Spiral Recognition Methodology and Its Application for Recognition of Chinese Bank Checks

Hanshen Tang 1, Emmanuel Augustin 2, Ching Y. Suen 1, Olivier Baret 2, Mohamed Cheriet 3

1 Centre for Pattern Recognition and Machine Intelligence, Concordia University,

1455 de Maisonneuve Blvd. West,

Montréal, Québec H3G 1M8, Canada

{hstang, suen}@cs.concordia.ca

2 Artificial Intelligence and Image Analysis (A2iA),

40 bis, rue Fabert,

75007 Paris, France

{ea, ob}@a2ia.com

3 École de Technologie Supérieure,

1100 rue Notre-dame Ouest,

Montréal, Québec H3C 1K3, Canada

Mohamed.Cheriet@etsmtl.ca

Abstract

.

This paper presents the spiral recognition methodology with its application in unconstrained handwritten Chinese legal amount recognition in a practical environment of a CheckReader™. This paper first describes the failed application of neural network - hidden Markov model hybrid recognizer on Chinese bank check legal amount recognition, and explains the reasons for the failure: the neural network - hidden Markov model hybrid recognizer can not handle the complexity in the training for Chinese legal amounts. Then a spiral recognition methodology is presented. This methodology enables the system to increase its recognition power (both the recognition rate and the number of recognized characters) during the training iterations. Some experiments were done to show that the spiral recognition methodology has a high performance in the recognition of unconstrained handwritten Chinese legal amounts. The recognition rate at the character level is 93.5%, and the recognition rate at the legal amount level is 60%.Combined with the recognition of courtesy amount, the overall error rate is less than 1%.

Keywords: spiral recognition methodology, offline handwriting recognition, Asian character recognition

1. Introduction

Recognition of unconstrained handwritten Chinese characters has been a challenging topic for a while. Automatic processing of legal amounts handwritten on Chinese bank checks is one of the most practical and promising applications. A lot of research has been done [1 - 3]. Because of the large size of the character set, variety in fonts and writing styles, the complex structures of characters, and the ambiguity in grammar, the problem remains very difficult. There is not yet a viable recognition product in the industry for the recognition of unconstrained handwritten Chinese script.

Neural network - hidden Markov model (NN-HMM) hybrid recognizer has a good performance in the recognition of legal amounts handwritten in Latin languages. The hybrid recognizer can handle the uncertainty in segmentation of the handwritten cursive script very well. A2iA CheckReader™ is based on it, and makes very good result [4].

However, the application of NN-HMM hybrid recognizer in the recognition of legal amounts handwritten in Chinese was not successful, due to the complexity of the training for unconstrained handwritten Chinese characters. More details are presented in Section 2.

In order to solve the problem, we introduced a spiral recognition methodology, as illustrated in Figure 1. In Phase 1, the system generates annotations of character samples. In Phase 2, the annotation results are analyzed (semi-automatically) and manually corrected (if necessary). In Phase 3, a neural network character recognizer is trained with the annotation results. Finally in Phase 4, the trained neural network character recognizer processes the input data, and then passes the results to a hidden Markov model legal amount recognizer for the final recognition results. As the process iterates, the number of recognized classes is increasing, while the number of annotation character samples is also increasing. Details and its application are presented in Section 3.
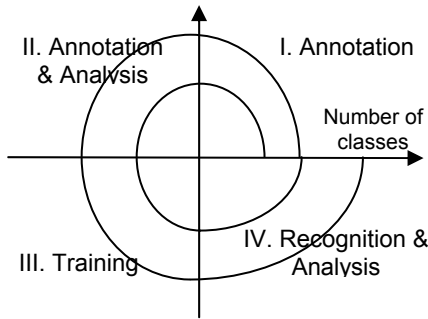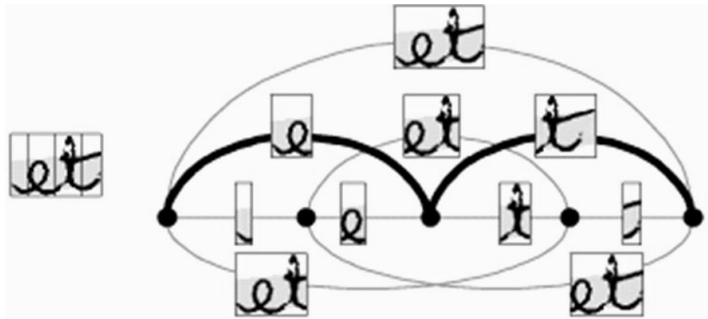
Fig. 1: Spiral recognition methodology.

Fig. 2: HMM processing of the French word "et". [5]

2. Neural Network – Hidden Markov Model Hybrid Recognizer

A neural network character recognizer handles well the variety of shapes, fonts, and handwriting styles, as long as enough data is fed to the network during the training phase. Hidden Markov model word recognizer can handle the uncertainty in segmentation of the cursive handwritten script. Figure 2 illustrates how the HMM word recognizer processes the French word "et".
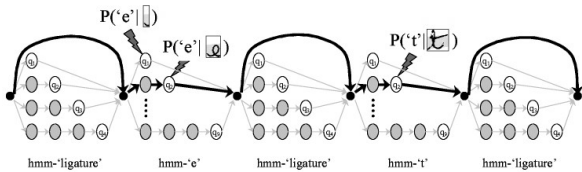

Fig. 3: NN-HMM hybrid recognizer. [5]

Thus NN-HMM hybrid recognizer has a good performance in the recognition of legal amounts handwritten in Latin languages. Figure 3 illustrates one such system: the NN-HMM hybrid recognizer used in A2iA Check-Reader™ [4]. Figure 4 shows virtually the idea of the training and recognition of the system, although the system is not actually implemented in the exact way. The neural network recognizer shown here is a probabilistic neural network recognizer. It produces the character recognition results as character candidate lists with confidence values. The hidden Markov model recognizer takes the candidate lists and produces the word recognition results as word candidate lists with confidence values. In the training phase, the true sequences of words are passed to the forward-backward process of the HMM word recognizer. Then, the expected character results can be computed. They are then used to conduct the back propagation of the NN character recognizer. Eventually, the system can achieve recognition rates varying from 65% to 85% (depending on countries and quality of the bank check images) with the error rates as low as 0.1% [4].
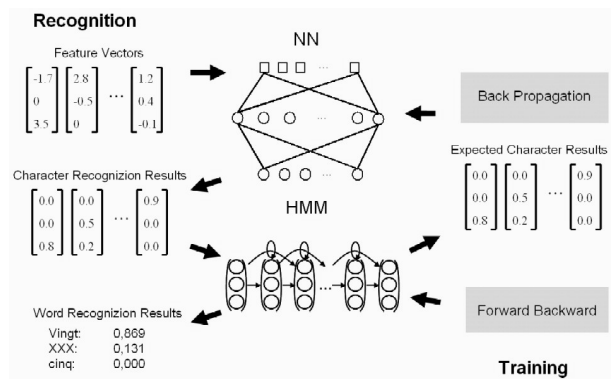

Fig. 4: Training and testing of a NN-HMM hybrid recognizer.

Table 1: Chinese character set for legal amounts.

Digit or Amount	0	1	2	3	4	5	6
Simplified Chinese	0 零	一	二 两	三	四	五	六
Traditional Chinese	零	壹	貳 兩	叁	肆	伍	陸 陸
Commonly Accepted Synonym	另		貳 貳 式 弍	弍	㙺四		

Digit or Amount	7	8	9	10	100	1k	10k
Simplified Chinese	七	八	九	十	百	千	万
Traditional Chinese	柒	捌	玖	拾	佰	仟	萬
Commonly Accepted Synonym			玖				

Digit or Amount	100,000k	Dollar	10 Cents	1 Cent	Only
Simplified Chinese	亿	元	毛 角 毫	分 仙	正 整
Traditional Chinese	億	圓			

However, when we tried to build an application of the NN-HMM hybrid recognizer on Chinese characters, the experimental results were not satisfactory. The recognition rate at the legal amount level was around 27%. The training of the recognizer did not converge. We analyzed the problem, and found the following reasons:

1) Chinese characters have more complex structures. Some different versions of characters have the same meanings. Many characters can be further split into radicals, while there are some common radicals in different characters. Table 1 shows the character set for Chinese legal amounts. Figure 5 illustrates that some of the characters can be further split into radicals. Moreover, some radicals themselves are characters. These make accurate segmentation at character level very difficult, and thus significantly increases the complexity of the candidate lists that the HMM recognizer need to process.

Fig. 5 Chinese radicals that form characters.
The characters are on the left of the arrows. The radicals are on the right of the arrows. Some of the radicals themselves are also characters.

Figure 6 shows a graph of segmentation paths for a Chinese legal amount. The complexity of the graph is relatively low. One can easily find many more complicated cases.
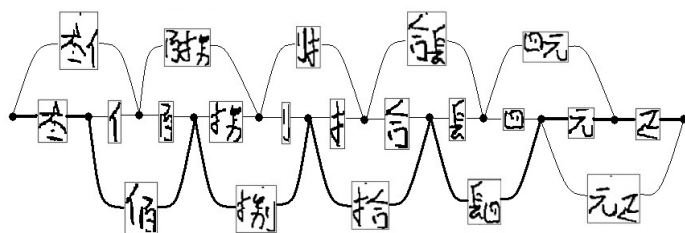

Fig. 6: Graph of segmentation paths for a Chinese legal amount.
The correct segmentation path is bolded.

2) For legal amounts, each Chinese character itself can form a word, which makes it difficult to build a word dictionary. After character recognition, the system directly proceeds to amount level recognition with the HMM recognizer. Usually, a legal amount consists of a long character sequence.

3) The grammar for Chinese legal amounts is ambiguous. For a given numeric amount, there can be more than tens of possible character sequences in Chinese legal amount. This makes the forward-backward process during the training phase of the HMM recognizer very difficult to stabilize.

4) The NN-HMM hybrid recognizer is trained based on the maximum mutual information estimation. The following discriminant training criterion is applied:

$$\log R(\Lambda) = \sum_r \log P(O^r|\lambda_r) - \sum_w \log P(O^r|m_w)P(m_w)$$

where $P(O^r|\lambda_r)$ is the probability of a true path, and $P(O^r|m_w)P(m_w)$ is the probability of all the other possible paths. The training process intends to maximize $\log R(\Lambda)$. However, if there are too many possible paths, $\sum_w \log P(O^r|m_w)P(m_w)$ becomes excessively large, and eventually stops proper training. Unfortunately, this is exactly the case we had.

Since the NN-HMM hybrid recognizer could not be trained properly, we introduced the spiral recognition methodology.

3. Spiral Recognition Methodology

The idea of the spiral recognition is incremental: train and develop the system step by step. As shown in Figure 1, the spiral recognition methodology has 4 phases: annotation, annotation analysis, training, and recognition. Each phase is dependent on its previous phase. Kornai et al. introduced an iterative method to reduce human involvement to prepare training data [6]. They first manually built a bootstrap set, and then tried to enlarge the set with some techniques. Our methodology is more robust and more automatic. Grammar and contextual information is used to aid the automatic process. Although some manual analysis during the process may be required, the human involvement is further reduced.

We built a grammar-contextual tool for Chinese legal amount. The grammar tool can accomplish two tasks: One is to check whether a sequence of hypothetical character / radical observations is a valid legal amount; if yes, the equivalent numeric amount is returned. The other one is to generate all possible sequences of characters / radicals with a given numeric amount. During the recognition phase, the tool can help the recognizer to bypass some of the syntactically or contextually impossible sequences of hypothetical character / radical observations, which result from over-segmentation and under-segmentation, etc. It may also be used to combine redundant sequences of hypothetical character / radical observations due to the presentations of radicals (refer to Figure 5). During the training phase, the tool can help to direct the recognition system to correct and stable states. It may also be used to extend the character level recognizer to a radical level recognizer (refer to Figure 5).

3.1 Initial iteration

In the first iteration of the process, the annotation tool has not much prior knowledge. With the segmentation information (including segmentation points, estimated character width, and inter-character spaces, etc.) and grammar and contextual information (possible character sequences generated for a given numeric amount of the bank check), the automatic annotation tool can produce annotated character samples for neat and simple bank checks. All the checks with noises and uncertainty in segmentation are rejected. Figure 7 shows a snapshot of the annotation result.
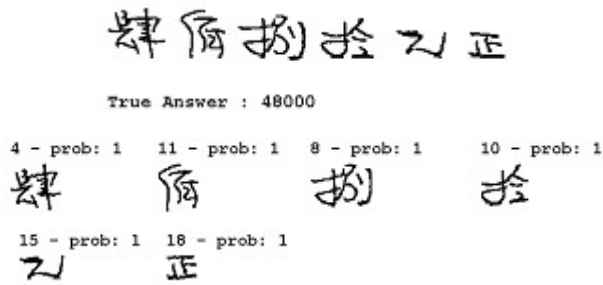

Fig. 7: Annotation result.
The upper part is the image of the legal amount on a neat bank check. The lower part is the annotated character samples.

Then an annotation analysis tool is applied, as shown in Figure 8. In this stage, the annotation accuracy is not very high. But the qualities of the annotations for some key characters, such as "dollar", "only", "ten", and "hundred" are high enough to train the neural network character recognizer. The average annotation error rate for these characters is around 4%. If the annotation quality is not good enough, manual correction may be necessary.

After annotation analysis and manual correction (if necessary), the annotated character samples are fed to the neural network character recognizer. Then the recognition phase is the same as the normal NN-HMM hybrid recognizer.
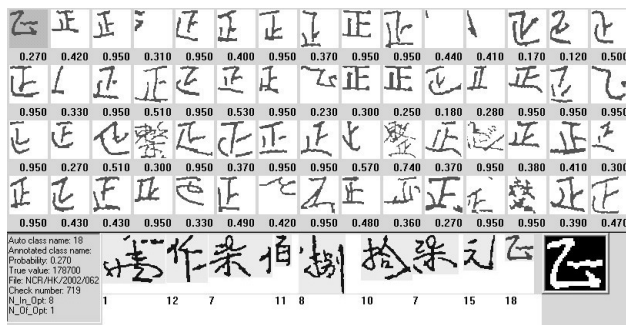

Fig. 8: Annotation analysis tool.
The upper part of the window shows the annotated characters. The lower part of the window shows from which original legal amount a selected character is.

3.2 Following Iterations

In the second and following iterations of the process, annotation has gathered more prior knowledge. With the segmentation information, grammar and contextual information, and recognition information, the automatic annotation tool can produce more annotated character samples for more bank checks. Figure 9 shows a segmentation-grammar graph for the legal amount on a bank check. With the graph, annotated character samples can be generated of better quality.

The annotation analysis, training, and recognition phase is similar to those of the initial iteration. Analysis shows that the number of annotated samples, the quality of the annotation, and the recognition power are all increasing during the iterations. Figures 10 and 11 show the recognition result of a bank check handwritten in Chinese.
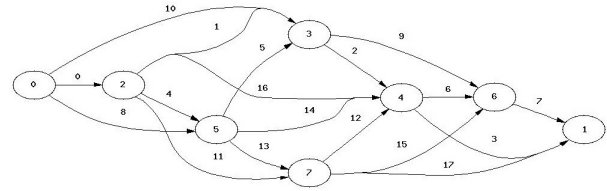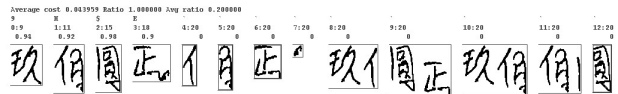

Fig. 9: Segmentation-grammar graph for a legal amount image.
The numbers in the vertices are transition states. The edges are labeled with the number associated with the character object indices. This directed graph show all the possible segmentation for a give legal amount.

Fig. 10: Input bank check image.

```
Literal amount candidates:              Cent part candidates:

 0 :     2214.00 ----> 0.00207      0 :      0 ----> 0.00367
 1 :     7214.00 ----> 0.00138      1 :      1 ----> 0.00000
 2 :     2714.00 ----> 0.00006      2 :      2 ----> 0.00000
 3 :     3214.00 ----> 0.00005      3 :      3 ----> 0.00000
 4 :     7714.00 ----> 0.00004      4 :      4 ----> 0.00000
 5 : 20000014.00 ----> 0.00002      5 :      5 ----> 0.00000
 6 :     2414.00 ----> 0.00001      6 :      6 ----> 0.00000
 7 :     7314.00 ----> 0.00001      7 :      7 ----> 0.00000
 8 :     9214.00 ----> 0.00001      8 :      8 ----> 0.00000
 9 :     2814.00 ----> 0.00001      9 :      9 ----> 0.00000
10 :     2314.00 ----> 0.00001     10 :     10 ----> 0.00000
11 :     7014.00 ----> 0.00000     11 :     11 ----> 0.00000
```

Fig. 11: Recognition result for the image in Figure 10.
The recognition result option 0 is "2214.00", which is the correct numeric value.

4. Experimental results and analysis

After 8 iterations of the process on the database of a training set of 47.8 thousand real bank checks, and a test set of 12 thousand, the average annotation error rate reduces to 3%. The recognition rate for annotated samples is 90.66% for the top candidate. The real recognition at the character level is estimated by

$$\text{Real Recognition Rate} = \frac{\text{Annotation Recognition Rate}}{1 - \text{Annotation Error Rate}}.$$

Thus at the character level, the estimated real recognition rate is 93.5% for the top candidate, while the amount level recognition rate is 60% for the top candidate, and 76% for the top 4 candidates. Combined with the recognition of the courtesy amount, the recognition rate is 85%, with an error rate of 1%. Figure 12 shows the character level recognition rate increased from 79.4% to 93.5% during the 8 iterations. Figure 13 shows that the number of annotated character samples increased from 221k to 315k for the training set, and increased from 72k to 79k for the test set. Figures 14 and 15 show the confusion matrices after the second and the eighth iterations, respectively. It's obvious that after several iterations, the confusion among different characters decreased. For key characters, such as "dollar", "only", "ten", and "hundred", the changes were relatively small (average 7% increase). However, for the other characters, the recognition rates increase rapidly (average 15% increase).
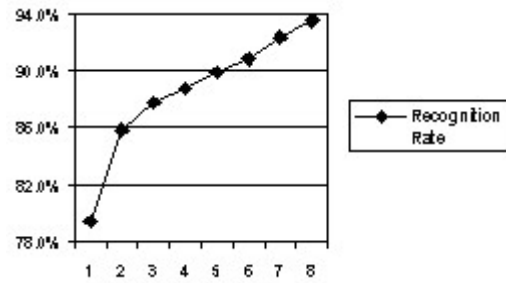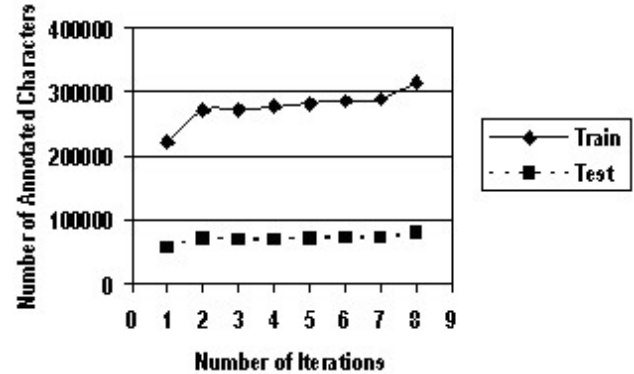

Fig. 12: Recognition rate at character level.

Fig. 13: Number of annotated character samples

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	72.1	**6.2**	0.2	**11.1**	0.4	0.0	1.6	0.2	0.0	0.4	0.2	0.2	0.0	0.7	0.0	0.0	0.0	0.0	1.2	0.0	**5.6**
1	0.2	79.2	0.3	**2.0**	0.1	0.3	0.3	0.1	0.0	0.0	0.2	0.2	0.0	0.7	0.0	0.0	0.0	0.0	0.8	0.0	**15.6**
2	0.3	0.4	85.9	0.5	0.0	0.1	0.0	0.4	0.0	0.4	0.4	0.0	0.0	0.2	0.0	0.6	0.0	0.0	0.2	0.0	**10.4**
3	0.5	1.3	0.3	84.0	0.0	0.2	0.4	0.2	0.0	0.1	0.4	0.1	0.0	0.2	0.0	0.2	0.0	0.0	1.0	0.0	**11.1**
4	0.0	0.1	0.1	0.1	81.8	0.2	0.4	0.1	0.1	0.1	0.2	0.7	0.1	0.8	0.1	0.0	0.1	0.1	0.3	0.0	**14.8**
5	0.0	0.1	0.0	0.0	0.1	85.9	0.5	0.0	0.0	0.0	0.1	**5.7**	0.6	0.0	0.0	0.0	0.0	0.2	0.5	0.0	**6.3**
6	0.1	0.4	0.1	0.1	0.1	1.1	84.5	0.0	0.1	0.4	**2.2**	1.3	0.0	0.1	0.0	0.2	0.0	0.0	0.4	0.0	**9.0**
7	0.0	0.6	0.9	0.4	0.6	0.2	0.1	82.3	0.0	0.9	0.4	0.1	0.2	0.1	0.0	0.3	0.0	0.1	0.6	0.0	**12.2**
8	0.0	0.0	0.0	0.0	0.3	0.0	0.1	0.0	84.7	0.0	1.1	0.4	0.1	0.0	0.0	0.6	0.0	0.1	0.0	0.0	**12.7**
9	0.1	0.2	0.6	0.2	0.2	0.3	0.3	0.1	0.2	89.9	0.6	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.4	0.0	**6.7**
10	0.0	0.0	0.1	0.1	0.0	0.3	0.6	0.0	0.1	0.2	89.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	**8.3**
11	0.0	0.0	0.0	0.0	0.0	1.9	0.1	0.0	0.0	0.0	0.1	89.9	0.2	0.1	0.0	0.0	0.1	0.0	0.0	0.0	**7.4**
12	0.0	0.0	0.1	0.0	0.0	0.4	0.1	0.0	0.0	0.0	0.1	0.5	91.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	**7.6**
13	0.1	0.9	0.9	1.1	0.3	0.0	0.6	0.1	0.0	0.0	0.1	0.6	0.0	81.7	0.0	0.7	0.4	0.0	1.0	0.0	**11.4**
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.0**
15	0.0	0.1	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.1	0.0	0.1	0.0	76.0	0.0	0.0	0.8	0.0	**22.0**
16	0.0	**2.7**	0.0	0.2	0.2	0.0	0.2	0.0	0.3	0.0	0.0	**6.4**	0.8	2.5	0.0	1.3	66.2	0.2	0.3	0.0	**18.7**
17	0.0	0.0	0.9	0.0	0.9	**7.0**	0.0	0.0	0.0	0.0	0.0	3.5	2.6	0.9	0.0	1.7	0.0	32.2	0.0	0.9	**49.6**
18	0.0	0.2	0.1	0.2	0.1	0.1	0.4	0.0	0.0	0.1	0.7	0.0	0.0	0.0	0.0	0.1	0.0	0.0	86.0	0.2	**11.8**
19	0.0	0.5	0.3	0.2	0.2	0.3	0.0	0.0	0.2	0.2	0.3	0.0	0.2	0.2	0.0	0.0	0.0	0.2	2.3	50.5	**44.7**
20	0.0	0.5	0.4	0.5	0.3	0.5	0.3	0.1	0.2	0.2	0.6	0.9	0.7	0.2	0.0	0.6	0.0	0.1	1.0	0.2	92.6

Fig. 14: Confusion matrix after 2nd iteration.
Classes 10, 11, 12, 15, and 18 are "ten", "hundred", "thousand", "dollar" and "only", respectively. Class 14 is "hundred billion", which we have never seen in a real bank check. Consequently, its recognition rate is always 0.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	87.9	3.8	0.2	2.7	0.0	0.0	0.0	0.6	0.0	0.0	0.2	0.0	0.0	0.6	0.0	0.4	0.0	0.0	0.0	0.2	3.5
1	0.4	91.1	0.2	0.8	0.1	0.2	0.1	0.4	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.2	0.0	0.0	0.5	0.1	5.6
2	0.2	0.3	89.8	0.4	0.0	0.0	0.0	1.0	0.1	0.2	0.3	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.2	0.7	5.9
3	1.6	2.5	0.4	84.3	0.1	0.0	0.2	0.4	0.0	0.0	0.4	0.1	0.0	0.1	0.0	0.7	0.0	0.0	0.7	0.4	8.2
4	0.0	0.2	0.0	0.0	90.1	0.2	0.2	0.1	0.1	0.1	0.2	0.2	0.5	0.0	0.0	0.2	0.1	0.0	0.0	0.2	7.5
5	0.0	0.1	0.0	0.1	0.1	88.1	0.5	0.0	0.0	0.0	0.2	5.0	0.2	0.0	0.0	0.1	0.1	0.0	0.1	0.1	5.5
6	0.0	0.1	0.0	0.1	0.3	0.2	90.6	0.0	0.1	0.1	1.1	0.5	0.0	0.1	0.0	0.1	0.1	0.0	0.2	0.0	6.5
7	0.2	0.5	0.9	0.1	0.1	0.0	0.3	91.0	0.0	0.2	0.4	0.0	0.2	0.1	0.0	0.5	0.0	0.0	0.4	0.5	4.7
8	0.0	0.0	0.0	0.0	0.2	0.2	0.1	0.0	90.0	0.2	0.8	0.1	0.0	0.0	0.0	0.4	0.2	0.0	0.0	0.3	7.6
9	0.2	0.0	1.2	0.0	0.3	0.0	0.1	0.1	0.1	90.5	0.9	0.0	0.1	0.0	0.0	0.4	0.0	0.0	0.2	0.1	5.8
10	0.0	0.1	0.1	0.0	0.1	0.0	0.7	0.1	0.1	0.2	93.3	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.3	0.0	4.5
11	0.0	0.0	0.0	0.0	0.0	1.4	0.1	0.0	0.0	0.0	0.1	93.2	0.2	0.0	0.0	0.0	0.1	0.0	0.0	0.0	4.6
12	0.0	0.0	0.0	0.1	0.1	0.2	0.0	0.1	0.0	0.0	0.1	0.1	93.3	0.0	0.0	0.0	0.1	0.0	0.0	0.0	5.8
13	0.2	2.0	0.8	0.0	0.2	0.0	0.2	0.2	0.5	0.0	0.5	0.2	0.0	85.5	0.0	1.1	1.1	0.0	0.5	0.0	7.2
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.2	0.3	0.1	0.1	0.0	0.0	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.0	90.3	0.0	0.0	0.3	0.2	7.9
16	0.0	3.7	0.2	0.4	0.0	0.2	0.0	0.2	0.4	0.0	0.2	2.8	0.7	0.6	0.0	1.5	73.8	0.0	0.4	0.2	14.9
17	0.0	0.0	0.9	0.9	0.0	1.8	0.0	0.0	0.0	0.0	0.2	3.7	1.8	0.0	0.0	1.8	0.0	42.2	0.9	0.0	45.9
18	0.0	0.4	0.0	0.1	0.0	0.0	0.2	0.0	0.0	0.1	0.2	0.0	0.0	0.1	0.0	0.3	0.0	0.0	91.9	0.6	6.0
19	0.0	0.2	0.3	0.1	0.2	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.1	1.6	84.6	12.2
20	0.1	0.7	0.4	0.3	0.3	0.3	0.3	0.2	0.3	0.2	0.6	0.8	0.6	0.1	0.0	1.2	0.2	0.0	0.9	1.6	90.9

Fig. 15: Confusion matrix after 8th iteration.
Classes 10, 11, 12, 15, and 18 are "ten", "hundred", "thousand", "dollar" and "only", respectively.

The reason that the increase of the recognition rates of key characters was not significant is the following: Key characters were better annotated in the initial iteration, and consequently, better trained. However, for the other classes, the power of the spiral recognition methodology is clearly illustrated. Particularly, the recognition rate of Class 19 (for under-segmented observations) increased 34.1%. This means that the system was well trained towards true characters against segmentation errors, and thus the overall legal amount recognition rate was increased.

5. Conclusion

Neural network – hidden Markov model hybrid recognizer is powerful. However, it can not handle excessively complicated cases in the training phase. The spiral recognition methodology is introduced to solve the problem. More contextual information can be extracted from the bank check images. The NN-HMM hybrid recognizer is trained step-by-step, and eventually it achieves a better recognition performance. The recognition rate is 93.5% at the character level, and 60% at the amount level. The automatic annotation strategy looks quite promising, which enables the system to be trained with more samples for each character, as well as with more variety of characters, if a proper grammar and vocabulary are given. The system can be further extended to recognize Chinese radicals (as mentioned in Section 3), which is a promising approach for general-purpose handwritten Chinese character recognition.

Acknowledgement

References

[1] J. W. Tai, "A syntactic-semantic approach for describing Chinese characters," Computer Processing of Chinese and Oriental Languages, Vol. 1, No.3, Chinese Language Computer Society, San Francisco, May 1984, pp. 186-194

[2] H. Chen, L. Gareth, Y. Wu, and R. Zitserman, "Segmentation and recognition of continuous handwriting Chinese text," Advances in Oriental Document Analysis and Recognition Techniques, World Scientific, Singapore, 1998, pp. 223-232

[3] M. L. Yu, P. C. K. Kwok, C. H. Leung, and K. W. Tse, "Recognition of Chinese bank cheque amounts," International Journal on Document Analysis and Recognition, Vol. 3, Issue 4, Springer, Berlin, 2001, pp. 207-217

[4] N. Gorski, V. Anisimov, E, Augustin, O. Baret, and S. Maximov, "Industrial bank check processing: the A2iA CheckReader™," International Journal on Document Analysis and Recognition, Vol. 3, Springer, Berlin, 2001, pp. 196-206

[5] Y. H. Tay, P.-M. Lallican, M. Khalid, S, Knerr, and C. Viard-Gaudin, "An Analytical Handwritten Word Recognition System with Word-level Discriminant Training," Proceeding of ICDAR 2001, Seattle, 2001, pp726-730

[6] A. Kornai, K. Mohiuddin, and S. Connell, "Recognition of cursive writing on personal checks," Proceeding of the Fifth International Workshop on Frontiers in Handwriting Recognition (IWFHR-5), Colchester, 1996, pp 373-378