 1462, 0;

Figure 25: A legal amount image and its tag information. The tag format is: "ch: x, y", in which 'ch' is the character, 'x' and 'y' are the x- and y- coordinates of the starting point in the contour of the character. In the tag, '#' indicates an inter-word gap, '~' indicates an inter-character gap, and '-' indicates a horizontal line.

- Proposed feedback-based method with MLPA (FB-MLPA)
- Proposed feedback-based method with MLPB (FB-MLPB)

We trained the threshold t_9 in SEG1. The threshold is incremented in a pace of 2 each time. The value with the highest performance is used in the testing. Figure 26 shows how the number of correctly segmented samples changes with different thresholds. When t_9 equals 24, we got the best training result.

When training FB-MLPA, we first fixed t_{16} to a certain value and trained t_9. The value with the highest performance was recorded. Then, we trained t_{16} while keeping t_9 to the value we had just obtained. The training result is shown in Figure 27. From the training result we find that when t_9 is bigger than t_{16}, the segmentation approach can achieve the best performance. It is consistent with our intention to use a bigger threshold in the pre-segmentation to control the over-segmentation, and then use a smaller value of re-segmentation to split the under-segmented segments.

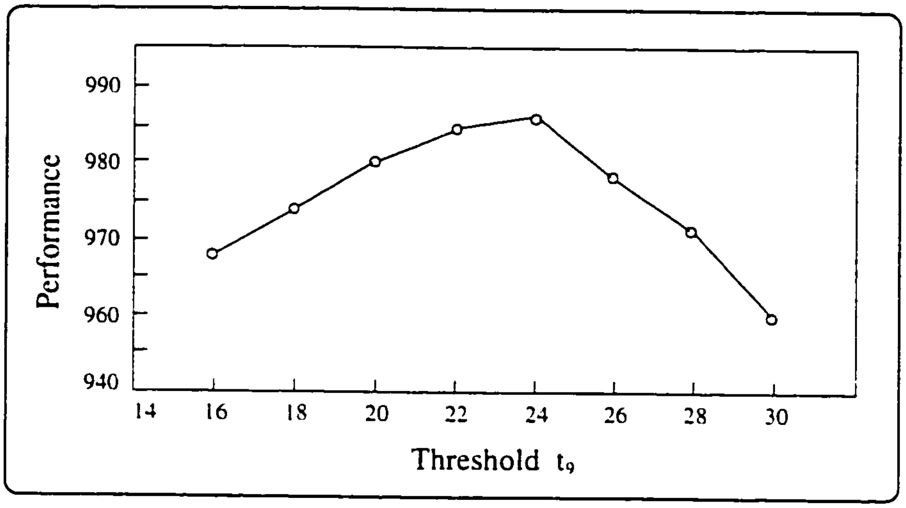


Figure 26: Training result of SEG1

We use the same method to train the FB-MLPB, but we did not use and train the HMM classifier in the segmentation because its performance is the worst among the classifiers and their combination. The HMM-MLP hybrid classifier is also useless for the segmentation because the produced confidence values for combined word pieces are binarized. The confidence value of the top one choice of the recognition result is close to 1, while the values of the second to fifth choices are usually close to 0. Thus, we can not compare the confidence values of different combinations of the pieces, which prevents the generation of segmentation points.

6.4 Test Result of Legal Amount Segmentation

6.4.1 Comparison of Test Results

We tested the above mentioned segmentation approaches on the test set containing 680 legal amount images. The results are shown in Table 6.

Comparing with the previous approach, the two feedback-based approaches increase the correct segmentation rates by 15.0% and 5.1% respectively. However, if we

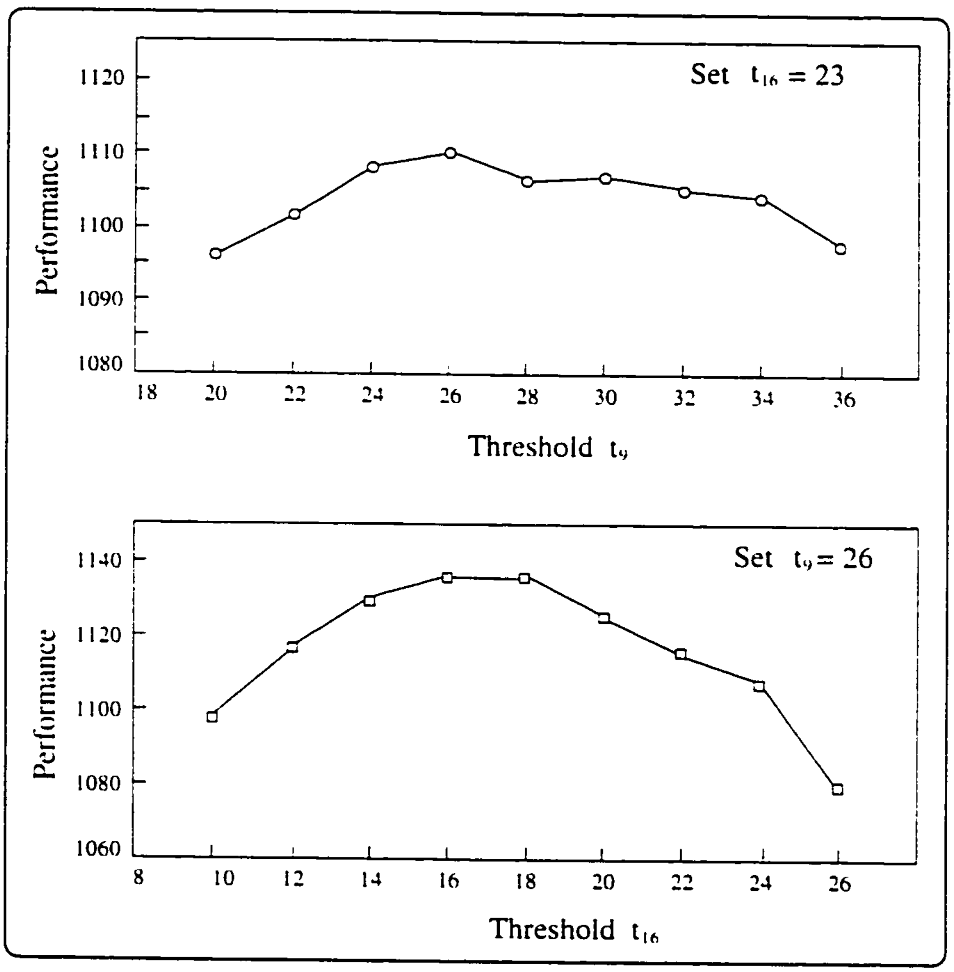


Figure 27: Training result of FB-MLPA

	Correct(%)	Under(%)	Over(%)	Hybrid(%)	Time
SEG1	66.8	29.4	3.1	0.7	1.0
FB-MLPA	81.8	12.0	5.3	0.9	1.6
FB-MLPB	71.9	16.5	4.4	0.6	1.8
MAN	96.2	3.8	0	0	—

Table 6: Comparison of legal amount segmentation results

consider the average processing time for legal amount by SEG1 as one time unit, the time costs for the two proposed approaches increase by 60% and 80% respectively. One point should be mentioned here is that in the manual segmentation (MAN) of the samples in the test set, the correct segmentation rate is 96.2%. The errors solely come from samples with words connected physically. This rate can be considered as the perfect segmentation rate for the test set. Thus, there is 14.4% difference between the best performance of the proposed approach and the perfect segmentation result.

To compare our work with the other researchers is difficult due to the use of different databases. Knerr et al. [15] reported a 76.7% top one segmentation rate on 2000 French legal amount images extracted from a postal cheque database.

Using a method introduced in section 1.2, Kim et al. [11] reported an 80.0% correct segmentation rate without mention of the database. The method was measured by the recognition result of the legal amount. If the true value was present in the list of recognized amount candidates, the segmentation was correct. The average number of candidates in the lists is 107.

6.4.2 Analysis of Test Results

We noticed that the performance of FB-MLPA is better than FB-MLPB. On the contrary, the reported performance of MLPB is better than that of the MLPA. The difference is caused by two problems in the implementation of MLPB. The first one is robustness. The design of MLPB is aimed for word recognition. Thus, it restricts the length of the words. When facing the cases that two long words are combined into one segment, MLPB tends to reject them. The second problem is fitness. MLPB tends to give a high confidence value to the top one result for each word candidate recognized. It makes the comparison of confidence values between different combination hypotheses in the feedback step not as accurate as MLPA. Even if the problems exist. FB-MLPB still can improve the segmentation performance compared with SEG1.

In the test result. "Correct" word segmentation indicates that all the words in the amount are isolated successfully. "Under" category contains the under-segmented samples in which at least one segment encompasses more than one word. "Over" category contains the over-segmented samples in which at least one word is segmented into more than one part. "Hybrid" category contains those samples that have both under- and over-segmentation cases. All the three latter categories are considered as segmentation error. Some samples of these four categories are shown in Figures 28-31.

The segmentation errors are caused by the following reasons:

- Due to the side effect of preprocessing. some small strokes. or even letters. may be considered as noises or lines and are removed. It changes the distribution of the gaps. which may incur errors.

- The gaps between words are too narrow to be detected. Sometimes. the words are even connected physically. This is one reason that causes under-segmentation.

- The classifier is not 100% accurate. If it can not classify a word correctly. or it can not provide accurate and comparable confidence values. the word pieces will not be correctly combined. Then the segmentation module can not produce the correct segmentation result. A portion of under-, over- and hybrid-segmentation errors are due to this reason.

- Even though we tried to reduce the over-segmentation in pre-segmentation. it still may happen. However. in our feedback system. we only deal with the under-segmentation cases. Thus. the over-segmentation errors are not solved and become a portion of the final errors.

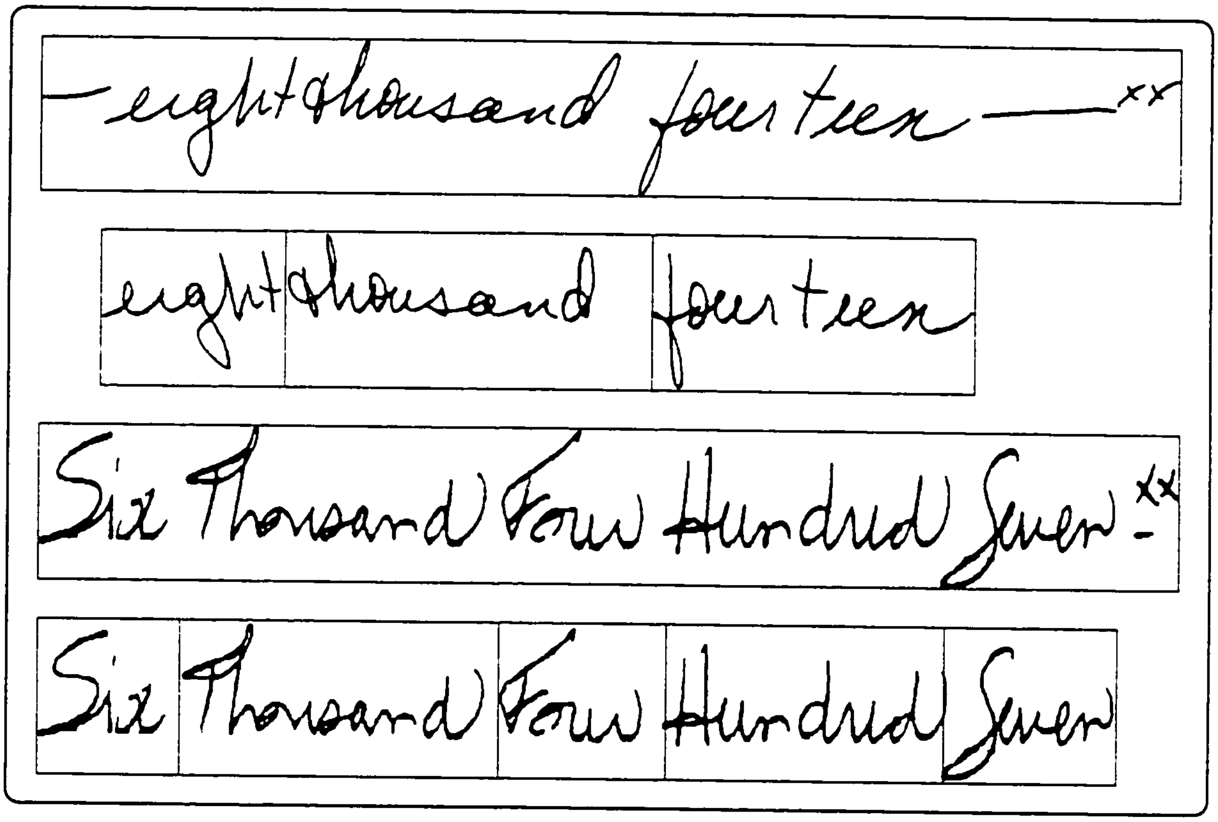


Figure 28: Correctly segmented samples of FB-MLPA

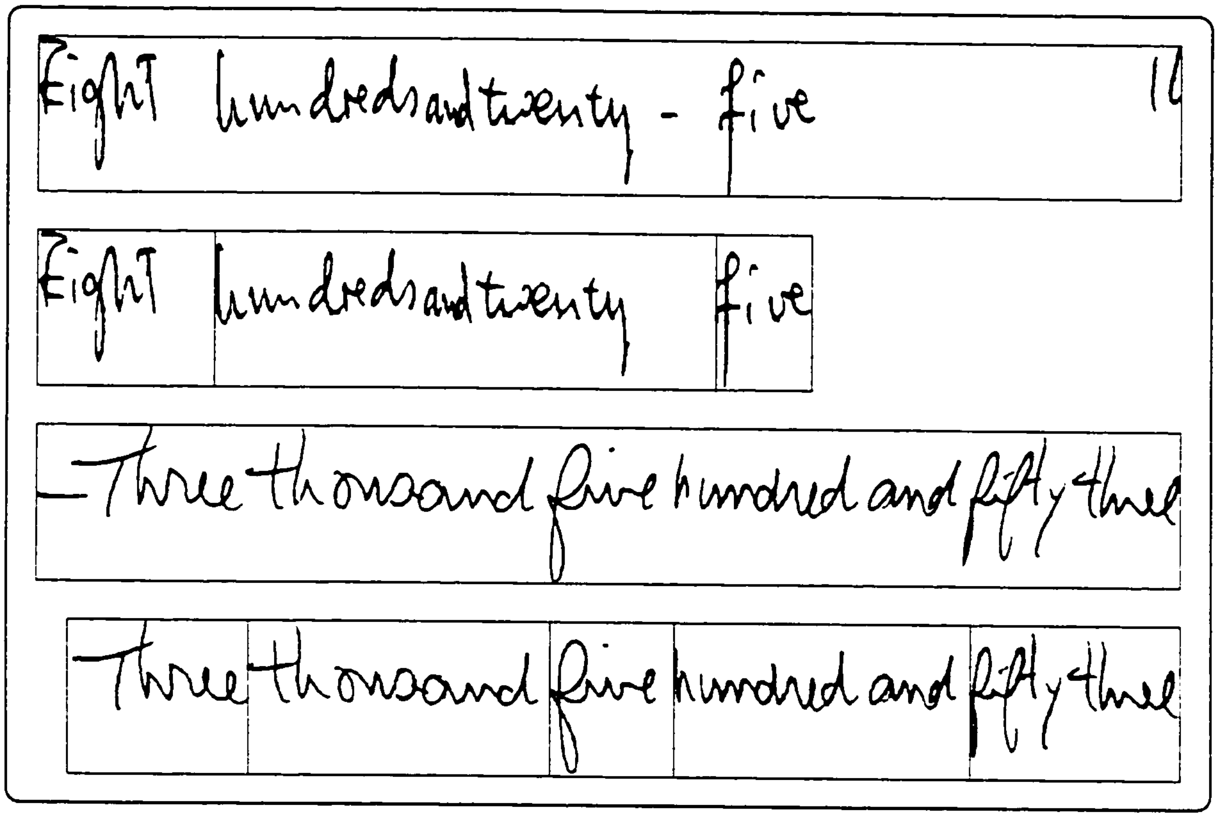


Figure 29: Under-segmented samples of FB-MLPA

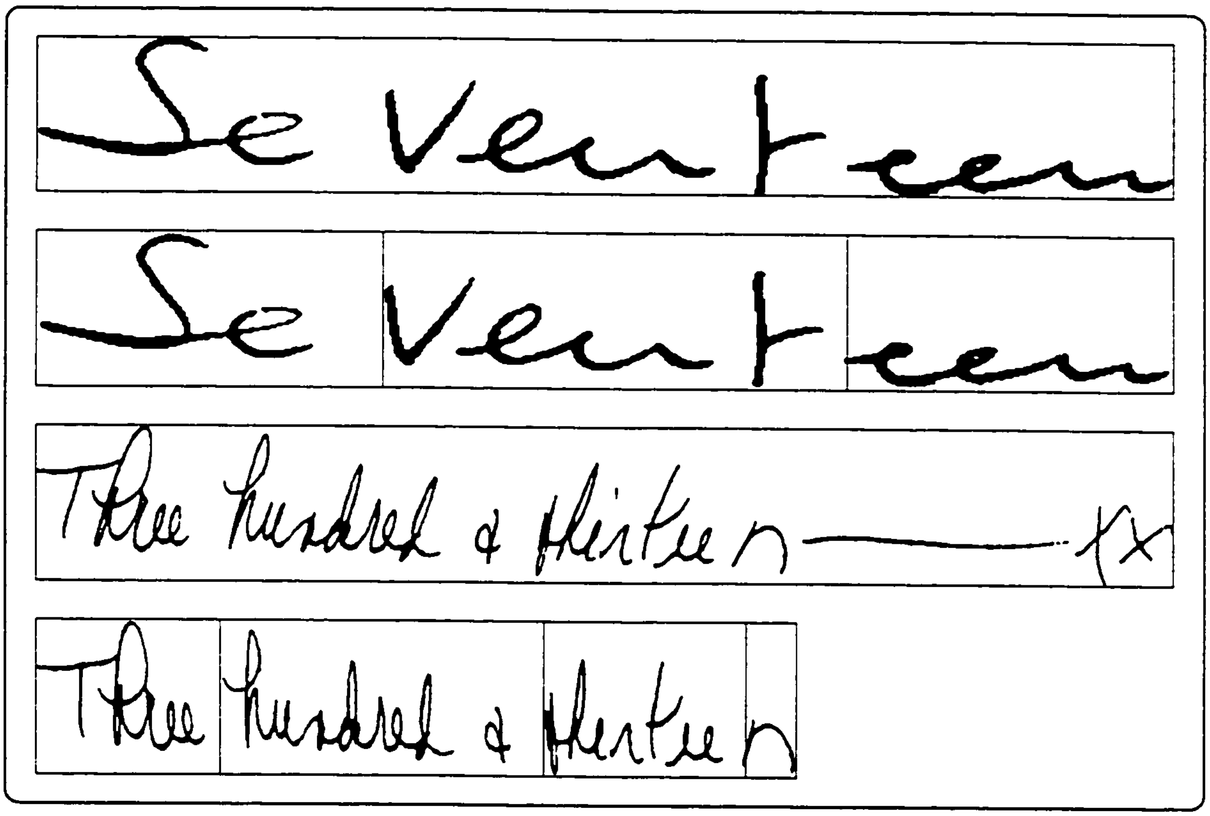


Figure 30: Over-segmented samples of FB-MLPA

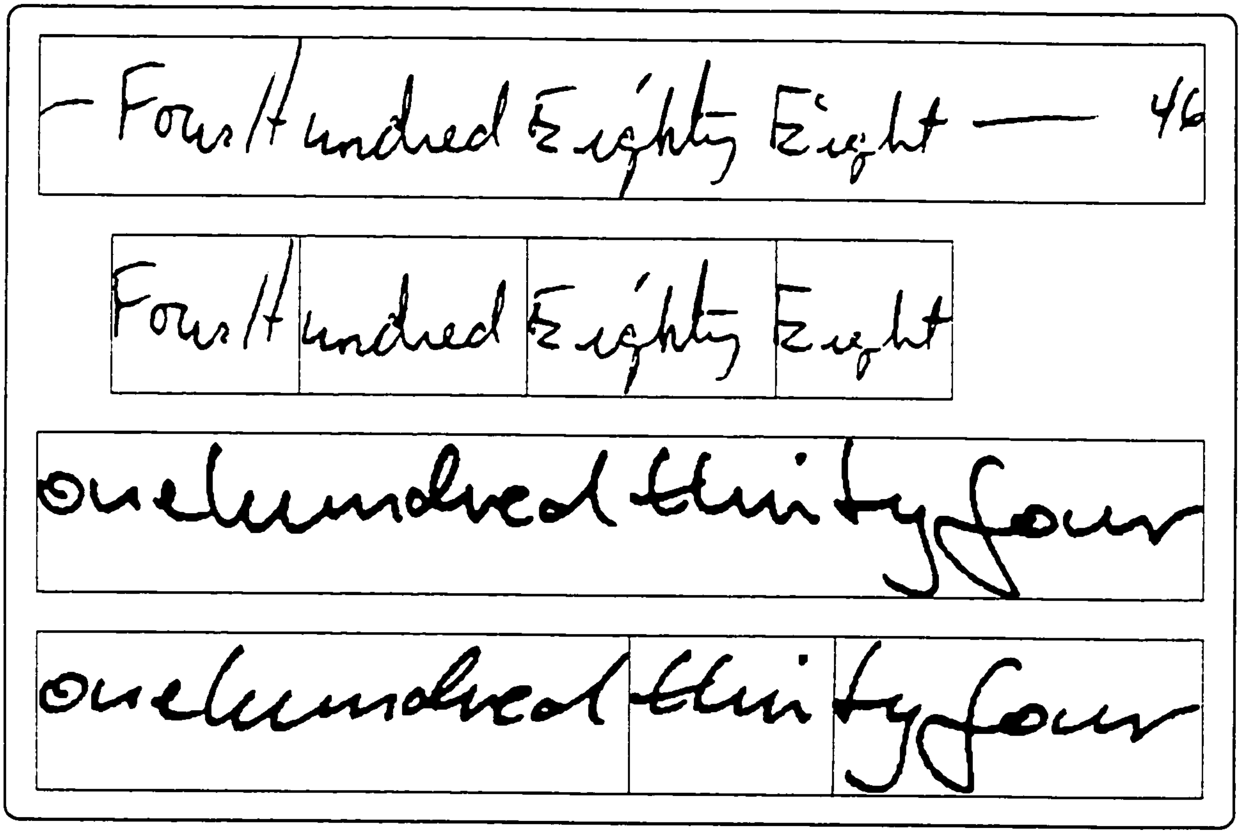


Figure 31: Hybrid-segmented samples of FB-MLPA

6.5 Test Result of Legal Amount Recognition

6.5.1 Comparison of Recognition Results

Now that we have three segmentation approaches and two classifiers, we want to compare the performances of their combinations. In the recognition phase, we use HMM-MLPB instead of MLPB because the former has a better recognition rate. We also tested the performance of the classifiers with manual segmentation. Table 7 shows the recognition results of the combinations on the test database of 680 samples.

Segmentation	Recognition	Top1(%)	Top2(%)	Top5(%)	Top10(%)
SEG1	MLPA	51.8	56.9	58.1	58.2
SEG1	HMM-MLPB	47.4	54.0	57.2	58.0
FB-MLPA	MLPA	61.5	68.0	70.0	70.5
FB-MLPA	HMM-MLPB	56.3	63.1	68.5	70.0
FB-MLPB	MLPA	55.3	61.5	63.7	68.8
FB-MLPB	HMM-MLPB	50.2	56.7	61.5	62.4
MAN	MLPA	81.3	87.2	89.4	89.6
MAN	HMM-MLPB	73.2	82.1	86.9	88.7

Table 7: Comparison of legal amount recognition results

Only a few comparable results on legal amount recognition are available from the other researchers because many results bypass the segmentation step. Knerr et al. [15] reported a 63.6% top one recognition rate with a rejection rate at 2.4% on French legal amount images extracted from a database of 32,000 postal cheques. The result reported by Kim et al [11] on an English legal amount database with 253 samples is 47.2%.

6.5.2 Analysis of Test Results

With the manual segmentation, our system gets a top one legal amount recognition rate at 81.3%. Since manual segmentation is an approximation of the perfect segmentation, we consider this result as the highest performance that our current legal amount processing system can achieve. Further improvement of the performance can only be obtained by improving the legal amount recognition system.

From the above result, we find that except for the manual case, the best combination of segmentation approach and classifier achieves top one legal amount recognition rate at 61.5% on the database. It is a 9.7% improvement compared with the performance of the same classifier with the previous segmentation approach. If we consider the top ten case, the improvement is 12.3%. Comparing with the SEG1 and HMM-MLPB combination, the FB-MLPA and HMM-MLPB combination also increases the top one recognition performance by 8.9%.

To find how the classifiers perform after the perfect segmentation, we analyze the recognition result of MLPA, which is the best one [40]. Among the recognition errors, 3.8% come from the unsuccessful separation of connected words, as shown in Figure 32. This kind of error can be considered as the mistake of segmentation. Another 1.6% errors come from the grammatical mistakes of the legal amounts, such as the amounts shown in Figure 33. The recognized amounts can not pass through the parser. The remaining 5.0% errors are caused by the mistakes of the classifier.

We also analyze the 29.5% recognition errors of MLPA after FB-MLPA. 18.2% errors come from segmentation mistakes. 2.7% errors come from useless information in the legal amount image, such as symbols, digit cent part and lines connected with words. They can not be identified and removed easily. 1.6% errors come from grammatical mistake of the legal amounts. The remaining 3.8% are caused by the mistakes of the classifier.

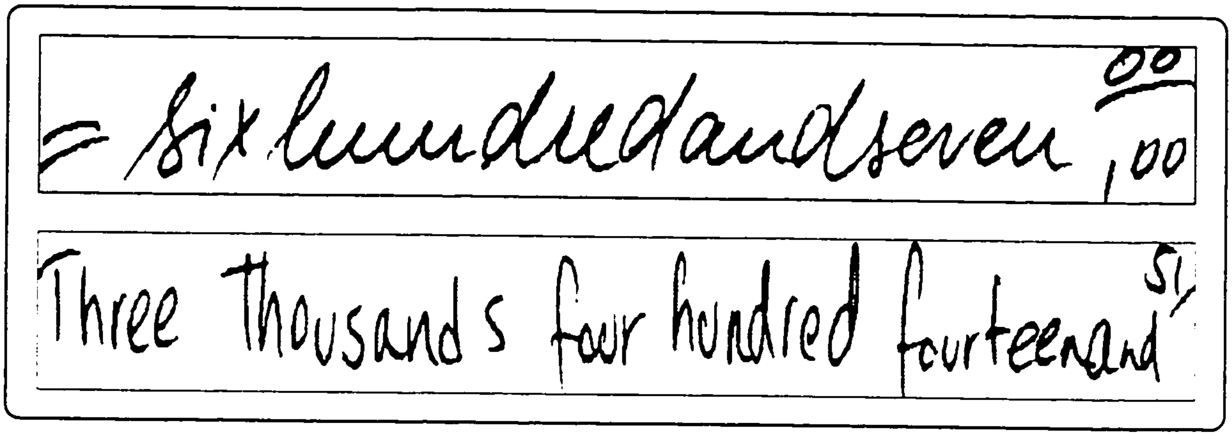


Figure 32: Unsuccessful separation of words by the manual segmentation

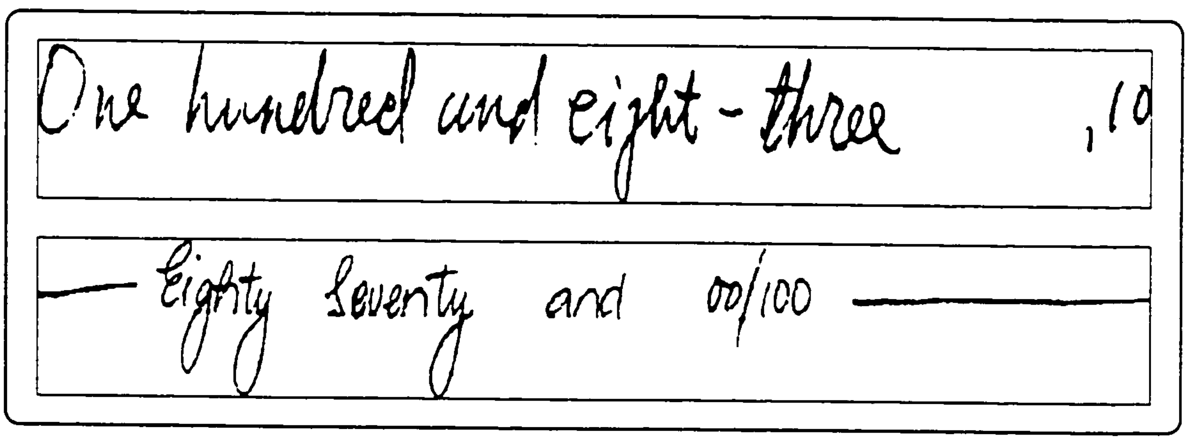


Figure 33: Grammatical mistakes in the legal amounts

Chapter 7

Conclusion

7.1 Summary of Contributions

From our study, we have found that the sentence-to-word segmentation plays an important role in the legal amount recognition. The importance comes from various reasons.

First, the segmentation serves as the basis of the recognition. Its performance greatly affects the final recognition result. With the intensive study on the word recognition engine, the performance of the classifier has been greatly improved. From our experimental results, we find that only a small portion of the legal amount recognition errors are due to the mistakes of the classifier, while the errors from segmentation form the majority. The performance of segmentation has become a bottleneck in the system. Our approach has increased the recognition rate by 9.71% because of the great improvement of the segmentation module. Even so, more than 60% of the final recognition errors are still caused by poor segmentation.

Second, segmentation and recognition in a legal amount processing system interact with each other. They are not isolated entities. Recognition can be introduced into segmentation to generate segmentation hypothesis, to verify segmentation results, and so on. In our approach, classifiers are used to provide confidence values. These values are used to generate the segmentation result. Then, the segmentation result is recognized by the classifiers. This approach can solve the segmentation problem very well. However, the classifiers are not specially designed for the segmentation. Thus, they can not thoroughly distinguish words and word pieces. If we can develop a classifier to fulfill this task, that is, providing a higher confidence value to words than to word piece, the performance of the segmentation could be further improved.

The contribution of the work also comes from the proposal of a feedback-based segmentation approach. This approach adopts a two-step segmentation strategy. Segmentation-by-recognition method and traditional structural-based segmentation method are integrated in our approach by a feedback mechanism. This structure provides an effective and efficient solution to the under- and over-segmentation dilemmas in the legal amount segmentation and other applications. This approach has been tested on a CENPARMI database with 680 legal amount image samples. It has increased the correct segmentation rate by 15.0% from the previous approach in the CENPARMI Cheque Processing System.

7.2 Strengths and Weaknesses of the Approach

The development of the proposed approach is aimed to improve the performance of the segmentation module in the CENPARMI Cheque Processing System. During the development, we effectively used the resources currently available in CENPARMI. We based our work on the existing segmentation and recognition modules. In this respect, some of the strengths of the proposed approach are:

- The system is modular and flexible. With the control of the feedback system, we can either apply or not apply the classifiers in the system. We can apply different classifiers in the segmentation module and select one with the best performance. We can also try various combinations of different segmentation approaches and classifiers.

- The proposed approach can effectively reduce the under- and over-segmentation errors. In preprocessing, the over-segmentation rate is controlled to stay at a very low level. Then, in the re-segmentation and feedback steps, the under-segmented word candidates are split and combined. Thus, the errors from segmentation are greatly reduced in the system.

- The time cost is low. First, the approach generates a final segmentation result. Thus the recognition module needs not to recognize several hypothetical segmentation paths as applied in some other systems. Second, only the suspected under-segmented word candidates from pre-segmentation have to be split and recognized, the time spent on recognition is greatly reduced comparing with the approaches that incorporate classifiers from the beginning of the segmentation.

The approach also has some weakness:

- The approach relies heavily on the classifier. Since the confidence value generated by the classifier is the key for the feedback system to produce the best segmentation path, the performance of the classifier greatly affects the segmentation. The attribute of the classifier also determines the attribute of the segmentation. For example, all the classifiers we used in the research are English word classifiers, so our segmentation approach can not separate French legal amount while the previous approach does not have this restriction.

- The approach can not process the legal amount samples with words that are physically connected. If lines or other symbols are connected to the words, the segmentation module can not split them either. However, such cases are common in the legal amount images.

- Although the approach can separate the digit cent part and some symbols such as "&" from the words, it can not recognize and remove them. These parts are considered as a word or word part and sent to recognition module.

7.3 Future Work

Segmentation of legal amount into words plays an important role in the legal amount processing system. So far an effective and efficient feedback-based segmentation approach has been proposed, we need to further enhance the study of the segmentation module in order to improve the system performance. Following are some topics for future work:

- Further study the relationship between the segmentation and the recognition. Introduce more classifiers into the segmentation process, compare their performances, and find out what kind of classifier is more suitable for our segmentation approach.

- Combination of classifiers could be introduced into the segmentation module to select the best hypothesis segmentation path.

- Develop new algorithms to select the combination of word pieces after the re-segmentation steps. Compare their performances with the proposed approach.

- Try to find out other features from the legal amount image and use them in the segmentation and the feedback process.

Bibliography

[1] C. J. C. Burges, J. I. Ben, J. S. Denker, Y. Lecun, and C. R. Nohl. Off line recognition of handwritten postal words using neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):689–704, August 1993.

[2] R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):690–706, July 1996.

[3] G. Dzuba, A. Filatov, D. Gershuny, I. Kil, and V. Nikitin. Check amount recognition based on the cross validation of courtesy and legal amount fields. In S. Impedovo, P. S. P. Wang, and H. Bunke, editors, *Automatic Bankcheck Processing*, pages 177–193. World Scientific, Singapore, 1997. Series in Machine Perception Artificial Intelligence, Volume 28.

[4] D. Guillevic. *Unconstrained handwriting recognition applied to the processing of bank cheques*. PhD thesis, Dept. of Computer Science, Concordia University, Montreal, Canada, 1995.

[5] D. Guillevic and C. Y. Suen. Cursive script recognition applied to the processing of bank cheques. In *Proc. of the Third International Conference on Document Analysis and Recognition*, pages 11–14, Montreal, Canada, August 1995.

[6] D. Guillevic and C. Y. Suen. Recognition of legal amounts on bank cheques. *Pattern Analysis and Applications*, 1(1):28–41, 1998.

[7] K. M. Hussein. A. Agarwal. A. Gupta, and P. S. P. Wang. A knowledge-based segmentation algorithm for enhanced recognition of handwritten courtesy amounts. *Pattern Recognition*, 32(2):305–316, February 1999.

[8] G. Kaufmann and H. Bunke. Automated reading of cheque amounts. *Pattern Analysis and Applications*, 3(2):132–141, 2000.

[9] G. Kim and V. Govindaraju. Handwritten phrase recognition as applied to street name images. *Pattern Recognition*, 31(1):41–51, January 1998.

[10] G. Kim, V. Govindaraju, and S. N. Srihari. An architecture for handwritten text recognition systems. *International Journal on Document Analysis and Recognition*, 2(1):37–44, July 1999.

[11] G. Kim and Venu Govindaraju. Bankcheck recognition using cross validation between legal and courtesy amounts. In S. Impedovo, P. S. P. Wang, and H. Bunke, editors, *Automatic Bankcheck Processing*, pages 43–85. World Scientific, Singapore, 1997. Series in Machine Perception Artificial Intelligence, Volume 28.

[12] J. H. Kim, K. K. Kim, and C. Y. Suen. An hmm-mlp hybrid model for cursive script recognition. *Pattern Analysis and Application*, 3(4):314–324, 2000.

[13] J. H. Kim, K. K. Kim, and C. Y. Suen. Hybrid schemes of homogeneous and heterogeneous classifiers for cursive word recognition. In *Proc. of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 433–442, Amsterdam, The Netherlands, September 2000.

[14] F. Kimura, M. Shridhar, and N. Narasimhamurthi. Lexicon directed segmentation-recognition procedure for unconstrained handwritten words. In *Proc. of the Third International Workshop on Frontiers in Handwriting Recognition*, pages 122–131, Buffalo, NY, May 1993.

[15] S. Knerr, V. Anisimov, O. Baret, N. Gorski, D. Price, and J. C. Simon. The a2ia intercheque system: courtesy amount and legal amount recognition for french checks. In S. Impedovo, P. S. P. Wang, and H. Bunke, editors, *Automatic*

Bankcheck Processing. pages 43–85. World Scientific. Singapore. 1997. Series in Machine Perception Artificial Intelligence, Volume 28.

[16] L. Lam. C. Y. Suen. D. Guillevic. N. W. Strathy. M. Cheriet. K. Liu. and J. N. Said. Automatic processing of information on cheques. In *Proc. of the International Conference on Systems, Man and Cybernetics*. volume 3. pages 2353–2358. Vancouver. Canada. October 1995.

[17] M. Leroux. E. Lethelier. M. Gilloux. and B. Lemarie. Automatic reading of handwritten amounts on french checks. In S. Impedovo. P. S. P. Wang. and H. Bunke. editors. *Automatic Bankcheck Processing*. pages 157–176. World Scientific. Singapore. 1997. Series in Machine Perception Artificial Intelligence. Volume 28.

[18] K. Liu. C. Y. Suen. and C. Nadal. Automatic extraction of items from cheque images for payment recognition. In *Proc. of the International Conference on Pattern Recognition*. pages 798–802. Vienna. Austria. August 1996.

[19] K. Liu. S. Y. Suen. M. Cheriet. J. N. Said. C. Nadal. and Y. Y. Tang. Automatic extraction of baselines and data from check images. *International Journal of Pattern Recogniton and Artificial Intelligence*. 11(4):675–697. 1997.

[20] S. Madhvanath. E. Kleinberg. and V. Govindaraju. Holistic verification of handwritten phrases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 21(12):1344–1356. December 1999.

[21] U. Mahadevan and R. C. Nagabushnam. Gap matrics for word separation in handwritten lines. In *Proc. of the Third International Conference on Document Analysis and Recognition*. pages 124–127. Montreal. Canada. August 1995.

[22] U. Mahadevan and S. N. Srihari. Hypotheses generation for word separation in handwritten lines. In *Proc. of the Fifth International Workshop on Frontiers in Handwriting Recognition*. pages 453–456. Colchester. England. September 1996.

[23] U. V. Marti and H. Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):65–90, February 2001.

[24] J. Park, V. Govindaraju, and S. N. Srihari. Efficient word segmentation driven by unconstrained handwritten phrase recognition. In *Proc. of the Fifth International Conference on Document Analysis and Recognition*, pages 605–608, Bangalore, India, September 1999.

[25] R. Plamondon and S. N. Srihari. On-line and off-line handwritting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, January 2000.

[26] G. Saon and A. Belaid. High performance unconstrained word recognition system combining hmms and markov random fields. In S. Impedovo, P. S. P. Wang, and H. Bunke, editors, *Automatic Bankcheck Processing*, pages 309–326. World Scientific, Singapore, 1997. Series in Machine Perception Artificial Intelligence, Volume 28.

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

[28] A. W. Senior and A. J. Robinson. An off-line cursive handwriting recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309–321, March 1998.

[29] N. W. Strathy. Handwriting recognition for cheque processing. In *Proc. of the Second International Conference on Multimodal Interface*, volume 3, pages 47–50, Hong Kong, China, January 1999.

[30] C. Y. Suen, J. Kim, K. Kim, Q. Xu, and L. Lam. Handwriting recognition - the last frontiers. In *Proc. of the Fifteenth International Conference on Pattern Recognition*, pages 1–10, Barcelona, Spain, September 2000.

[31] C. Y. Suen, L. Lam, D. Guillivic, N. W. Strathy, M. Cheriet, J. N. Said, and R. Fan. Bank check processing system. *International Journal of Imaging Systems and Technology*, 7(4):392–403, 1996.

[32] C. Y. Suen, Q. Xu, and L. Lam. Automatic recognition of handwritten data on cheques - fact or fiction? *Pattern Recognition Letters*, 20(11-13):1287–1295, 1999.

[33] C. Sun and D. Si. Skew and slant correction for document images using gradient direction. In *Proc. of the Fourth International Conference on Document Analysis and Recognition*, pages 142–146, Ulm, Germany, August 1997.

[34] C. C. Tappert and C. Y. Suen. The state of the art in on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):787–808, August 1990.

[35] B. Yanikoglu and P. A. Sandon. Segmentation of off-line cursive handwriting using linear programming. *Pattern Recognition*, 31(12):1825–1833, December 1998.

[36] X. Ye, M. Cheriet, C. Y. Suen, and K. Liu. Extraction of bankcheck items by mathematical morphology. *International Journal on Document Analysis and Recognition*, 12(8):787–808, August 1999.

[37] J. Zhou, A. Krzyzak, and C. Y. Suen. Recognition and verification of touching handwritten numerals. In *Proc. of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 179–188, Amsterdam, Netherlands, September 2000.

[38] J. Zhou, C. Y. Suen, and K. Liu. A feedback-based approach for segmenting handwritten legal amounts on bank cheques. In *Proc. of the Sixth International Conference on Document Analysis and Recognition*, Seattle, WA, September 2001.

[39] Jun Zhou. A feedback-based approach for segmentation of legal amount on bank cheques. Technical report. CENPARMI, Concordia University, November 2000.

[40] Jun Zhou. Segmentation and recognition of bank cheque legal amounts. Technical report, CENPARMI, Concordia University, July 2001.