Algorithms for Off-line Recognition of Chinese Characters

Manling Ren

26th June 1998

A thesis submitted in partial fulfilment of the requirements of The Nottingham Trent University for the degree of Doctor of Philosophy

Department of Computing, The Nottingham Trent University
Burton Street, Nottingham NG1 4BU, United Kingdom

Algorithms for Off-line Recognition of Chinese Characters

Manling Ren

26th June 1998

中国十大名曲

《高山流水》
传说先秦的琴师伯牙一次在荒山野地弹琴，樵夫钟子期竟能领会这是描绘"巍巍乎志在高山"和"洋洋乎志在流水"。伯牙惊道："善哉，子之心而与吾心同。"子期死后，伯牙痛失知音，摔琴绝弦，终身不操，故有高山流水之曲。

《广陵散琴曲》
[illegible]临刑前曾从容弹奏此曲以寄托。现存琴谱最早见于《神奇秘谱》。

《平沙落雁》
明朝此曲称《雁落平沙》。曲调悠扬流畅，通过时隐时现的雁鸣，描写雁群降落前在空际盘旋顾盼的情景。

《梅花三弄》
此曲系借物咏怀，通过梅花的洁白、芬芳和耐寒等特征，来赞颂具有高尚节操的人，曲中泛音曲调在不同的徽位上重复了三[illegible]"三弄"。

《十面埋伏》
乐曲描写公元前二零二年楚汉战争在垓下最后决战的情景。汉军用十面埋伏的阵法击败楚军。项羽自刎于乌江，刘邦取得胜利。

《夕阳箫鼓》
这是一首抒情写意的文曲，旋律优美流畅。一九二五年前后，上海大同乐社根据此曲改编成丝竹乐曲《春江花月夜》。

《渔樵问答》
此曲在历代传谱中，有三十多种版本，有的还附歌词。乐曲表现渔樵在青山绿水中间自得其乐的情趣。

《胡笳十八拍》
根据同名诗谱写的乐曲。所反映的主题是"文姬归汉"。汉末战乱中，蔡文姬流落到南匈奴达十二年之久，她身为左贤王妻，然而十分思念故乡。当曹操派人接她回内地时，她又舍不得离开两个孩子，还乡喜悦被骨肉离别之痛所淹没，心情非常矛盾。

《汉宫秋月》
此曲有两种较为流行的演奏形式，一为筝曲，另为二胡曲。据载，二胡是刘天华先生所传。《汉宫秋月》意在表现古代受压迫宫女的幽怨悲泣情绪，唤起人们对她们不幸遭遇的同情。

《阳春白雪》
相传这是春秋时期晋国的师旷或齐国的刘涓子所作。现存琴谱中的《阳春》和《白雪》是两首器乐曲，《阳春》取万物知春，和风淡荡之意，《白雪》取凛然清洁，雪竹琳琅之意。

Copyright

Abstract

Computer recognition of Chinese characters is a challenging topic and important research area. It is relevant to documentation, publications, language translation, handwriting of Chinese and Japanese '*Kanji*' in industry, business, diplomacy and daily life. Typical development of the recognition process focuses on printed, on-line and off-line hand-written characters using techniques including a two-layer hierarchy, four-corner, radical and a whole character recognition. Although existing recognition methods have achieved some success, the lack of fundamental algorithms for representing the structure of Chinese characters has prevented the recognition of characters within large vocabulary and having a complicated topological structure embedded within the 2-D pictorial format. The current project develops a new structural representation to remedy the lack of an effective recognition process of such characters. The research also investigates methods of dealing with variable size, position, shape, vagueness and ambiguity of a character. A key input character method using manual operation, called the '*Cang-Jie*' method, is applied as an effective tool for verification of a Chinese character.

A novel method is developed to represent the structure of Chinese characters: a three-layer hierarchy of character-radical-stroke and its process: character-radical-code, which is specially suited for 2-D objects with topological features. The character is deconstructed into radicals according to their shape, position and extraction order. Radicals are classified into 26 categories in terms of their shape structure and meanings. Recognition of a radical yields the code of the category to which it belongs. The chain code method is applied to restructure these category codes into a 1-D chain code. The chain code is verified by matching it to a code database. To further enhance the method, a fuzzy neural network system has been designed and implemented to recognise characters in printed and standard writing, using uncertainty and topology analysis, fuzzy possibilistic reasoning, neocognitron and associative memory neural networks, chain code method and error probability method. A software system has been written using the C programming language and X View function. Test results of the system have been obtained. Improvement of the system to deal with vagueness and ambiguity (two separate characteristics) during recognition has been carried out at several stages and the recognition rate has been increased to 96%.

The main achievements include the structural representation of Chinese characters, extraction of radicals, recognition and verification of characters, and simplifying the recognition process.

Key words: Analysis of error probability, associative memory neural network, chain code method, fuzzy possibilistic inference rules, recognition of radicals, restructuring of chain codes, three-layer hierarchy of Chinese characters, 2-D topological structure.

Acknowledgements

I would like to express my sincere appreciation to my first supervisor, Dr. David Al-Dabass, in the Department of Computing, for his supervision and support in making this project possible, and to my second supervisor, Dr. Daizhong Su, in the Department of Mechanical and Manufacturing Engineering, for his valuable suggestions and constant help.

I would also like to thank my father, Wangbao Ren, for his understanding and encouragement of my professional ambition, and for prompting me to change my career to academic research.

Finally, my honest thanks have to go to all my friends and colleagues in UK, China, Taiwan, Hong Kong, Japan, Thailand, USA, and Australia for their suggestions, help, and supports of my research and personal life during the past four years.

The following sections of this thesis have been previously published. Details of the publications are in Appendix D: Publications Related from This Research.

.3.4, 4.5 and 5.5 (published as [Ren98b]);
.3.2.3, 5.3 and 5.4 (published as [Ren98a]);
.3.2 and 3.3.4 (published as [Ren97b]);
.3.3.4 and 3.4.2 (published as [Ren97a]);
.3.2.3, 4.3.2 and 4.3.3 (published as [Ren96b]);
.3.2.1 and 3.3.4 (published as [Ren96a]);
.3.2.3 and 4.2.1 (published as [Ald95]);
.3.4.2 and 4.5.4 (published as [Ren95a]).

The thesis is dedicated to my mother, Xiulian Fan, in the memory of her love.

Contents

Abstract i

Acknowledgements ii

Glossary of Terms xii

Glossary of Symbols xiv

Chapter 1. Introduction 1
- 1.1 Chinese Characters 1
- 1.2 Aims and Objectives of the Project 5
- 1.3 Outline of the Thesis 6

Chapter 2. Limitations of Existing Methods 9
- 2.1 Introduction 9
- 2.2 Difficulties and Methods in Recognising Alphabetical Characters 11
  - 2.2.1 Difficulties 11
  - 2.2.2 Methods and Techniques 12
- 2.3 Chinese Characters Recognition 15
  - 2.3.1 Characteristics of the Structure 16
  - 2.3.2 Methods and Techniques 18
  - 2.3.3 Chinese Character Database 24
- 2.4 Comments 26

Chapter 3. A Three-layer Hierarchy Method for the Recognition of Chinese Characters 27
- 3.1 Introduction 27
- 3.2 Structural Representation of Chinese Characters 29
  - 3.2.1 Five-Level Structure 29
  - 3.2.2 *Cang-Jie* Method 31
  - 3.2.3 Three-Layer Hierarchy 33
- 3.3 Knowledge Representation Techniques 35
  - 3.3.1 Statistical Method 35
  - 3.3.2 Syntactic Method 36
  - 3.3.3 Fuzzy Logic 36
  - 3.3.4 Fuzzy Syntactic Method 37
  - 3.3.5 Fuzzy Possibilistic Reasoning 40
- 3.4 Artificial Neural Networks 41
  - 3.4.1 Neocognitron Neural Network 41
  - 3.4.2 Associative Memory Neural Network 45

3.5 Chain Code Method 47
3.6 Analysis of Error Probability 48

Chapter 4. Software Development of Chinese Character Recognition System (CCRS) 50
4.1 Introduction 50
4.2 Analysis, Design, and Implementation of CCRS 51
4.2.1 System Analysis 51
4.2.2 Design Method 55
4.2.3 Implementation 60
4.3 Preprocessing 61
4.3.1 Vagueness of Radicals 62
4.3.2 Situation 63
4.3.3 Shape 65
4.4 Normalisation 67
4.4.1 Variable Radicals 67
4.4.2 Normalisation Subsystem with Neocognitron Structure 69
4.5 Recognition 70
4.5.1 Discussion of the *Cang-Jie* Method 70
4.5.2 Development of Classification of Radicals 71
4.5.3 Architecture of the Associative Memory Neural Network 71
4.5.4 Modification of the Network 73
4.6 Restructuring 74
4.6.1 Restructuring Cues 74
4.6.2 Code Database 77
4.7 Postprocessing 78
4.7.1 Assessment Scheme 78
4.7.2 Error Detection 79
4.7.3 Error Correction 80
4.7.4 Rejection 81

Chapter 5. Results and Discussions 82
5.1 Introduction 82
5.2 Performance of CCRS 83
5.2.1 Structure of the Main System and Subsystems 83
5.2.2 Performance Data 84
5.2.3 Time Utilisation 85
5.2.4 Discussion on Performance Time 87
5.3 Extraction of Radicals 87
5.3.1 Different Positions 88
5.3.2 Different Shapes 88
5.3.3 Different Orders 89
5.3.4 Expansibility of Rules 89
5.3.5 Special Cases of Extraction 89
5.4 Radical Normalisation 90

5.4.1 Position Adjustment 91
5.4.2 Shape Selection 91
5.4.3 Discussion on Fixed Patterns 92
5.5 Classification and Recognition of Radicals 92
5.5.1 Classification of Radicals 93
5.5.2 Learning Phase 93
5.5.3 Training Phase 96
5.5.4 Modification of the Network 97
5.5.5 Discussion on Recognition 99
5.6 Restructuring of Chain Codes 100
5.6.1 Application of Restructuring Cues 101
5.6.2 Verification 102
5.6.3 Discussion on Restructuring 103
5.7 Error Assessment 103
5.7.1 Global Assessment 104
5.7.2 Assessment Action 105
5.7.3 Discussion on Correction 106

Chapter 6. Conclusions and Future Work 107
6.1 Conclusions 107
6.2 Future Work 109

References Ref-1

Appendix A. Support Tools A-1

Appendix B. Developed Software B-1
B.1 Programs B-1
B.2 Files B-6
B.3 Code Database B-7

Appendix C. Experimental Data and Outcomes C-1
C.1 Data C-1
C.2 Outcomes C-6

Appendix D. Publications Related from This Research D-1

List of Figures

Figure 1.1 Chinese phonetic symbols in two styles 2
Figure 1.2 The Chinese character 'book' written in two styles 3
Figure 1.3 Basic rules of stroke order 3
Figure 2.1 Three examples of alphabetical and pictorial (basic strokes) characters.... 9
Figure 2.2 The structure of the word '*alphabet*' in three different languages.... 10
Figure 2.3 The historical development of Chinese characters 17
Figure 2.4 2-D pictorial format of Chinese characters 17
Figure 2.5 Topological structure of Chinese characters 18
Figure 2.6 Tseng and Chuang's method 19
Figure 2.7 Hung and Chan's method 20
Figure 2.8 Hsieh and Lee's method 20
Figure 2.9 Lin et al method 21
Figure 2.10 Guo and Xuan's method 22
Figure 2.11 Mao and Kuo's method 23
Figure 3.1 Four writing styles of Chinese characters 27
Figure 3.2 An example of illustrating the five-level structure 30
Figure 3.3 Classification of radicals 32
Figure 3.4 Examples of the *Cang-Jie* method 33
Figure 3.5 An example proceeded by the three-layer hierarchy 33
Figure 3.6 Combinations of strokes 35
Figure 3.7 A radical fuzzy set 38
Figure 3.8 A character fuzzy set 39
Figure 3.9 Varying Y shapes 39
Figure 3.10 Neocognitron architecture 42
Figure 3.11 Illustration of S-cells and C-cells for recognising a radical 43
Figure 3.12 An example of chain codes 48
Figure 3.13 Format of radicals 48
Figure 4.1 The main and secondary problems in CCRS 53

Figure 4.2 Dataflow diagram of CCRS at the top level 53
Figure 4.3 Further dataflow diagram of CCRS... 54
Figure 4.4 Function analysis diagram.. 55
Figure 4.5 Data features of CCRS 55
Figure 4.6 The structure of Chinese Character Recognition System (CCRS).......... 57
Figure 4.7 Four data models for representing a character...................................... 59
Figure 4.8 The structure chart of modules in CCRS.. 60
Figure 4.9 Vagueness of radicals 62
Figure 4.10 Compressed features of radicals in a character 63
Figure 4.11Possible position of a radical.. 64
Figure 4.12 Angle of strokes connected... 66
Figure 4.13 Location of strokes connected 66
Figure 4.14 Discontinuous contour.. 66
Figure 4.15 Position variance of a radical 68
Figure 4.16 Size variation.. 68
Figure 4.17 Shape transformation.. 69
Figure 4.18 Ambiguity of shape discrimination 69
Figure 4.19 Some training patterns in the hidden layers....................................... 70
Figure 4.20 Architecture of the associative memory neural network in the Recognition subsystem.. 72
Figure 4.21 Function of the hidden-2 layer 73
Figure 4.22 Code data.. 74
Figure 4.23 Architecture of the Restructuring subsystem...................................... 74
Figure 4.24 Format of a chain code 74
Figure 4.25 Filling order.. 75
Figure 4.26 Selection of omitted codes... 76
Figure 4.27 Impossible position.. 76
Figure 4.28 The quality of a chain code 77
Figure 4.29 Structure of the code database.. 77
Figure 4.30 An example of possible solutions 78
Figure 4.31 Architecture of the Postprocessing subsystem.................................... 78
Figure 5.1 The performance structure of the main system 83
Figure 5.2 The performance structure of the subsystems....................................... 84

Figure 5.3 Performance data 84
Figure 5.4 (a) The monitoring time for learning formal radicals 85
Figure 5.4 (b) The user time 85
Figure 5.4 (c) The system time 86
Figure 5.5 (a) An example of the time statistics 86
Figure 5.5 (b) Details of the executing time 87
Figure 5.6 Test characters 88
Figure 5.7 Different shapes of radicals 88
Figure 5.8 Results of different orders 89
Figure 5.9 Examples for expansibility of rules 89
Figure 5.10 Two special cases 90
Figure 5.11 A case of strokes treated as a radical 90
Figure 5.12 Position adjustment 91
Figure 5.13 Shape selection and normalisation 91
Figure 5.14 Recognition of different radicals 92
Figure 5.15 Classification of test radicals 93
Figure 5.16 Some learning patterns 94
Figure 5.17 Different weights 94
Figure 5.18 Effects of different weights 95
Figure 5.19 Results of test data with different error rates 95
Figure 5.20 Weights for some sub-nets 96
Figure 5.21 Patterns in different orders 96
Figure 5.22 Results of different neurone structures 96
Figure 5.23 Test results with setting the parameters θ_i and φ_j to 0 97
Figure 5.24 Test results with modifying the parameters θ_i and φ_j to ½ ΣW_{ij} 98
Figure 5.25 Recognition rate of radicals in different categories 99
Figure 5.26 Recognition frequency of the radicals in the categories 99
Figure 5.27 A case of ambiguity 100
Figure 5.28 Restructuring of chain codes 100
Figure 5.29 Combinations of chain codes 101
Figure 5.30 Illegal positions of some radicals 102
Figure 5.31 An example of codes omitted 102
Figure 5.32 Test results on verifying chain codes 102

Figure 5.33 Results of the quality rate 103
Figure 5.34 Results of error assessment 104
Figure 5.35 The quality and reliability rates 104
Figure 5.36 Results of a chain code corresponding to a character 105
Figure 5.37 Results for the assessment action 106
Figure 5.38 The fail case in the assessment action 106
Figure B.1 The structure of CCRS programs in the main system B-1
Figure B.2 The structure of executing programs for the five subsystems B-2
Figure B.3 The structure of CCRS examples B-3
Figure B.4 Illustration of files in CCRS B-6
Figure B.5 The structure of the code database B-8
Figure C.1 Test data for outside and inside C-2
Figure C.2 Test data for inside and outside C-2
Figure C.3 Test data for top, middle and bottom C-3
Figure C.4 Test data for top and bottom C-3
Figure C.5 Test data for left, middle and right C-4
Figure C.6 Test data for left and right C-4
Figure C.7 Test data for radicals C-5
Figure C.8 Test data for special characters C-5
Figure C.9 Outcomes for outside and inside in Preprocessing C-6
Figure C.10 Outcomes for inside and outside in Preprocessing C-6
Figure C.11 Outcomes for top, middle and bottom in Preprocessing C-7
Figure C.12 Outcomes for top and bottom in Preprocessing C-7
Figure C.13 Outcomes for left, middle and right in Preprocessing C-8
Figure C.14 Outcomes for left and right in Preprocessing C-8
Figure C.15 Outcomes for shape selection in Normalisation C-9
Figure C.16 Learning patterns in Recognition C-9
Figure C.17 Weights for sub-nets in Recognition C-11
Figure C.18 Outcomes in Recognition C-11
Figure C.19 Outcomes in Restructuring and Postprocessing C-12

List of Tables

Table 2.1 GB standard 25

Table 2.2 Eten Big5 standard 25

Table 3.1 Different features of three handwritten types 28

Table 4.1 Measurable criteria for error detection 79

Table B.1 Data in the code database B-8

Glossary of Terms

Associative memory (AM) neural network, a neural network with the function of recalling a stored pattern from its partial or noisy input.

Caoshu, a cursive writing style for Chinese characters, which has features of executing swiftly with all strokes flowing together and sometimes simplifying some radicals.

CCRS, Chinese Character Recognition System.

Four-corner method, a method of using a code to stand for a stroke or a combination of strokes from one of the four corners of a Chinese character.

Fuzzy possibilistic reasoning, a knowledge-based approach of using information-compressed representations, termed as possibilistic inference if-then rules, to validate a possible resolution from various restrictions.

Fuzzy syntactic method, applying the syntactic approach to fuzzy logic to increase the descriptive power of syntactic recognition by fuzzifying the concepts of a grammar and the associated language.

Kaishu, a style of standard script of Chinese characters.

Neocognitron neural network, a neural network with the function of recognising patterns regardless of where they are placed in the field of view of the retina.

Normalisation, normalising a radical for its size, position, and shape, a subsystem of CCRS.

On-line recognition, recognition is taking place while the writing is in action.

Off-line recognition, recognition is carried out after the writing.

Pictorial characters, words derived from actual objects in the real world for writing in a particular language.

Preprocessing, processing an input character and segmenting it into radicals, a subsystem of CCRS.

Postprocessing, examining a chain code using the method of analysis of error probability for error detection, correction and rejection, a subsystem of CCRS.

Radical, a component of a Chinese character.

Recognition, identifying a radical in order to get its category code, a subsystem of CCRS.

Restructuring, reorganising radicals from the style of a character to a chain code for verification using a code database, a subsystem of CCRS.

SSADM, Structured System Analysis and Design Method.

Stroke, basic component of radicals or token-radicals.

Three-layer hierarchy, a structure representation consisting of character-radical-stroke in Chinese characters.

Token-radical, a simple combination of basic strokes.

Two-layer hierarchy, a method of presenting the structure of Chinese characters from strokes directly to characters.

Xingshu, a style of semi-cursive script between '*kaishu*' and '*caoshu*' in Chinese characters.

Glossary of Symbols

$\Re_j$	A possibilistic inference rule
$\cap \Re_j$	To validate a possible resolution from different restrictions
$\pi(D)$	Possibility distribution on the domain D
$Poss_\pi$	Possibility measures
ξ^S_j is μ_j	Antecedent in the IF clause in the possibilistic inference if-then rules, μ_j is a subset of possibility distributions on the space sets S_j with regard to j. ξ is a variable whose values can be arbitrary possibility distributions on S_j
ξ^T_j is v_j	Consequence in the THEN clause in the possibilistic inference if-then rules, v_j is a subset of possibility distributions on the space sets T_j with regard to j. ξ is a variable whose values can be arbitrary possibility distributions on T_j
S-cells	Neurones for extracting features in Neocogritron neural network
C-cells	Neurones for tolerating errors of the features extracting by the S-cells in Neocogritron neural network
V-cells	Neurones for supporting S-cells (subsidiary inhibitory) in Neocogritron neural network
U_{SI}	The S-layer
U_{CI}	The C-layer
U_{SI} (n, k)	The output of an S-cell in layer U_{SI}
U_{CI} (n, k)	The output of an C-cell in layer U_{CI}
U_{VI} (n)	The output of an V-cell in layer U_{SI}
n	A two dimensional set of co-ordinate indicating the position of cell's receptive field centre in the input layer U_0
k	A serial number of the cell plane, $1 \leq k \leq K_{SI}$ for S-cells; $1 \leq k \leq K_{CI}$ for C-cells; $\mathbf{K_{SI}}$ and $\mathbf{K_{CI}}$ are values in layers

r_I (k)	The efficiency of the inhibitory input to this S-cell and controls the selectivity in feature extraction
α_I (v, k, K) and **b_I** (k)	The excitatory and inhibitory interconnecting coefficients, respectively
v	The inhibitory input
C_I (v)	The strength of the fixed excitatory connections, and is a monotonically decreasing function of $\|v\|$
d_I (v)	The weight on the connection from the S-cell at position v in the receptive field of the C-cell
j_I (k, K)	The condition of joining for several S-planes **k**, sometimes joined together and made to converge to a single C-plane **K**
ψ []	A function specifying the characteristic of saturation of the C-cell
E	An energy equation for associative memory neural network
θ_i **and** φ_j	Constants of the energy equation E
W (i, j)	The connectivity matrix for the associative memory function
X_i (u)	Input patterns, u = 1, 2 ... M, i = 1, 2 ... N
Y_j (u)	Output patterns, u = 1, 2 ... M, j = 1, 2 ... N

Chapter 1
Introduction

1.1 Chinese Characters
1.2 Aims and Objectives of the Project
1.3 Outline of the Thesis

1.1 Chinese Characters

Chinese characters, used by a quarter of the world population for writing and communication in daily life, are thought to be the most difficult to learn. Usually, a native Chinese child needs to be educated for up to twelve years to become familiar with the characters and the language. The difficulty lies mainly in three areas: a variety of pronunciation, a large vocabulary with various sounds, and the complicated structures of characters.

The Chinese language is spoken by 94% of the population in China and can be divided into 7 major dialects: *Northern*, *Wu*, *Cantonese*, *Fujian*, *Hakka*, *Jiangxi* and *Hunan*. These dialects differ so much in pronunciation, vocabulary and grammar that it is difficult for a speaker of one dialect to communicate with a speaker of another. In 1924, a change in policy made *Beijing* pronunciation the standard one, and this was affirmed once again in 1955 [Zho92]. In the mainland of China, the standard language developed in *Beijing* pronunciation is called '*Putoghua*' ('Common Language'), whereas in Taiwan, it is called '*Guoyu*' ('National Language'). In Singapore and in overseas Chinese communities, another term is used: '*Huayu*' ('Chinese Language'). In the English dictionary, the standard language is translated into '*Mandarin*' ('Official Language'). However, these terms are all used to refer to the standard Chinese language.

For the 'standard Chinese language', the number of characters has reached 60,000 words. Although they have been in use as scripts for over 3,000 years, many of them have never been involved in a phonetic system for their pronunciation. In 1918, the first ever officially recognised phonetic system, '*Zhuyin Zimu*' ('Phonetic-Notation Symbols'), was promulgated [Zho86]. This was a character-based phonetic symbols formed from simplified versions of ancient Chinese characters. It was, therefore, much more appropriate for use within the confines of China itself rather than world-wide. Currently, a modified version, shown in Figure 1.1 (a), is still being used in Taiwan as the standard phonetic symbols of Chinese pronunciation.

In 1928 another phonetic system, '*Guoyu Romazi*' ('National Language Romanisation'), that was more suitable for use outside China was endorsed and promulgated. However, '*Guoyu Romazi*' was inconvenient, because its methods of transcription were too complicated. Thus, in 1958, a third revised and legally acceptable system was introduced: '*Pinyin*' ('The Chinese Phonetic alphabet') as illustrated in Figure 1.1 (b). From then on, people who want to learn the standard Chinese language will first be taught '*Pinyin*' or '*Zhuyin*' (used in Taiwan) so that they can use it as an aid to learn Chinese characters afterwards. In 1982, the International Standards Organisation (ISO) officially recognised '*Pinyin*' as the international standard for the phonetic transcription of Chinese and it has been given the code ISO 7098.

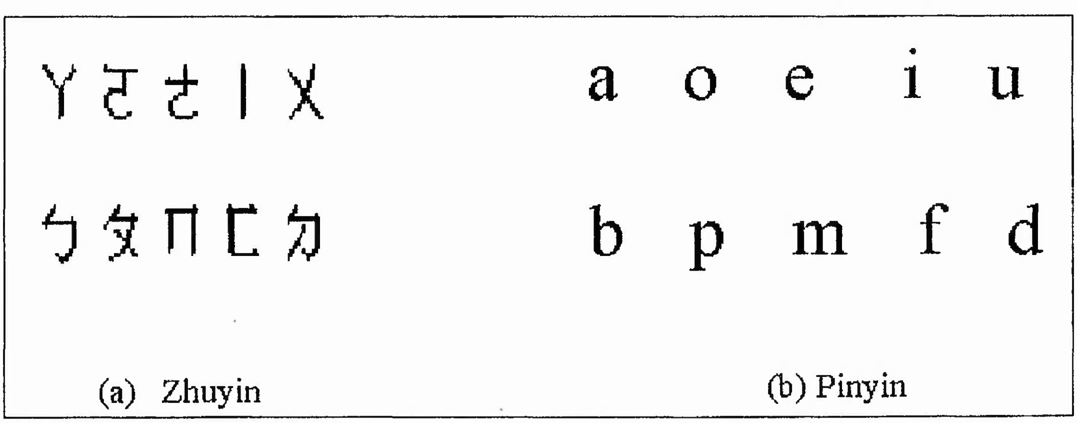


Figure 1.1 Chinese phonetic symbols in two styles

From a vocabulary of 60,000 Chinese characters, a working knowledge of about 7,000 characters is necessary to learn and write in modern Chinese. Of the 7,000 characters in common use in writing, some are still written in two styles: simplified and complex.

The simplified style is derived from the complex one and was enforced by the Chinese Government in 1956 to make it easier to learn and write. At that time, 515 characters and 54 radicals of the 7,000 characters were simplified by following the 'Scheme for Simplification of Chinese Characters' in the mainland of China. In 1964, these simplified characters and other characters that could be simplified by analogy were collected to form a total of 2,235 characters. In 1988, the Chinese Government promulgated a list of 7,000 characters in a 'Table of Modern Chinese Characters in Current Usage', 3,500 of which are commonly used.

Compared to the complex style, the simplified one possesses two advantages. Firstly, it has a greatly reduced number of strokes in a character. To show the advantage of the simplified style, Figure 1.2 gives an example of the character '*book*' in two writing styles.

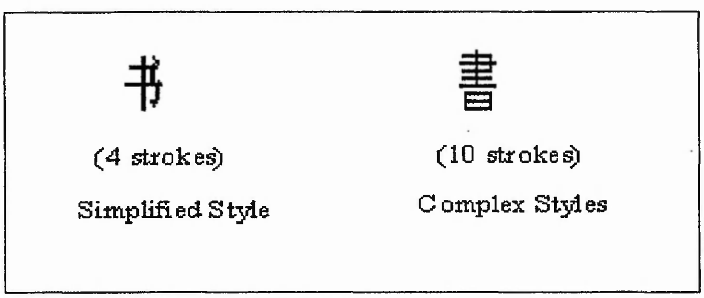


Figure 1.2 The Chinese character 'book' written in two styles

Secondly, it has a fixed order of strokes in a character. In the complex style, the order of strokes in a character was not pre-specified [Zho92]. The fixed order of strokes is defined as: first horizontal, then vertical; first left-diagonal, then right-diagonal;

十	→	一 丨		horizontal → vertical
八	→	丿 乀		left-diagonal → right-diagonal
志	→	士 心		top → bottom
明	→	日 月		left → right
闷	→	门 心		outside → inside
国	→	冂 玉	一	outside → inside → close
小	→	亅 八		middle → two sides

Figure 1.3 Basic rules of stroke order

from the top to bottom; from left to right; first outside, then inside; finish inside, then close; first middle, then two sides. Figure 1.3 gives examples illustrating these rules.

After the implementation of the simplified characters in the mainland of China during the last forty years, simplified characters have been completely accepted and used. Unfortunately, these characters are not officially accepted in Taiwan, where Chinese characters are still kept in the complex writing style. In Hong Kong, the two writing styles are both in use based on personal preference. In Singapore and Southeast Asia, most of the Chinese use simplified characters as well as '*Pinyin*'.

The internal structure of the simplified characters, however, still has two defects: too many strokes in a character; and no fixed category of radicals. Generally, a Chinese character consists of 1 to 30 strokes [Tap90] with most of the characters being composed of 8 to 16 strokes. A character can include one or more radicals. A radical may consist of 3 to 8 strokes. The vocabulary of Chinese characters can be defined as 3,500 characters in common use, 7,000 characters necessary in writing and 60,000 characters in total.

Having compared with the most popular 11 Chinese dictionaries published in the mainland of China, Taiwan and Hong Kong, confusion can be found in the categories of radical systems among these dictionaries. Some dictionaries contain only 188 radicals ('*Xin Hua*' Dictionary) while others contain 194 ('*Zhong Zheng*' Dictionary), 200 ('*Han Yu*' Dictionary), 214 (three dictionaries of '*Zhong Hua*', '*Zhong Wen*' and '*Ci Yuan*'), and even 227 radicals (Chinese English Dictionary) [Lam92].

The simplified Chinese characters can be recognised in three different ways: phonetic, code and character contour. The phonetic recognition uses a character pronunciation: '*Pinyin*', to identify a character. The method adheres to a standard pronunciation and the capability of identifying different characters with the same pronunciation. The code recognition is required to refer to a list of interpreting a code to a character. As an advantage, the method keeps the one-to-one relationship between a code and a character. However, its translation is too complicated to non-professional people. This method is normally used by telegraphic or other electronic communications. The character contour method is to recognise a character through its shape structure. Because the method is similar to human learning, it could be much easily accepted by general applications.

1.2 Aims and Objectives of the Project

The work described in this thesis centres on the investigation of the structure of Chinese characters used in the Chinese standard language and the simplified style, and of different methods of recognition.

Typical development of the recognition process focuses on printed, on-line and off-line handwritten characters using techniques such as a two-layer hierarchy, four-corner, radical and whole character recognition. Although their use has achieved some success, the lack of fundamental algorithms for representing the structure of Chinese characters has prevented the application of these methods to recognising characters with large vocabulary and complicated topological structure embedded with the 2-D pictorial format. This project develops a new structural representation to remedy the lack of an effective recognition process of such characters. The research also investigates methods of dealing with variable size, position, shape, vagueness and ambiguity of a character.

The objectives of the project include: (a) different representation of the internal topological structure of Chinese characters; (b) the analysis of radicals' categories; (c) simplifying the process of translating a 2-D picture to a chain code; (d) various techniques extensively used in the recognition of patterns and characters; and (e) the efficiency and utility of improvement of the architecture and the algorithms used in the project.

The project has been conducted in two stages: MPhil and Ph.D. During the MPhil stage, the investigation centres on the exploration of problems and difficulties of Chinese character recognition, comparison of different methods, and application of the three-layer hierarchy representation - a new method proposed by the author for off-line recognition of Chinese characters. Using uncertainty and topological analysis, fuzzy possibilistic reasoning and associative memory, a fuzzy neural network system is analysed, designed and partially implemented for examining the application of the hierarchy. At the PhD stage, the work involves a combination of theoretical and experimental study of different shape transformation and further verification at the character level. Neocognitron neural network and the chain code method are developed to further improve the system. The experimental work resulted in these new techniques is used to deal with the variability and flexibility of a character. Methods of the three-layer hierarchy representation and the character-radical-code process are compared with others in Chinese character recognition.

1.3 Outline of the Thesis

The organisation of the thesis is outlined below.

Chapter 2 reviews current techniques used in alphabetical character recognition and Chinese character recognition. Difficulties that occur in the recognition of both characters and limitations of existing methods in Chinese character recognition are re-examined.

Chapter 3 proposes a novel method of the three-layer hierarchy to represent the structure of Chinese characters. A combination and comparison of the structural representation of Chinese characters, for instance, five-level structure, *Cang-Jie* method, are made. Different methods are proposed and evaluated for dealing with difficulties of vagueness, compression, and representation of such characters. In the representation scheme, fuzzy possibilistic rules are applied to extract radicals from a character. Variability of a radical in position and shape has been analysed and discussed. Classification, recognition, and translation of radicals from a 2-D pattern to a single code are analysed. The methods of forming a chain code in a 1-D format and matching it to one in a code database are applied for verification. Several factors that affect the formation of a chain code, such as organising cues and the establishment of the code database, are presented. An assessment scheme is applied to deal with error detection, correction and rejection of a chain code using the method of analysis of error probability.

Chapter 4 describes a software system: Chinese Character Recognition System (CCRS), developed by the author. The system analysis includes problem definition, dataflow diagrams, functions, and data analysis. Dataflow diagrams for function and data analysis are produced for the design of the system.

The system design includes the determination of framework, data model and program design. It consists of a main system and five subsystems: Preprocessing, Normalisation, Recognition, Restructuring and Postprocessing. Two further subsystems are added to process and maintain the CCRS.

The system is designed and implemented to recognise characters in printed and standard writing using techniques of uncertainty and topology, analysis, fuzzy possibilistic reasoning, Neocognitron, associative memory neural networks, and the chain code method. Special software for the system has been written using the C programming language and X View function.

Chapter 5 centres on different test results and discussions of CCRS experimental data. Various special cases are discussed. Improvements of the system to deal with vagueness and ambiguity (two separate characteristics) during recognition are carried out at several stages to increase recognition rate. Strengths and weaknesses of the different algorithms devised from the research program are discussed in terms of the four basic difficulties of vagueness, variability, ambiguity and flexibility in pattern and character recognition, and special problems in Chinese character recognition.

Chapter 6 gives an overview of the contributions, methods and techniques used in the research and comparison of the three-layer hierarchy method with others. Several conclusions of the overall project are drawn. Finally, suggestions are provided for future work and possible development of the research.

The supporting tools, developed software, experimental data and outcomes, and publications are included in Appendices for reference.

Chapter 2

Limitations of Existing Methods

2.1 Introduction
2.2 Difficulties and Methods in Recognising Alphabetical Characters
2.3 Chinese Character Recognition
2.4 Comments

2.1 Introduction

Characters used in different natural languages can be classified into two major categories: the alphabetical and the pictorial. Alphabetical lexical items (words) consist of a set of letters arranged in a fixed order. Such letters are, for instance, 26 characters in English, French, ... or even a syllabary of 46 full-size and 25 half-size, additional markings indicating subtle phonetic differences, symbols in the Japanese '*Kana*' (the Japanese is written in a mixture of '*Kanji*' (Chinese characters) and '*Kana*' ('*Hiragana*' and '*Katakana*')) as shown in Figure 2.1 (a) and (b) respectively.

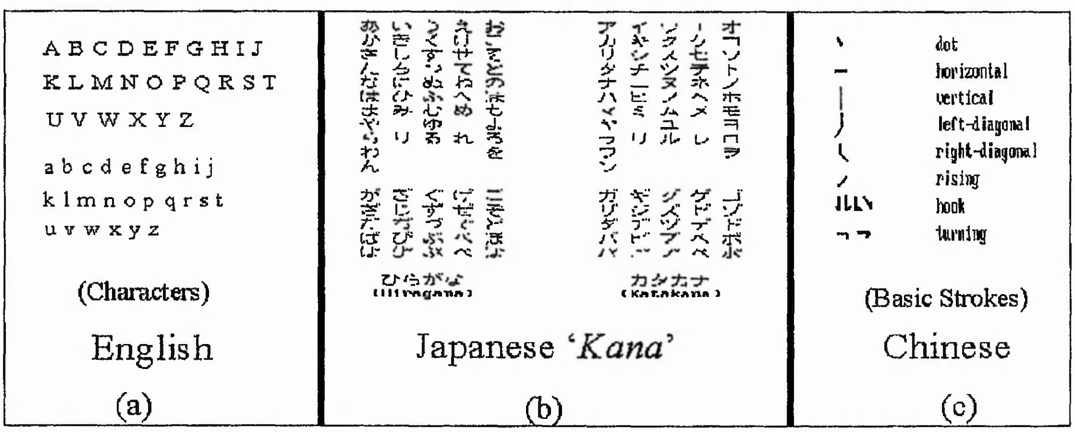


Figure 2.1 Three examples of alphabetical and pictorial (basic stroke) characters

Pictorial characters are words derived from actual objects in the real world for writing in a particular language, for instance, Chinese characters and Japanese '*Kanji*'. In the Chinese language, a character is composed of basic strokes as illustrated in Figure 2.1 (c).

The difference between the alphabetical and pictorial characters leads to significant differences in the structure of words. For instance, the word '*alphabet*' in English is made of a set of characters from the English letters in a one-dimensional structure. This structure is similar to a Japanese word written in '*Kana*'. In Chinese characters, the structure of word '*alphabet*' is a combination of different strokes in a two-dimensional picture. Figure 2.2 shows the word written in three different languages: English, Japanese '*Kana*' and Chinese, to reveal the differences of the internal structure among alphabetical and pictorial characters.

alphabet	アルファベット	字 母
(English)	(Japanese)	(Chinese)

Figure 2.2 The structure of the word 'alphabet' in three different languages

In the research of character recognition, mainly Optical Character Recognition (OCR), the alphabetical and pictorial characters are major objects to be investigated and recognised by machines to translate human-readable printed or hand-written characters to machine-readable codes [Imp91].

To gain an insight into the development of Chinese character recognition, current techniques applied to both alphabetical and pictorial characters are reviewed. The review for recognising alphabetical characters centres on methods and techniques dealing with the four major features: vagueness, variability, ambiguity and flexibility of a character.

These features are generally seen as the four main difficulties in recognition of various characters. The understanding of these features is discussed in Section 2.2.1. Methods and techniques that deal with these difficulties with particular reference to current work of Chinese character recognition are surveyed in Section 2.2.2. The review emphasises three specific difficulties in Chinese characters: a 2-D pictorial format, topological structure, and a large vocabulary. Different methods of structure representation and recognising techniques of Chinese characters are analysed. Comparison and evaluation of these methods are reviewed in Section 2.3 and some comments are given in Section 2.4.

2.2 Difficulties and Methods in Recognising Alphabetical Characters

Research in the recognition of alphabetical characters has achieved significant results in on-line and off-line, printed, hand-written, single character and script since it started in 1958 [Gov90]. A wide range of methods and techniques have been devised and implemented in different industrial and commercial systems [Tap90]. Evaluation of these methods and techniques is difficult, due to the lack of objective, measurable and universal definitions of writing characteristics, and the absence of 'standard' databases that would enable algorithm comparison [Gue93]. Therefore, the methods and techniques are reviewed around specified difficulties on computer recognition.

2.2.1 Difficulties

Basic problems in pattern and character recognition are variable size, position and orientation of an object [Cow90]. Based on these basic problems, the main difficulty for computer aided recognition is to deal with the four major features of a character: vagueness, variability, ambiguity and flexibility.

The **vagueness** of a character is associated with the difficulty of making sharp or precise distinctions among characters [Kli88]. This feature centres on a character with (a) a fuzzy boundary that is hard to be identified due to difficulties in determining its shape or position, or (b) noise in the data, for instance, deformation of the image or a disconnected line [Imp91].

The **variability** of a character is related to the range of different forms used to represent it [Nou91], such as a character with a variety of fonts, sizes, shapes, stroke numbers, orders or directions, the movement of a whole character or its components, or orientation in cursive scripts or isolated characters [Imp91].

The **ambiguity** of a character refers to the 'one-to-many relations' in which the choice between two or more alternatives is left unspecified for a character [Kli88]. The feature leaves the character with no unique identity for its structure, shape, position or stroke order.

The **flexibility** of a character is related to "where" and "how" a character is written [Tap90]. There will be different if a character is written in different places by many distinct ways, for instance, written in boxes by both vertical and horizontal spacing well controlled, or on lined paper by only the vertical spacing controlled or on blank paper by different writers or even by the same writer.

2.2.2 Methods and Techniques

Although there are different techniques used in character recognition systems, all the systems developed generally share a common sequence: Preprocessing, Features extraction, Classification and Postprocessing [Imp91]. Some systems classify Features extraction and Classification into Recognition [Nou91].

Preprocessing includes all the processes that are necessary to bring the input data into a form acceptable to the feature extraction phase [Hil93]. The main objectives at this step include a reduction in the amount of information to be retained, the elimination of certain imperfection and normalisation of the data. The main techniques available are smoothing, normalisation, segmentation, and line segment approximation.

Features extraction is to extract different types of shape features from a character [Cox94]. Its techniques include template matching and correlation, distribution of points, transformations, series expansions and structural analysis.

Classification organises to which class of features the input belong [Imp91]. Techniques include statistic analysis of the training set features, Bayesian rules based on *prior* knowledge, shape derivation and matching, and hierarchical feature matching in the form of decision trees.

Postprocessing uses the contextual information to detect errors and even to correct them [Imp91]. Two approaches are used in the Postprocessing: a dictionary of words and a set of binary matrices that approximates the structure of a dictionary.

In general, methods used in **preprocessing** for segmentation of characters could be classified into three aspects: (a) dividing a character into primary elements for determining the appearance of the elements with a vague boundary; (b) describing the geometrical and topological properties of a character, for instance, position, various directions and slopes, using mathematical models to cope with problems of variable characters [Nou91]; and (c) separating a character into components to be described by a chain code that includes more codes, for instance, the Markov models [Cho95], to deal with the character written in a flexible way [Nou90].

Advanced techniques used in **features extraction** include fuzzy logic, statistical and syntactic structures, rule-based, and neural networks. Fuzzy logic methods introduced by L. A. Zadeh [Zad65], use concepts of logic set of class membership functions to represent features of their members. In 1974, Siy and Chen [Siy74] applied fuzzy logic set to represent features of straight line, portion of a circle and circle instead of the probabilistic approach which cannot be calculated for non *priori* knowledge for character recognition. Afterwards, Poon et al [Poo93] used a fuzzy language for syntactic pattern recognition. In the recognition process, the 11 primitive strokes and a set of inferences based on production rules and their membership grades are defined for expressing a stroke in a character. The fuzzy logic method has merits in dealing with vague and noising characters. However, the method still has problems of how to distinguish between shape deformation and an internal distance of a character, and between considering the number and order of a stroke and free recognition [Wak92].

The statistical method uses sets of characteristic measurement values to extract features from the pattern data. The syntactic approach applies a set of rules that are developed using formal language-theoretic models to describe the structure of a pattern class in terms of primitive features of the pattern. The set of rules is formalised into a grammar. Although both statistical and syntactic methods are based on mathematical models, the statistical method cannot handle well interconnection between complex patterns [Ham92], and ambiguity occurs while a pattern is generated by more than one pattern grammar in the syntactic approach [Man87].

Using neural networks that simulate the working procedure of the human brain to extract features of a character is another technique in character recognition [Wil93]. Johnson et al [Joh88] employed the neocognitron neural network, a fixed architecture network with thousands of neurones and connections on layers of input [Woo96], for pattern recognition. The network has built up relations between the extraction process and the concepts of discriminate surface features.

In comparing neural networks with rule-based methods for feature extraction, the method of neural networks attempts to simulate human intelligent behaviour using adaptive learning. The rule-based method uses logical symbol manipulation to represent features of patterns [Wil93]. Although the two methods could not cope well with the problem of confusing characters, applying human knowledge to the two methods is an effective suggestion for improving their recognition [Nad93].

Methods and techniques used in the **classification** process include Fourier analysis, component combination, syntactic, and statistical methods. In the application of Fourier analysis to classification of numbers and characters, Shridhar and Radreldin [Shr84] used 4 quadrants, horizontal and vertical transition, and 4 directions to determine the 2-D features of a stroke in a character. They suggested that misclassification could be eliminated only if topological structures of the characters are considered in the recognition algorithms. Component combination uses a coding chain to identify a component in a character [War88]. The classification process only looks for a combination of components in the dictionary. The syntactic method forms a grammar to the classification process. The statistical method uses a description of stroke parameters to complete character classification [Nou90].

The dictionary method in **Postprocessing** has been shown to be the most effective for error detection and correction [Imp91]. The main weaknesses for the method are time-consuming searches and comparisons, and the increasing size of the dictionary. An improvement on this method could use the probability of n-grams (letter pairs or triplets) represented in a binary matrix.

2.3 Chinese Character Recognition

Computer recognition of Chinese characters was acknowledged to be a very hard task and regarded as one of the ultimate goals of character recognition research [Gov90]. The first researchers to face the challenge were Casey and Nagy at IBM in 1966 [Man86]. In their attempts, 1,000 Chinese characters had been recognised by using a step by step approach. Their work showed that the number of strokes and their position could be adequate for recognition.

In the early 1970s, Parks et al [Par74] produced a recognition method of hierarchical structure for extracting topological features of characters. In 1972, Tou and Gonzalez [Man86] introduced a two-stage scheme to recognise Chinese characters. At the first stage, some measurements are taken by means of measuring grids in order to separate the pattern classes into several subgroups; and at the second stage, a number of specialised features are extracted. In the middle of 1970s, Pavlidis used the split-and-merge algorithm to produce polygonal approximations of the characters that could provide enough information for both the character shapes and the syntactic analyser [Man86].

In the 1980's, the structural analysis approach was enhanced by providing about 100 primitive strokes to permit constituent shape distortion and improve stroke recognition accuracy [Wak92]. However, the stroke number and stroke order permutations were not successfully realised, due to lack of an efficient systematic search technique of combinatorial exhaustion. Sekita et al [Gov90] introduced a method of extracting features by using spline approximation. The method represents a character using its contour expressed by well-approximating functions and stable break points that characterise the connection of the strokes, so that it provides proper features for recognition with relaxation matching. Young [You88] suggested the use of neural networks for achieving fast recognition of Chinese characters.

Compared with the recognition of alphabetical characters, Chinese character recognition was developed quite slowly from 1960s to the early of 1980s. The main reasons are the complex structure of Chinese characters and lack of good methods to deconstruct the structure [Hil93].

2.3.1 Characteristics of the Structure

The complex structure of Chinese characters is formulated through a long history (about 5,500 years recorded in history). Early Chinese characters were mainly symbols and *pictographs* that could also represent some abstract concepts of daily life as shown in Figure 2.3 (a). In order to express more complex ideas and concepts, *pictographs* were developed and combined to form *ideographs* for multiple meanings. These *ideographs* form some 90% of the total Chinese characters in current usage [Scu91]. Most *ideographs* are made up of two components: (a) a radical, i.e. a *pictograph* before it becomes part of an *ideograph*, which indicates the classification of a character; and (b) a '*phonetic*' symbol for partially aid the pronunciation of a character. Figure 2.3 (b) shows several examples of the *ideographs*' development.

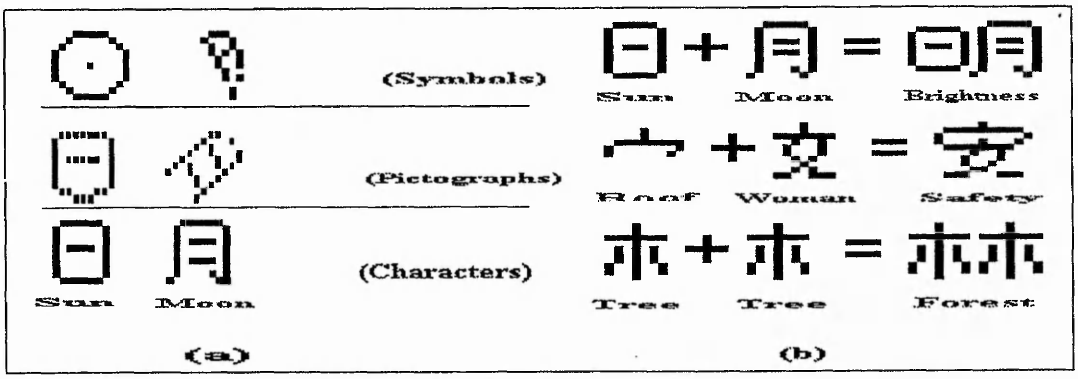


Figure 2.3 The historical development of Chinese characters

Chinese characters possess three major features in their structures and quantities: a two-dimension (2-D) pictorial format, topological structure and a large vocabulary.

In the **2-D** pictorial format, basic components, strokes, can be situated at any position of a character. Figure 2.4 (a) shows that the stroke '*horizontal line*' can be located at several places in a character. In Figure 2.4 (b), the stroke '*horizontal line*' may change its identity once its direction is altered. Figure 2.4 (c) displays that the stroke '*horizontal line*' has three different lengths in a character.

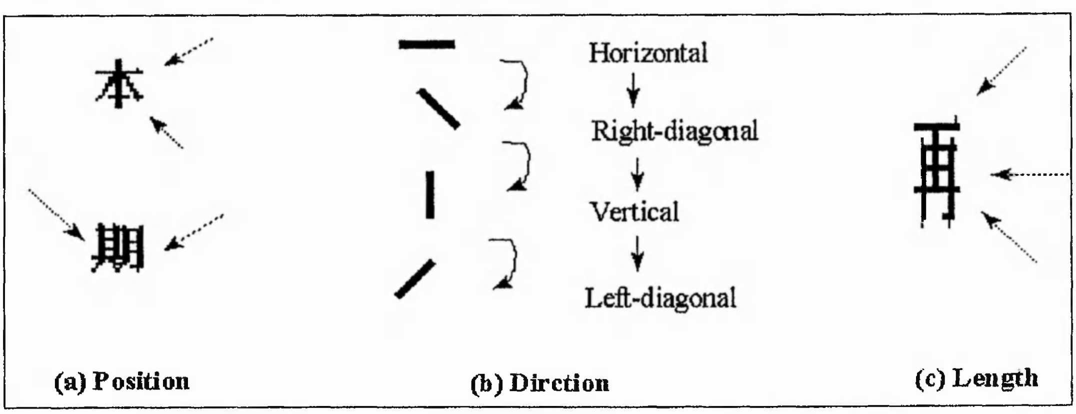


Figure 2.4 2-D pictorial format of Chinese characters

The **topological structure** of a character means that the character is combined from or deconstructed into several components as shown in Figure 2.5 (a). The same component may appear in different characters as illustrated in Figure 2.5 (b). A component can be located at different positions in a character as shown in Figure 2.5 (c).

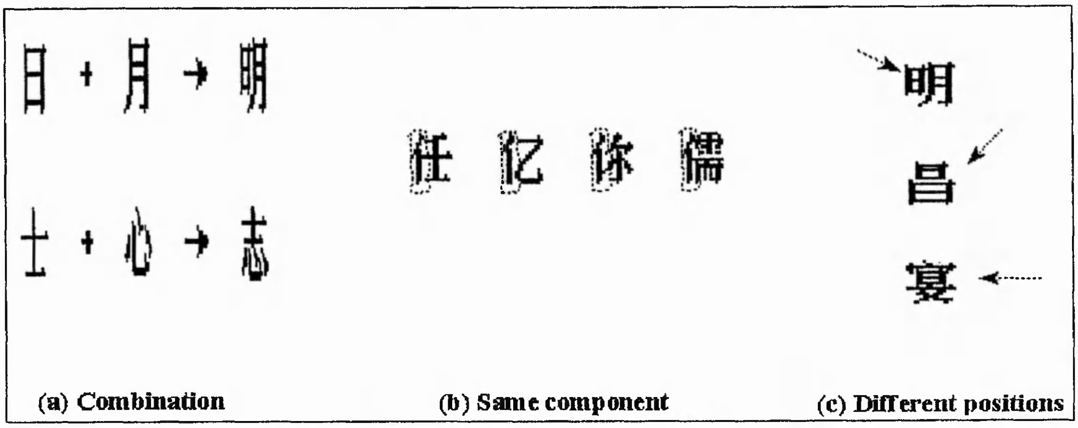


Figure 2.5 Topological structure of Chinese characters

The **vocabulary** of Chinese characters is defined as 3,500 characters for daily use, 7,000 characters necessary in writing, and 60,000 characters in total that include complex and simplified styles. Based on the feature of a 2-D picture, each Chinese character may be seen as a pattern different from others. Therefore, an adequate representation of a character requires a matrix of pixels about 10 times the number needed for a Roman letter. The vocabulary of Chinese characters is roughly equivalent to Western words in total [Gov90].

2.3.2 Methods and Techniques

The recognition of Chinese characters can be classified into three categories for process: printed, on-line hand-written, and off-line hand-written [Hsi92a]. Distinctions between on-line and off-line hand-written are: (a) that the order of strokes is available in the on-line case, whereas it is lost for the off-line case; and (b) that the on-line recognition only deals with a 1-D representation of the input, i.e. a stroke has been recognised while the writing is in action, whereas the off-line needs to analyse a 2-D image. Methods of recognition of characters in printed and off-line hand-written are reviewed in order to understand different techniques of representing a character.

i) Two-Layer Hierarchy

Basically, the two-layer hierarchy is seen to relate **strokes** directly to **characters**. The number of strokes contained in a character forms the basis of a relationship between these strokes and the characters. Determining the shape of strokes and accounting for the number of strokes contained in a characters are important stages in the method. As a traditional method of learning Chinese characters in daily life, this method has been reported in recent and current research literature of Chinese character recognition.

Cheung and Leung [Che85] used a method of parameters transformation, called the chain-code transformation, suitable for recognition of patterns containing straight lines, to map the strokes of a character into a 2-dimensional parameter space.

Tseng and Chuang [Tse92] applied knowledge-based stroke extraction method to analyse the structure of strokes in Chinese characters. The 4 primitive strokes within a possible range are given to determine the shape of a stroke. Giving an array of stroke starting and ending co-ordinates express the position of a stroke. Figure 2.6 gives the definition of strokes and allocation of their range. The range is defined as a field formed by starting and ending angles.

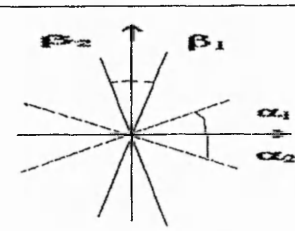
Strokes		Range	Angle Illustration
Horizontal	—	$-\alpha_2$ to α_1	
Vertical	$\vert$	$\frac{1}{2}\pi - \beta_1$ to $\frac{1}{2}\pi + \beta_2$	
Up-right-slanting	/	α_1 to $\frac{1}{2}\pi - \beta_1$	
Up-left-slanting	\	$-\frac{1}{2}\pi + \beta_2$ to $-\alpha_2$	

Figure 2.6 Tseng and Chuang's method

In their back-propagation neural networks, Hung and Chan [Hun93] divided basic strokes into five categories for encoding a character. If the number of basic strokes of a character is fewer than 13, all stroke codes are encoded. Otherwise, only the first 6 and the last 6 codes are encoded. Figure 2.7 illustrates the method.

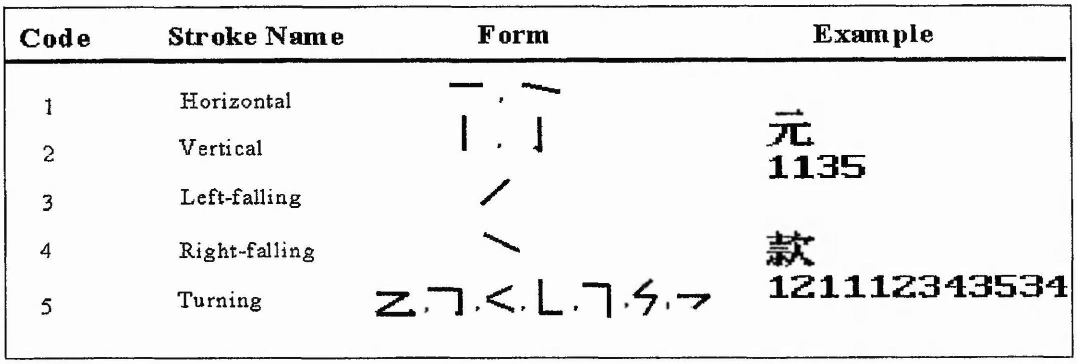


Figure 2.7 Hung and Chan's method

Hsieh and Lee [Hsi92b] applied a model-guided matching method to recognise a character. The method classified primitive strokes into 8 types defined by 8 quantities of stroke directions and 8 regular expressions. A possible range for starting and ending points of a stroke is examined. Figure 2.8 shows the method. Some ambiguous cases can be found in the method.

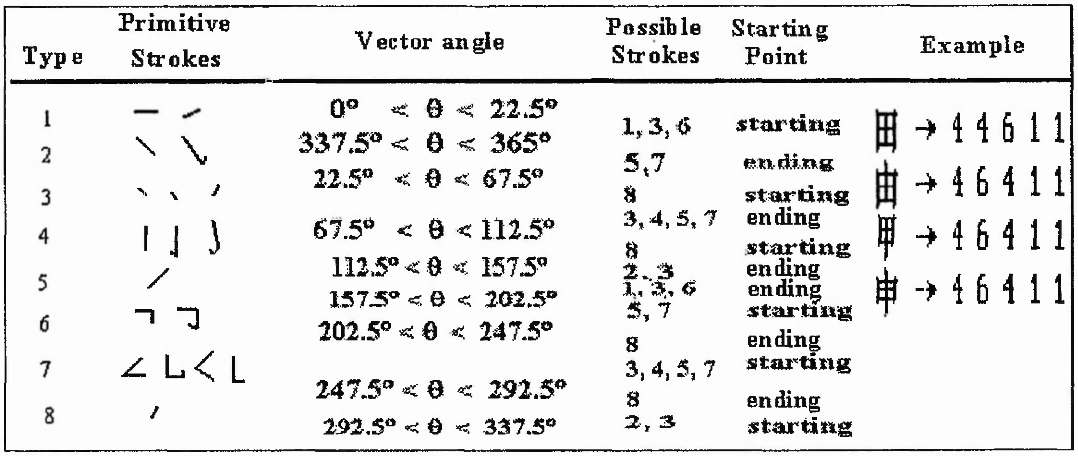


Figure 2.8 Hsieh and Lee's method

Lin et al [Lin93] applied a deviation-expansion model and dynamic match for recognising a character. The method defined 41 primitive strokes (32 basic primitive and 9 compound strokes), 16 direction quantities representing line segments, and some regular expressions to determine the shape and order of these strokes. Figure 2.9 gives some of the definitions and expressions of the method.

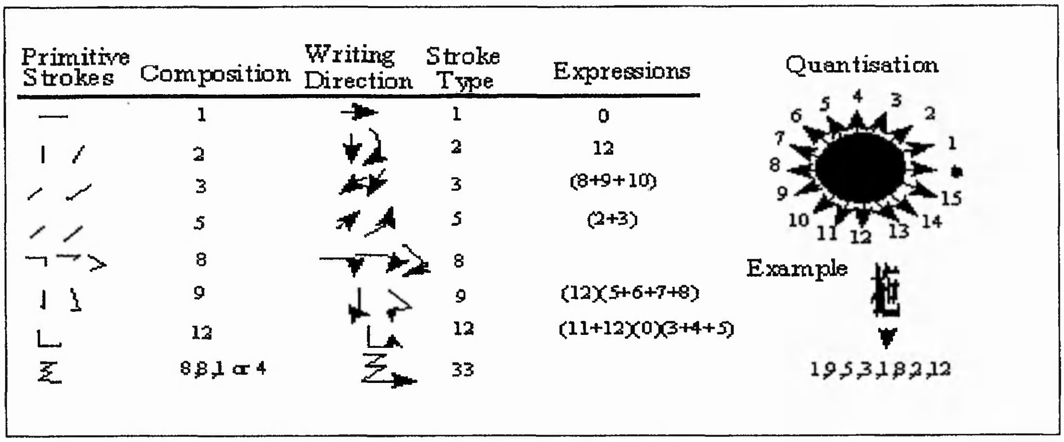


Figure 2.9 Lin et al method

Ku and Chiu [Ku92] used syntactic grammar to define stroke width, length, direction, starting and terminating co-ordinates for smoothing and thinning process. In 1993, they [Ku93] employed the knowledge inference for concept representation of strokes. In 1994, a stroke extraction method on corner point detection, location of the cross-region, edge segment merging and stroke segmentation was further investigated [Ku94].

Nakayama and Chigawa [Nak92] used a cellular neural network to extract strokes from a '*Kanji*' character as a pattern. The 4 strokes (vertical, horizontal, +45 degree, and -45 degree) and angle deviations are defined for the structure mapping of invariant features.

In conclusion, the two-layer hierarchy has some advantages of (a) only several strokes to be learnt, and (b) each stroke to be considered as a simple one-dimensional element. However, there are two major weaknesses: (a) the large number of strokes in a character, (b) the complex internal structural relationships between strokes and characters so that the combination, position and features of the strokes in the characters are difficult to be described.

ii) Four-Corner Method

The four-corner method uses a four-digit code to represent a Chinese character, where each digit (0 to 9) stands for a stroke or for a combination of strokes from one of the four corners of the character [Nag88]. These digital codes represent the classification of character semantics and are easily processed by computer. The method simulates the way of searching for a character from a Chinese dictionary called 'Four Corners Code Dictionary'.

Guo and Xuan [Guo86] applied the four-corner method based on the shape of a character at its four corner zones to recognise a character. The method is a combination of both the syntactic and the statistical methods. It should be possible to classify 6,763 characters into 1,586 groups with 97.2% accuracy. Figure 2.10 gives its grammar of codes and shape definition, and an example.

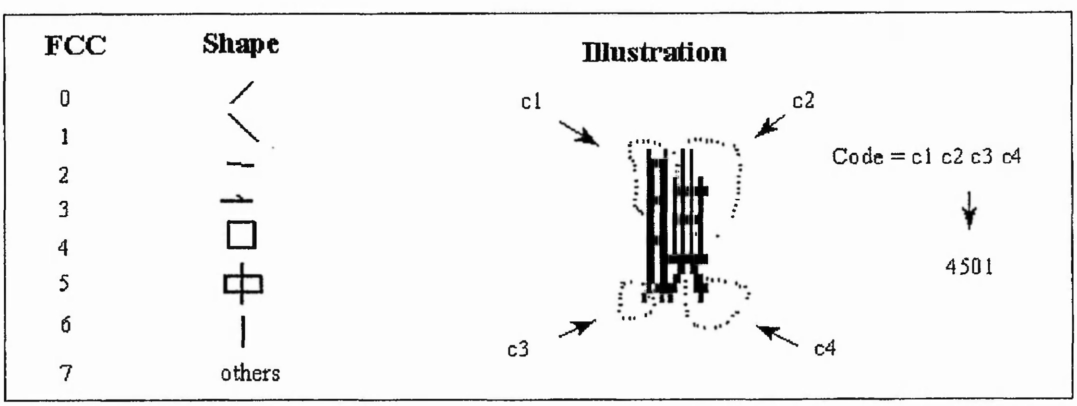


Figure 2.10 Guo and Xuan's method

A complex multi-level classification scheme, based on the 4C code obtained by encoding four corner zones of a character and 4P code obtained by encoding four peripheral rectangular zones, was developed by Huang and Chung [Hua87]. The scheme can ideally encode 4,096 classes with 1-6 characters in each class.

The four-corner method has a good structure of character classification with grammar, and translates a character into a code for easily searching of a database. The weakness of the method includes complex knowledge of its grammar (based on human observation) and the ambiguity of a code used to represent several different characters.

iii) Radical Method

The radical method uses 'components' to recognise a character. These components are radicals that have more complete features and a low degree of interaction with other radicals in a character. The method gives a partial division representation of the topological structure of Chinese characters. Using radicals to search for a character in a dictionary is a method used in the real world, although the category of radicals is altered in different dictionaries.

Mao and Kuo [Mao92] used a coded block adaptive neural network with the back propagation algorithm to extract a radical from a character. In 1994, the network had been improved using the modified Sigmoid function and the weight-dotted radical selector for recognition [Kuo94]. The 1,000 Chinese characters in common use have been trained with 25 radical blocks. Unfortunately, the method had considered only left and right divisions. Figure 2.11 displays the method in the 1994's version.

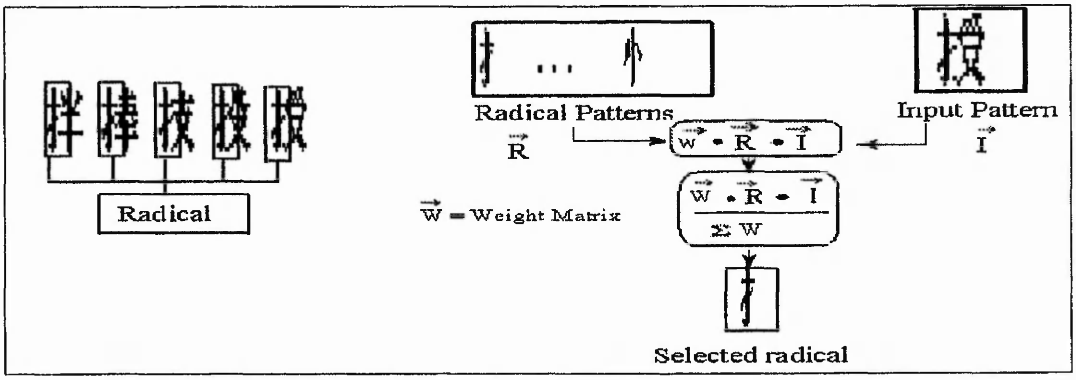


Figure 2.11 Mao and Kuo's method

Hyman et al [Hym91] employed principal component analysis that was carried out by an artificial feed-forward neural network with Hebbian and anti-Hebbian rules to extract components from a character. 40 different characters were tested and the dimension of each pattern was reduced by more than 95%. The definition of components was not given in the method.

The radical method has the merit of dividing a character into several independent and non-overlaid parts that simplify the representation of the internal structure of a character. However, the category and description of radicals in characters are not very clear.

iv) Character Method

The character method uses the concept of patterns instead of characters to avoid the problem of complex internal structure of characters.

Wakahara et al [Wak92] in their research gave two advantages of using the method. The first advantage is that recognising a whole character is easier than its parts or primitives, if it is considerably deformed. The second one is that providing a prototype or prototypes for each character is more reliable than devising *ad hoc* primitive strokes or radicals, when flexible adjustment is needed to cope with an increase or decrease in the number of characters to be recognised. Here, a prototype means a template generated by averaging x-y co-ordinate values of feature points over learning samples for each character.

However, the method has an unsolved problem in dealing with a huge number of such patterns that might be only slightly different in their structure.

2.3.3 Chinese Character Database

Chinese characters in a database are encoded in order of their computer codes, font sizes and styles. A computer code uses two bytes to represent a character. Different font sizes and styles of a character are given in a bitmap format. There are four encoding schemes to represent a Chinese character: GB, Big5, JIS and KS.

GB ('*GuoBiao*') is a national standard in China for encoding Chinese characters. The definition of the computer codes, font sizes and styles in the scheme is shown in Table 2.1.

Code	Meaning	Font Size		Font Style	
		Size	*Character*	*Style*	*Character*
A1A1 - A7FE	Symbols				
A8A1 - AFFE	Unused	16*16	a character	*SongTi*	Simplified and complex for all styles
B0A1 - D7F9	3837 characters for common use in the order of '*Pinyin*'	24*24		*KanTi*	
		16*16 to 128*128	Maximum 15 characters	*HeiTi*	
				FangSongTi	
D8F1 - F7FE	3008 characters left for necessary writing in the order of 40 radicals				

Table 2.1 GB standard

Big5 is a *de facto* standard used in Taiwan and Hong Kong. There are several different implementations, for instance, Eten Big5, HKU Big5. Table 2.2 shows Eten Big5's standard.

Code	Meaning	Font Size		Font Style	
		Size	*Character*	*Style*	*Character*
A140 -A3E0	441 symbols				
A440 - C67E	5401 characters in common use	16*15	a character	*SongTi*	Complex
		24*24		*KanTi*	
C6A1 - C8FE	408 unused			*HeiTi*	
C940 - F9D5	7652 characters left in necessary writing			*FangSongTi*	
F9D6 - F9FE	Unused				
FA40 - A0FE	3678 self-defined characters				

Table 2.2 Eten Big5 standard

JIS is a Japanese standard for '*Kana*' and '*Kanji*' characters. It has two font sizes, 16*16 and 24*24, and 11 different font styles. KS is a Korean standard.

2.4 Comments

In this chapter, the recognition of alphabetical and pictorial characters is reviewed. The four main difficulties in character recognition and features of structure of Chinese characters are discussed and investigated. Different methods and techniques to deal with these difficulties and features are surveyed and evaluated as well.

The two-layer hierarchy has the advantage of learning simple strokes to recognise various characters. The weakness of the method lies in the fact that it is difficult to describe strokes in characters.

The four-corner method has the merit of using a digital code to stand for a character, which is easy for the classification of characters and the computer processing. However, matching the shape of a character to several abstracted patterns that are represented by codes is rather difficult to be processed by the computer.

The radical method gives a partial division representation of the topological structure of Chinese characters. The category and description of radicals in characters need to be further investigated.

The character method uses the concept of patterns instead of characters to avoid the problem of complex internal structures of characters. However, a large volume of patterns over 3,500 different patterns, at least, is of main concern to the method.

Chapter 3

A Three-layer Hierarchy Method for the Recognition of Chinese Characters

3.1 Introduction
3.2 Structural Representation of Chinese Characters
3.3 Knowledge Representation Techniques
3.4 Artificial Neural Networks
3.5 Chain Code Method
3.6 Analysis of Error Probability

3.1 Introduction

There are four writing styles of Chinese characters: printed, '*kaishu*', '*xingshu*' and '*caoshu*'. Apart from the printed style that is for 'machine-printing' purpose, the other three are for handwritten usage. '*Kaishu*' is a standard script, developed from the '*lishu*' (clerical type) which was the simplification of the '*zhuanshu*' (seal script) used in ancient China as an official writing style. '*Xingshu*' is semi-cursive script between '*kaishu*' and '*caoshu*' because the flow of some strokes leads to them being written together. '*Caoshu*' is cursive writing executed swiftly with all the strokes flowing together and

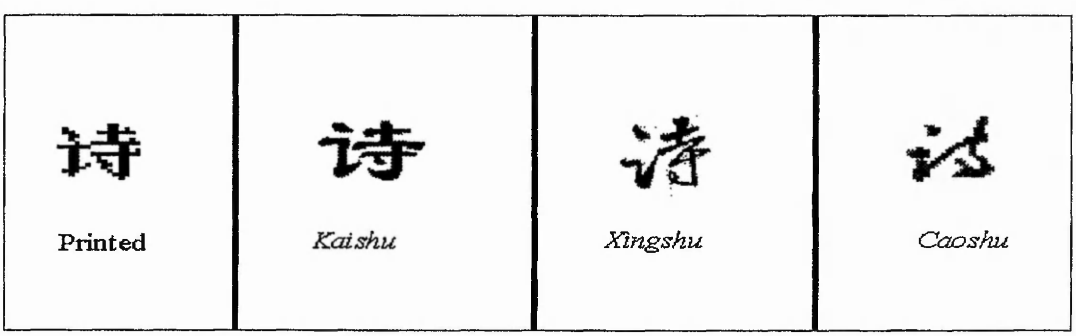


Figure 3.1 Four writing styles of Chinese characters

sometimes simplifying some characters for writing convenience. Figure 3.1 illustrates these styles.

There are some essential differences among the three handwritten styles in terms of features of writing strokes and the flowing together in strokes and radicals. Table 3.1 lists the differences [Liu88].

Styles	Flowing together between strokes	Flowing together between radicals	Feature of writing strokes
Kaishu	No	No	dot and line
Xingshu	Sometimes	No	dot, line or cursive line
Caoshu	Yes	Yes	dot and cursive line

Table 3.1. Different features of three handwritten styles

Although characters written in the '*xingshu*' and '*caoshu*' are more often and flexibly used in daily life, many researchers concentrate on recognition of Chinese characters written in the '*kaishu*' [Hsi92b] [Hil93] [Hun93] [Kuo94] [Tap90]. The reasons might be (a) the lack of an effective method for describing the complex structure of characters, and (b) cursive scripts are so different that characters written in the two styles are more complicated for computer to recognise.

Because this project focuses on developing methods for the representation of the internal structure of characters and for the recognition of characters, only the printed and '*kaishu*' styles are considered.

The representation of complex structure of Chinese characters is a key problem. After solving the problem of recognising characters in printed and '*kaishu*' styles, problems in recognition of the '*xingshu*' characters ought to concentrate on dealing with a cursive line and the flowing together of strokes. Problems in the '*caoshu*' ones would be the flowing together of radicals and simplified parts in a character.

3.2 Structural Representation of Chinese Characters

Based on the five-level structure and the *Cang-Jie* methods, a method of three-layer hierarchy is developed to deal with structural representation of characters. It is focused on problems of topological structure, 2-D extraction, and classification of Chinese characters.

The five-level structure method [Liu88] uses syntactical relationship to set up a fuzzy mathematical set with several attributes grammatically defining the structure of characters in five levels: phrase, character, radical, token-radical, and stroke. The method shows a fair representation scheme of some Chinese characters. Except for the theoretical definition, the method has not been put into a practical application.

The *Cang-Jie* method [Liu93] is a method of classifying different radicals into related categories, and uses a chain code instead of a 2-D character. The method represents an advanced technique for classifying characters following their shapes. However, the method relies heavily on observation and analysis of the structure of characters by applying human expert knowledge.

The three-layer hierarchy method provided in this thesis represents characters in three layers: character, radical, and stroke, and deals with a character in a sequence of character, radical and chain code. The method has the following advantages: (a) it uses the position independence of radicals within a character for dividing the character as a picture into several individual sub-pictures, which considerably eases recognition by computer, and (b) it reduces the complexity of a character's structure so that a character would consist of less than five radicals [Liu93] and these in turn would individually contain some two to eight strokes [Ren97c].

3.2.1 Five-Level Structure

A Chinese character may have different meanings in different phrases. To deconstruct a character into basic strokes, a phrase is introduced into the five-level structure method for understanding the structure of Chinese characters [Liu88].

In the five-level method, the structure of characters is described as phrase, character, radical, token-radical, and stroke as illustrated in Figure 3.2. From the first to the fourth level, the structure of a character is changed in its shape structure. At the fifth level, a character is altered in terms of its context meaning.

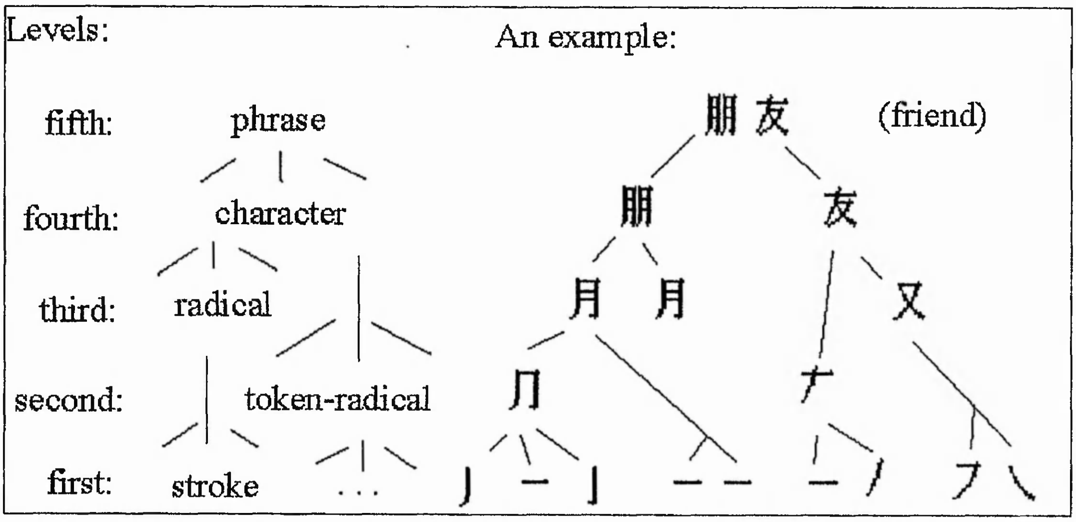


Figure 3.2 An example of illustrating the five-level structure

The description of the structure can be grammatically defined as a fuzzy mathematical set with attributes.

Set = {Component, Relationship, Description},

where:

Set stands for a set of an object with attributes described at a certain level;

Component represents elements from which an object is constructed. An element can be a subset;

Relationship signifies a structural relationship between components;

Description expresses the physical relationship between components, such as, position, order and so on.

Applying the above mathematical set to objects at different levels of the method, five mathematical sets can be defined as follows.

Phrase = {Characters, Relationship between the characters, Description}

Character = {Radicals, Token-radicals, Relationship of the radicals and token-radicals, Description}

Radical = {Token-radicals, Strokes, Relationship of the token-radicals and strokes, Description}

Token-radical = {Strokes, Relationship of the strokes, Description}

Stroke = {Description}

3.2.2 Cang-Jie Method

The Cang-Jie method, pioneered by Bang-Fu Zu [Liu93], is a method of classifying radicals into related categories using human observation and operation. Usage of the method is mainly for the input of Chinese characters from a normal computer keyboard. The method is based on the fact that a character is a combination of radicals. The inspiration for this idea came from Cang Jie, the originator of written characters in Chinese legends. Bang-Fu Zu and his team spent 8 years in the 1980's on analysing the structures of 30,000 Chinese characters and finally classifying them into 26 main categories (24 for radicals and 2 for special cases). At the same time, there were other methods produced for the input of Chinese characters as well, for instance, the *Five-Stroke* method that takes less than five strokes in a character to identify it. Compared with the *Five-Stroke* method, the *Cang-Jie* method has the advantage of extracting a more complete and a low degree of interaction with other radicals in a character through human visual inspection.

In the 24 radical categories, each category includes a standard radical and some other radicals and token-radicals that are similar to or simplified from the standard radical, in terms of either their shape or meanings. The total number of radicals and token-radicals in these categories is 108. The 2 special case categories are designed for difficult characters and for making a new character. All categories are numbered by following the alphabetic letters A, B, C, ... Z, where X and Z stand for difficult and new characters in the 2 special case categories, respectively.

The 24 radical categories are arranged in four groups: philosophy, stroke combination, physical symbol, and shape similarity. The **philosophy** group includes 7 categories, termed *sun*, *moon*, *metal*, *wood*, *water*, *fire*, and *soil*. Radicals that belong to this group are derived from abstract concepts of basic requirements of human lives in ancient China. The **stroke combination** group has 7 categories, which come from the combination of basic strokes: *left-diagonal*, *dot*, *cross*, *x connection*, *vertical*, *horizontal*, and *hook* or *turning*. The **physical symbol** group consists of 4 categories, which stand for an abstract expression of such symbols from the human body: *person*, *heart*, *hand*, and

Group: Philosophy

Category	Name	Standard Radical	Radicals/ Token-radicals
A	Sun	日	[illegible]
B	Moon	月	[illegible]
C	Metal	金	[illegible]
D	Wood	木	[illegible]
E	Water	水	[illegible]
F	Fire	火	[illegible]
G	Soil	土	[illegible]

(a)

Group: Stroke Combination

Category	Name	Standard Radical	Radicals/ Token-radicals
H	Left-diagonal	竹	[illegible]
I	Dot	戈	[illegible]
J	Cross	十	[illegible]
K	X connection	大	[illegible]
L	Vertical	中	[illegible]
M	Horizontal	一	[illegible]
N	Hook/Turning	弓	[illegible]

(b)

Group: Physical Symbol

Category	Name	Standard Radical	Radicals/ Token-radicals
O	Person	人	[illegible]
P	Heart	心	[illegible]
Q	Hand	手	[illegible]
R	Mouth	口	
X	Difficult		
Z	New		

(c)

Group: Shape Similarity

Category	Name	Standard Radical	Radicals/ Token-radicals
S	Flanking open	尸	[illegible]
T	Abreast balance	廿	[illegible]
U	U shape	山	[illegible]
V	Twisting shape	女	[illegible]
W	Square	田	[illegible]
Y	Y shape	卜	[illegible]

(d)

Figure 3.3 Classification of radicals *mouth*. The **shape similarity** group has 6 categories, called *flanking open*, *abreast balance*, *u shape*, *twisting shape*, *square*, and *y shape*. Figure 3.3 shows these groups.

In this method, rules of dividing a character into its radicals or token-radicals are followed the basic rules of strokes order shown in Figure 1.3 of Chapter 1.

Figure 3.4 gives two examples to show how the method works.

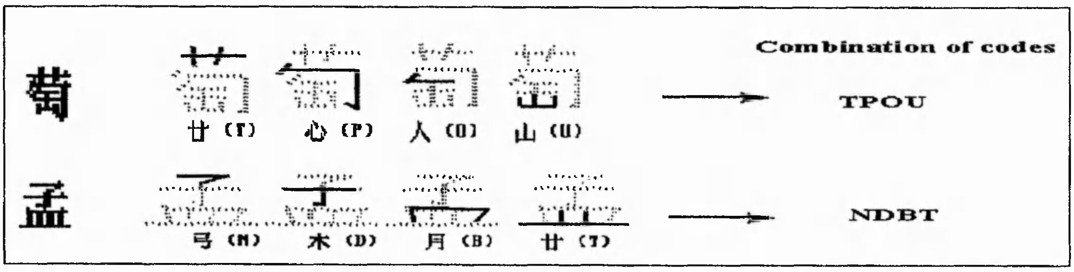


Figure 3.4 Examples of the *Cang-Jie* method

3.2.3 Three-Layer Hierarchy

In the three-layer hierarchy method, the structure of Chinese characters is represented in three layers: character, radical and stroke [Ren96b]. Basic strokes in the bottom layer are treated as indexes to determine the shape of radicals in a character. Radicals in the second layer are used to deconstruct the internal topological structure of a character in order to reduce the number of characters to be learnt by computer. Characters in the top layer are recognised by restructuring radicals into a chain code, and verifying it by means of a code database. Based on this method, the process of recognising a character is

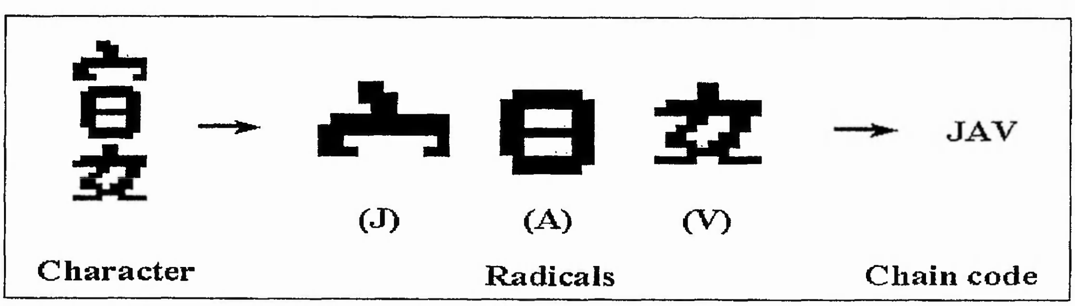


Figure 3.5 An example proceeded by the three-layer hierarchy

carried out in sequence of character, radical and chain code. Figure 3.5 gives an example for illustration of the method.

The structure representation in the three-layer hierarchy centres on the relationship of objects in these layers in two aspects: extraction and classification. Extraction takes into account the relationship of objects in their shape alteration between layers, which is also called vertical alteration. In the second aspect, classification generalises the relationship of objects in the same layer and forms different categories of these objects, which is termed horizontal classification.

i) Character

The analysis of characters is based on their topological structure embedded within the 2-D pictorial format. According to the investigation of characters in common use, 90% of characters are a combination of several components [Scu91]. The remnant is radicals existing as independent characters. These components come from original characters in the historical development of Chinese characters. Some of the original characters still keep as individual characters and some of them degenerate as only components of a character. In current Chinese dictionaries, most of these original characters are termed radicals.

A character can be seen as a pattern with the appearance of rectangle or square. The appearance is basically determined by the shape of basic strokes and their combination (It is a reason for calling Chinese characters '*square words*' (a nickname), sometimes).

ii) Radical

The radical layer consists of two parts: radical and token-radical. A radical is an original pictograph in Chinese characters. It may inherit its individual meaning and occupy a completely independent position in a character. A token-radical is developed from such radicals that are further simplified to become ultimately simple combinations of basic strokes. Although some of these are overlaid or crossed in a character, token-radicals common to many characters might be classified into a certain category of radicals in terms of their abstract concepts and shape structures. In the three-layer hierarchy, radicals and token-radicals are classified into 26 different categories (24 for radicals and 2 for special cases) alongside the categories in the *Cang-Jie* method.

iii) Stroke

A stroke is defined as a dot or a continuous line, shown in Figure 2.1 (c) in Chapter 2. Although radicals form the foundation of extraction and classification of the hierarchy, the structure of their shape has to be defined entirely by these primitive strokes. For instance, a combination of horizontal and vertical strokes can form a token-radical, or even a radical, as shown in Figure 3.6. Therefore, the structural representation in this layer will concentrate on these combinations as well as strokes themselves.

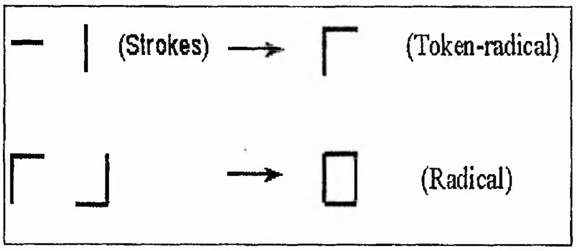


Figure 3.6 Combinations of strokes

The basic rules of stroke order shown in Figure 1.3 in Chapter 1 are developed as the essential rules for the structural representation of a character.

3.3 Knowledge Representation Techniques

In knowledge representation techniques, heuristics and common knowledge are used to specify uncertain and compressed information, as they are input data [Zim87]. In order to further understand features of such techniques, a comparison among statistical (decision-theoretic), syntactic (structural), fuzzy logic, fuzzy syntactic analysis and fuzzy possibilistic reasoning is made.

3.3.1 Statistical Method

In the **statistical** method, a set of characteristic measurement values called features are extracted from pattern data by using each n-sample data set represented as a point in a n-dimensional feature space [Gon78]. The method is ideally suitable for applications where patterns can be meaningfully represented in a clustered feature space. However, its drawbacks include: (a) that the method becomes difficult if the feature space does not cluster well; (b) that it fails if usable transformation can not be found; and (c) that it tends to be unsuitable for highly structured patterns [Ham92].

3.3.2 Syntactic Method

The **syntactic** method uses a set of rules to describe the structure of a pattern class in terms of primitive features of the pattern. These rules are formalised into grammars as the basis of a recogniser or a parser. A grammar is a formalised set of rules that describe the structure of a pattern class. The advantage of the method is that it is possible to recognise highly structured and complex patterns that are not suitable for the statistical method because the nature of the grammars and flexible structure of a class permit variation within each class. Its disadvantages include that (a) its grammars and recognisers are too complex and (b) rules are especially true when they are dealing with noise [Ham92].

3.3.3 Fuzzy Logic

The **fuzzy logic** approach is the logic underlying forms of reasoning which are approximate rather than exact. It provides a computational framework for knowledge representation and inference in an environment of uncertainty and imprecision. In such an environment, the fuzzy logic is effective when the solution needs not to be precise and/or it is acceptable for the conclusion to have a disposition rather than categorical validity. In this approach, exact reasoning is viewed as a limiting case of approximate reasoning; everything is a matter of degree. Knowledge is interpreted as a collection of elastic or, equivalently, fuzzy constraints on a collection of variables. Inference is viewed as a process of propagation of elastic constraints [Zad92].

Compared with the traditional logic, fuzzy logic has differences in some principal definitions. Probability, in classical logic, is numerical or internal-valued. However, it has additional options of employing linguistic meanings in fuzzy logic. Such probabilities may be interpreted as fuzzy numbers, which may be manipulated through the use of fuzzy arithmetic. The concept of possibility in fuzzy logic is graded rather than bivalent in classical logic. A possibility distribution is a fuzzy set of all possible values [Zad92].

Fuzzy logic has some advantages of simplifying knowledge acquisition and representation only by a few valuable rules, making it similar to the way humans think, used in linguistic or not numerical variables and solving previously unsolved problems [Mcn94]. However, it has some drawbacks of (a) being difficult or impossible to model these rules; (b) rules being defined by human experts; (c) using human observation as input or as the basis for rules; and (d) rules possessing features of natural vagueness.

3.3.4 Fuzzy Syntactic Method

The fuzzy syntactic method applies the syntactic approach to fuzzy logic for increasing the descriptive power of syntactic recognition by fuzzifying the concepts of a grammar and an associated language [Pal92]. This can be done by fuzzifying the primitives involved (i.e., the primitives become labels of fuzzy sets) or the production rules of the grammar and also the language defined by the grammar (i.e., the language becomes a fuzzy set of strings formed by symbols from the terminal vocabulary) [Kli95].

Applying the fuzzy syntactic method to Chinese character recognition is a potential method for representing the structure of characters [Ren96a]. According to the definition of the five-level structure, the structure of a radical is based on the relationship between token-radicals and strokes. The description of a radical can be grammatically defined as a fuzzy mathematical set with some attributes as the following expression.

Radical = {Token-radicals, Strokes, Relationship of the token-radicals and strokes, Description},

i.e. R = {T, S, RS, P, N};

Where:

R stands for a fuzzy set of a radical;

T represents a fuzzy subset of token-radicals that may consist of the radical;

S is a fuzzy subset of strokes that are parts of the radical but cannot be represented by any token-radical;

RS expresses a structural relationship between these token-radicals or between a token-radical and strokes;

P stands for a fuzzy subset, as a description, of the position of the token-radical and strokes in a radical;

N is a subset, as a description, of the categories to which the radical, token-radical and strokes belong.

Figure 3.7 shows an example of a radical set.

```
月  (moon)

R = {T, S, RS, P, N}
T = {t1}
    t1 = {冂}
S = {s1, s2}
    s1 = {一}; s2 = {一}
RS= {rs1, rs2, rs3, rs4, rs5}
    rs1 = {t1 connects s1},   rs2 = {t1 connects s2};
    rs3 = {t1 is around s1}, rs4 = {t1 is around s2};
    rs5 = {s1 and s2 are independent}
P = {p1, p2}
    p1 = {s1 and s2 are in the inside of t1};
    p2 = {s1 is on the top of s2}
N = {n1, n2, n3, n4}
    n1 = {R  ∈  'moon' radical category};
    n2 = {t1 ∈  token-radicals in the 'moon' radical category};
    n3 = {s1 ∈  basic strokes};
    n4 = {s2 ∈  basic strokes}
```

Figure 3.7 A radical fuzzy set

A character in the five-level structure is defined as:

Character = {Radicals, Token-radicals, Relationship of the radicals and token-radicals, Description},

i.e. C = {R, T, RS, P, O, D};

Where:

C is a fuzzy set of a character;

R is a fuzzy subset of radicals made of the character;

T represents a fuzzy subset of token-radicals that are parts of the character but cannot be represented by any radical;

RS expresses a structural relationship between these radicals and token-radicals;

P stands for a fuzzy subset of positions of the radicals and token-radicals in a character;

O is a fuzzy subset of an omitted part in a character;

D is a fuzzy subset of parts to be difficult described in the character.

Figure 3.8 gives an example for illustrating a fuzzy set of a character.

```
    朋 (friend)

C = {R, RS, P}
R = {r1, r2}
     r1 = {月}; r2 = {月}
RS = {rs1}
     rs1 = {r1 and r2 are independent}
P = {p1, p2}
     p1 = {r1 is on the left of r2};
     p2 = {r1 and r2 are side by side}
```

Figure 3.8 A character fuzzy set

The representation scheme shown above is a fair description for the structure of some Chinese characters. However, there is a problem about definition and matching of primitives. Primitives in the method can be defined as basic strokes and token-radicals for matching up parts of a character or a radical with these grammatical rules. Unfortunately, these basic strokes and token-radicals are either varied or overlaid in a character so that the rules become very difficult to match up as examples as shown in Figure 3.9.

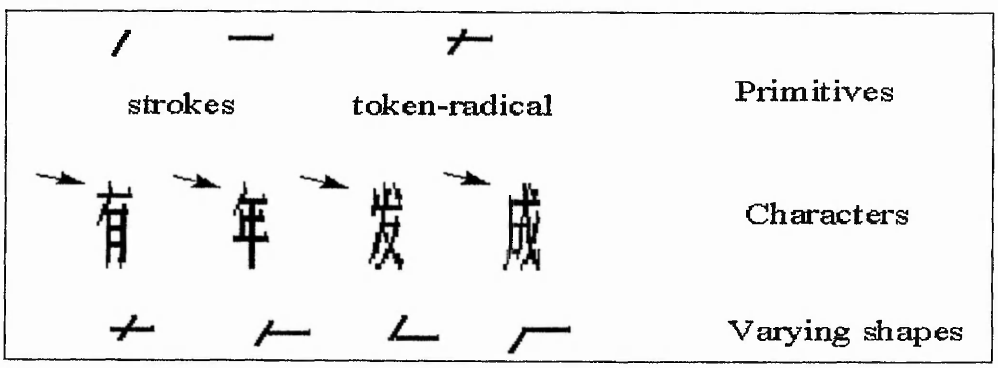


Figure 3.9 Varying Y shapes

Many researchers working on Chinese character recognition have concentrated on this problem by using the different techniques reviewed earlier in the two-layer hierarchy in Section 2.3.2 of Chapter 2.

3.3.5 Fuzzy Possibilistic Reasoning

Fuzzy possibilistic reasoning in knowledge representation approaches is well suited to dealing with imperfect, uncertain and vague information [Kru94]. Reducing the complexity of imperfect information is achieved by information-compressed representations based on if-then rules. These rules are interpreted as logical implications and are termed as possibilistic inference rules defined by the notation $\mathfrak{R}$.

i) Concepts

Based on the approximate reasoning and probability theory, fuzzy possibilistic representation uses conjunctly combined rules to validate a possible resolution from various restrictions. In these if-then rules, antecedent in the *IF* clause and consequence in the *THEN* clause are constrained by their possibility distributions denoted by π. The possibility distributions are related with the interpretation of vague concepts as contour functions of random sets. Physical quantities of the distributions are defined by the possibility measures denoted by Poss_{π}.

ii) Expression

Generally, a possibilistic inference rule $\mathfrak{R}_j$ can be expressed by

$$\mathfrak{R}_j : \textbf{IF } \xi^{S}_{j} \textbf{ is } \mu_j \textbf{ THEN } \xi^{T}_{j} \textbf{ is } v_j, \qquad j = 1, \ldots, r,$$

or

$$\mathfrak{R}_j : \textbf{IF } \xi^{S}_{j}{}^{(1)} \textbf{ is } \mu_j{}^{(1)} \textbf{ AND } \xi^{S}_{j}{}^{(2)} \textbf{ is } \mu_j{}^{(2)} \textbf{ THEN } \xi^{T}_{j} \textbf{ is } v_j, \qquad j = 1, \ldots, r,$$

where μ_j, $\mu_j^{(1)}$, $\mu_j^{(2)}$ and v_j are subsets of possibility distributions on the space sets S_j and T_j with regard to j. ξ is a variable whose values can be arbitrary possibility distributions on S_j or T_j. The symbol **is**, appearing in possibilistic inference rules, serves as a linguistic description of the operator $\subseteq$ and is therefore to be interpreted as 'is at least as specific as'.

The relation $\Re$ of all rules is

$$\Re = \bigcap_{j=1}^{r} \Re_j$$

iii) Possibility and Probability

There are similarities and differences between probability theory and possibility theory [Kli95]. The two theories are similar in the sense that both are subsumed not only under fuzzy measure theory, but also under the more restricted evidence theory. The differences in mathematical properties of the two theories make each theory suitable for modelling certain types of uncertainty and less suitable for modelling other types. As is well known, for instance, probability theory is an ideal tool for formalising uncertainty in situations where class frequencies are known or where evidence is based on outcomes of a sufficiently long series of independent random experiments. Possibility theory, on the other hand, is ideal for formalising incomplete information expressed in terms of fuzzy propositions.

3.4 Artificial Neural Networks

Neural networks are developed on the basis of combination of mathematical foundation, inherent parallelism, knowledge store [Day90], fault tolerance, adaptability [Nel91], and simulating behaviour and brains of human in learning and training. Some mathematical equations affect inputs, memory, recall, determining energy levels, convergence, and stability of the networks.

3.4.1 Neocognitron Neural Network

The Neocognitron network, proposed by Fukushima et al. [Fuk82] in 1982 and evolved from an earlier model called the *cognitron*, is able to recognise patterns in different sizes [Fre92] regardless of where they are placed in the field. Moreover, the network has a high degree of tolerance to distortion of patterns in terms of their features [Fuk88].

i) Architecture

Essentially, the Neocognitron architecture is a fixed-architecture network with thousands of neurones and hundreds of thousands of connections [Woo96]. Some of them are fixed. Nodes of neurones are organised into layers, which (after the input layer) alternate between S-layers and C-Layers (the prefixes S and C stand for *simple* and *complex* and derive from biological cells). The layers are also divided into planes (subgroups) called S-planes and C-planes.

The neurones themselves are called S-cells and C-cells arranged in a two-dimensional array. S-cells are for extracting features. C-cells are inserted in the network to tolerate errors of the features extracted by the S-cells. The layer of C cells at the highest stage is the recognition layer representing the final result of the pattern recognition. There are also V-cells that support S-cells (subsidiary inhibitory) and occur in single planes per S-layer.

Each S-plane is trained to respond to a particular feature. The S-planes in the lower layers are trained features at the lower level. Two planes in the same layer are trained to respond to similar but distinct features. Each node in a given plane picks out the same feature though in a different position. Each S-node is connected to a window of neurones in each of the immediately preceding C-planes.

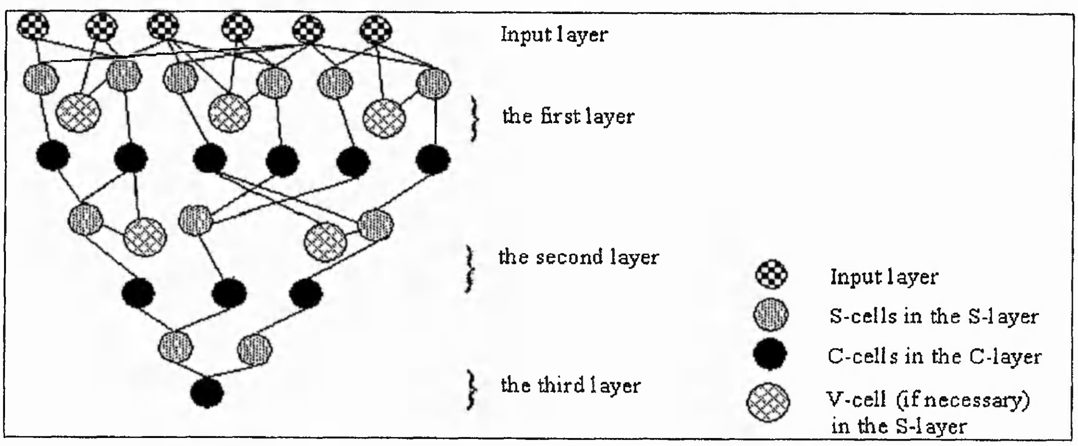


Figure 3.10 Neocognitron architecture

Weights of the network are variable and shared between S-cells in the same plane.

Connections from S-cells to C-cells are fixed and invariable. The output of an S-cell is an effective measure of similarity between the input and corresponding weights. Figure 3.10 illustrates this architecture.

An example is given in Figure 3.11 for showing how S-cells and C-cells of the network recognise a radical. The notation U_{SI} and U_{CI} are used to indicate the layers of S-cell and C-cells at the Ith stage, respectively. The input layer is denoted by U_0.

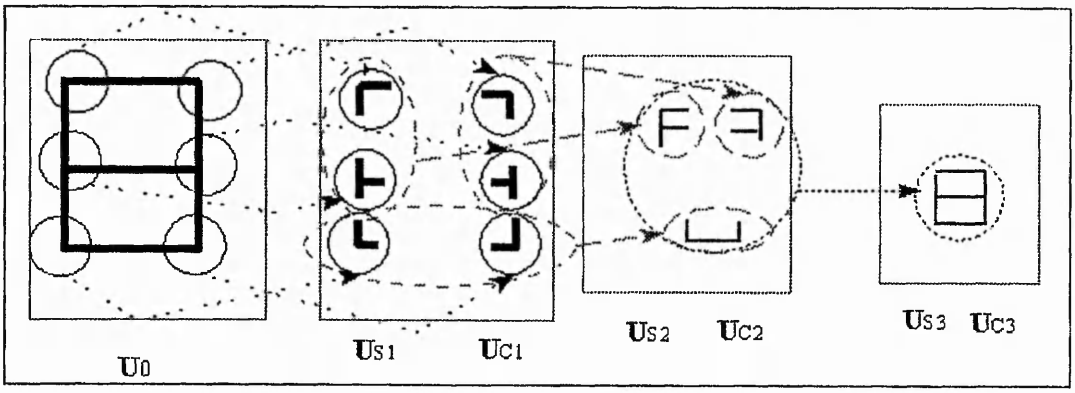


Figure 3.11 Illustration of S-cells and C-cells for recognising a radical

ii) Algorithms

The fundamental mathematical formulae for supporting S-cells, C-cells and V-cells of the Neocognitron network to learn and train are listed as follows [Fre92].

The output of an S-cell in layer U_{SI} is given by

$$U_{SI}(n,k)=r_I(k).\varphi\left[\frac{x}{y}-1\right] \qquad \cdots\cdots \qquad (3.1)$$

Where,

n is a two dimensional set of co-ordinate indicating the position of cell's receptive field centre in the input layer U_0;

k is a serial number of the cell plane, $1 \le k \le K_{SI}$ for S-cells; $1 \le k \le K_{CI}$ for C-cells; $\mathbf{K_{SI}}$ and $\mathbf{K_{CI}}$ are values in layers;

$\mathbf{r_I}$ **(k)** determines the efficiency of the inhibitory input to this S-cell and controls the selectivity in feature extraction.

$$x = 1 + \sum_{k=1}^{K_{CI-1}} \sum_{v \in A_I} a_I(v, k, K) \cdot U_{CI-1}(n + v, k)$$

$$y = 1 + r_I(k) \{1 + r_I(k)\}^{-1} \cdot b_I(k) \cdot u_{VI}(n) \quad \cdots\cdots \quad (3.2)$$

$$\varphi[x] = \begin{cases} x & if \ x \geq 0 \\ 0 & if \ x < 0 \end{cases}$$

Where,

α_I **(v, k, K)** and $\mathbf{b_I}$ (k) represent the excitatory and inhibitory interconnecting coefficients, respectively;

U $_{\mathbf{VI}}$ **(n)** is the excitatory inputs;

v is the inhibitory input;

$\mathbf{U_{CI-1}}$ **(n + v, k)** stands for U_0 (n) when I =1;

The V-cell, which sends an inhibitory signal to the S-cell, yields an output p to the weight that is expressed by root mean square of the inputs from the preceding c-cells. The output U_{VI} is:

$$U_{VI}(n) = \left[\sum_{k=1}^{K_{CI-1}} \sum_{v \in A_I} C_I(v) \cdot \{U_{CI-1}(n + v, k)\}^2 \right]^{1/2} \quad \cdots\cdots \quad (3.3)$$

Where,

$\mathbf{C_I}$ **(v)** (≥0) represents the strength of the fixed excitatory connections, and is a monotonically decreasing function of |v|.

The output of a C-cell in the layer U_{CI} at the position n is given by

$$U_{CI}(n,k) = \psi\left[\sum_{k=1}^{K_{SI}} j_I(k, K) \cdot \sum_{v \in D_I} d_I(v) \cdot U_{SI}(n + v, k) \right] \quad \cdots\cdots \quad (3.4)$$

Where,

$$\psi[x] = \frac{\varphi[x]}{1+\varphi[x]} \qquad \ldots\ldots \qquad (3.5)$$

d_l (v) is the weight on the connection from the S-cell at position ν in the receptive field of the C-cell;

j_l (k, K) is the condition of joining for several S-planes **k**, sometimes joined together and made to converge to a single C-plane **K**;

ψ [] is a function specifying the characteristic of saturation of the C-cell.

iii) Evaluation

The Neocognitron network can be trained far more efficiently than using conventional neural network, such as back-propagation because weight sharing in Neocognitron is again constantly enforced and partially invariant under translation [Woo96]. Besides, the network has advantages of dealing with position and size variance, deformed patterns without any normalisation processing [Fuk91].

However, it has the following weaknesses (a) complex architecture and processing equations; (b) difficult to state to what degree the network can cope with deformed patterns because there is no appropriate maths measure to express the psychological feeling of the deformation [Min90]; (c) confusion between similar patterns generally increasing with the number of patterns to be recognised; and (d) discrimination between similar patterns in different categories relying on a skilful choice of training patterns.

3.4.2 Associative Memory Neural Networks

The associative memory neural networks, **Bi-directional associative memory** (BAM) introduced by B. Kosko in 1985 [Kos87] and **Hopfield memory** introduced by J. Hopfield in 1982 [Hop82], are able to recognise an incomplete pattern with their associative memory. The network architecture can be built up with neurones and connectivity on one layer or more layers.

i) Algorithms

The mathematical formula for the associative memory function is established on the construction of an energy equation E [Hop82] [Kos87] [Kos88], called the Steepest Gradient Descent algorithm:

$$E = -\sum_i \sum_j X_j W_{ij} Y_j + \sum_i \theta_i X_i + \sum_j \varphi_j Y_j \quad \cdots\cdots \quad (3.6)$$

Where, θ_i and φ_j are constants of the energy equation E.

The algorithm contains two phases: learning and training. In the learning phase, the associative memory function is used to form the connectivity matrix W for training a set of input patterns X_i (u) and output patterns Y_j (u), where u = 1, 2 ... M; i, j = 1, 2 ... N, the weight W (i, j) is determined by the Hebbian rules:

$$W(i,j) = \begin{cases} \sum_u X_i(u) Y_j(u) & if \quad i \neq j \\ 0 & if \quad i = j \end{cases} \quad \cdots\cdots \quad (3.7)$$

In the training phase, the algorithm aids convergence because its value in equation (3.6) is either reduced or to remain constant during the recall procedure [Wan94], providing the following conditions are satisfied.

$$Y_j^{n+1} = \begin{cases} 1 & \sum_i W_{ij} X_i^n - \varphi_j > 0 \\ Y_j^n & \sum_i W_{ij} X_i^n - \varphi_j = 0 \\ -1 & \sum_i W_{ij} X_i^n - \varphi_j < 0 \end{cases}$$

$$X_i^{n+1} = \begin{cases} 1 & \sum_j W_{ij} Y_j^n - \theta_i > 0 \\ X_i^n & \sum_j W_{ij} Y_j^n - \theta_i = 0 \\ -1 & \sum_j W_{ij} Y_j^n - \theta_i < 0 \end{cases} \quad \cdots\cdots \quad (3.8)$$

When a portion of the original pattern is used as a retrieval cue, the algorithm is denoted as auto-associative memory. When the desired output is different from the input, the algorithm is called hetero-associative. The static method requires that the network is trained in advance with specific patterns, and retrieves them from portions or distorted inputs. When the learning and retrieval are embedded in the training process randomly, it is defined as a 'dynamic' method.

ii) Evaluation

The network with the associative function has recurrence (closed loops) and non-linearity, i.e. positive weights are excitatory and will strengthen connectivity; and negative weights are inhibitory and weaken connectivity [Nel91].

It has the advantage of eliminating noise to recognise a pattern from its incomplete version. Compared with the back-propagation network, which has advantages in storing many more patterns than the number of dimensions and learning a large variety of pattern mapping relationship [Wan93] but with no guarantee of convergence [Nel91], the associative memory network can converge to a stable point even though it may be slow sometimes.

Its disadvantages lie mainly in the limitations of associative memory and convergence to a local solution rather than a global minimum [Day90]. In addition, the memory, defined as M and used for evaluating the capability of associated with patterns, is limited as a constant value as shown in equation (3.9) while the number of learning patterns n increases to a big value. For instance, M is equal to 25 when n is 1000.

$$M = \frac{n}{4\log_2 n} \qquad \cdots\cdots \qquad (3.9)$$

3.5 Chain Code Method

The chain code method is able to translate radicals into letters to restructure them into a chain code instead of a character for verification. The format of a chain code consists of fewer than or 5 English letters from A-Z into a one-dimension order. Figure 3.12 gives an example of the format.

While a radical is encoded from a pictorial format to an English letter, as shown in Figure 3.13, the letter only stands for a category label to which the radical belongs. In a sense, the final intention of recognising a radical has to find the classification of the radical rather than what exactly the radical is. The intention can be considered as the major feature of topological structure of Chinese characters, i.e. a character is a combination of several classifications of radicals.

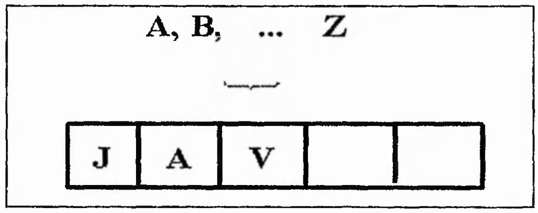


Figure 3.12 An example of chain codes

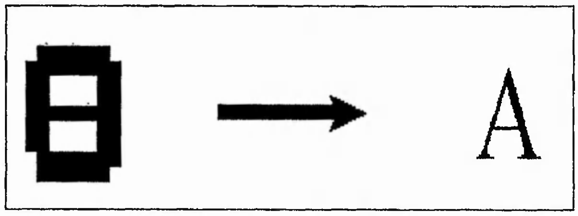


Figure 3.13 Format of radicals

The advantages of using a chain code substitute are: (a) restructuring several radicals into a character without considering their shape, size and position in a 2-D picture; and (b) confirming a character in a database using its chain code, i.e. a combination of several letters standing for a character, rather than a character in a bitmap or other different fonts.

Its disadvantage is the fact that a letter is an unrelated symbol of a category set rather than its member. If a mis-recognised case happens, the only solution for recovering the symbol is to refer to its neighbours in its chain code for correction, or by a guessing process. In this case, even expert knowledge will not have been of much help. Therefore, an ambiguous solution will be possible even if the relation between letters in the chain code is very tight.

3.6 Analysis of Error Probability

Two methods can be used in the error assessment of a chain code or a character:

contextual information and analysis of error probability. In general, recognition deals with only a single character. The method of **contextual information** refers to other characters in context to detect errors or even to correct them [Imp91]. The contextual information, for instance, can be a phrase or a sentence. A dictionary or grammar check will be available in the method.

The **analysis of error probability** uses some statistical outcomes obtained from the recognition process to detect errors. Applying human expert knowledge forms cues for correction and rejection of errors. The analysis method focuses more on a single character and its quality produced by the recognition process for the assessment of errors.

Chapter 4

Software Development of Chinese Character Recognition System (CCRS)

4.1 Introduction
4.2 Analysis, Design, and Implementation of CCRS
4.3 Preprocessing
4.4 Normalisation
4.5 Recognition
4.6 Restructuring
4.7 Postprocessing

4.1 Introduction

Based on the application of the three-layer hierarchy method, a system called Chinese Character Recognition System (CCRS) has been developed for off-line recognition of Chinese characters. The development of the system includes three phases: analysis, design and implementation

The analysis of the system includes problem definition, dataflow diagrams, functions, and data analysis. The main problems encountered in the application of the three-layer hierarchy method are discussed. Dataflow diagrams of function and data analysis are produced for the design of the system.

The system design includes determination of framework, data model and program design. It consists of a main system and five subsystems: Preprocessing, Normalisation, Recognition, Restructuring and Postprocessing. Two further subsystems are added to process and maintain the CCRS.

The system implementation focuses on integration of models, programming, performing methods, and maintaining the system. The development of the system also includes software application and monitoring strategy.

4.2 Analysis, Design, and Implementation of CCRS

The analysis and design of CCRS are based on the application of the Structured Systems Analysis and Design Method (SSADM). SSADM is the UK government structural and procedural standards for carrying out systems analysis and the design of an information technology system [Dow88]. Advantages for using SSADM include description of how a project is organised, the order in which the many jobs are to be done, and the structural description is to be comprehensive and logically coherent.

4.2.1 System Analysis

The system analysis aims to transform the objectives of a project into a structured specification [You89]. Its methods include conventional, classical, structured, and marriage of structured and artificial intelligence. Conventional analysis was used before the mid-1970s by narrative novel specifications that were hard to understand and virtually impossible to maintain.

The classical analysis was applied from the mid-1970s to the mid-1980s by early versions of graphical models and had the emphasis on modelling the current system before a new one. Its disadvantage was a fuzzy, poorly defined distinction between physical and logical models.

The structured system analysis was developed from the end of the 1980s and focused on giving structural and procedural standards of the system. The standards include the structural intercommunication, diagrams, and design of physical data [Dow88]. The structured analysis helps to break down a project into smaller phases that are easier for design, and to integrate all phases or tasks into a structured system for its performance.

The combination of structured analysis with artificial intelligence (AI) was formed in the early 1990s. Because AI aims at imitating human performance in a wide variety of tasks, combination of the structured analysis with AI gives benefits on two aspects: structural analysis can be helpful in the process of building up an AI system, and AI system technology can enhance the structured analysis in respect of simulating human expert knowledge.

The combined analysis has been applied to the CCRS in four phases: problem definition, dataflow diagrams, function analysis, and data analysis. Three major problems and several secondary problems of the CCRS are analysed. Using the technique of dataflow diagrams, a structural model that includes functional and data analysis is designed to form the fundamental architecture of the CCRS. The function analysis defines specifications of the implementation. The data analysis concerns date changes on their quantity and formats.

i) Problem Definition

The problem definition ought to determine the difficulties associated with acquiring pertinent information, complexity to be handled, and accommodating changes. Problem domains mean partitioning large and complex problems into small parts that can be easily understood. Communications between these parts have to match up with the logical and physical views of software requirements.

According to the three-layer hierarchy method, a character is identified using a chain code that is restructured from recognised radicals [Ald95]. Hence, the main problem domains of CCRS are defined as extraction, recognition and classification. Secondary problems include normalisation, restructuring, organisation and search for a chain code database, and verification.

Communications between these domains determine the relations between three layers of character-radical-stroke as shown in Figure 4.1. Within the extraction domain, the relation between a character and radicals is relied on the position and order of the radicals, and between a radical and strokes is on the shape of the radical.

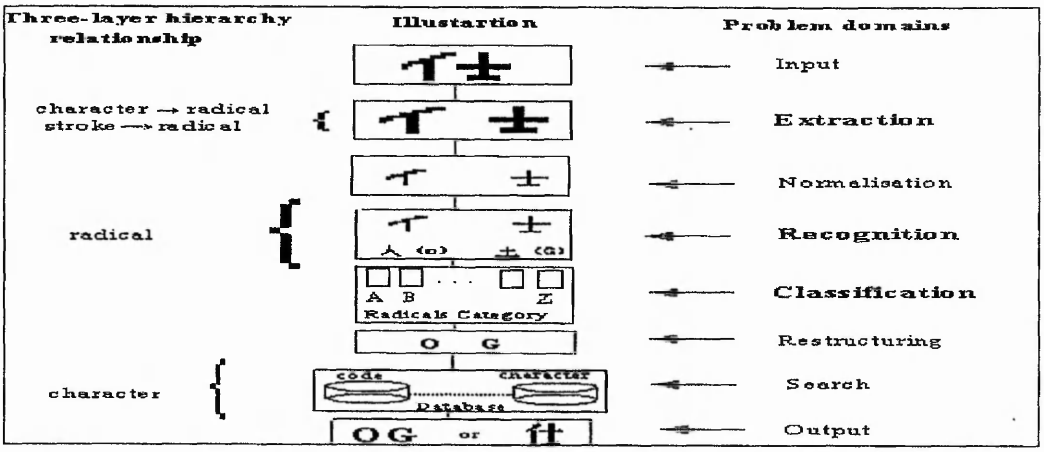


Figure 4.1 The main and secondary problems in CCRS

ii) Dataflow Diagrams

The dataflow diagram is a graphical technique of depicting information flow and transforming data from input to output. The technique has features of illustrating data flowing around the system, processing requirements, and communications between processors [You89]. Based on Figure 4.1, the dataflow diagram of CCRS at the top level is displayed in Figure 4.2. Further dataflow diagram of the system is shown in Figure 4.3 in terms of breaking processors at the top level downwards from large aggregates to smaller pieces.

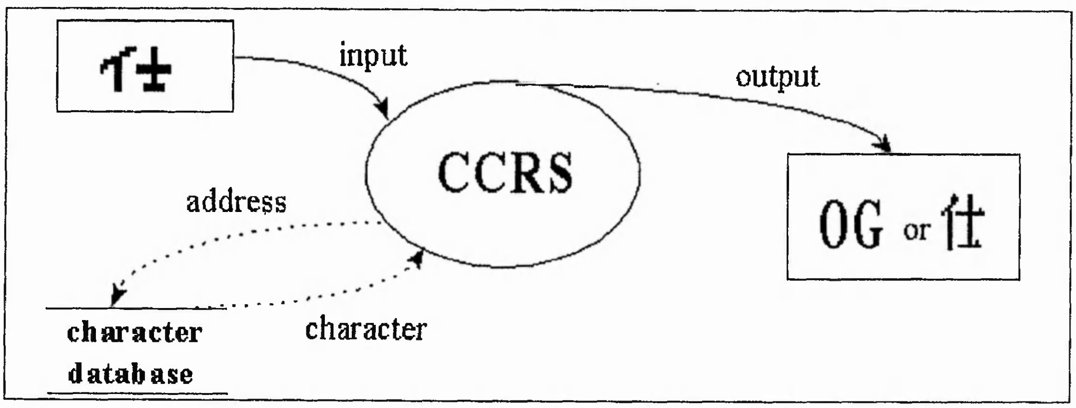


Figure 4.2 Dataflow diagram of CCRS at the top level

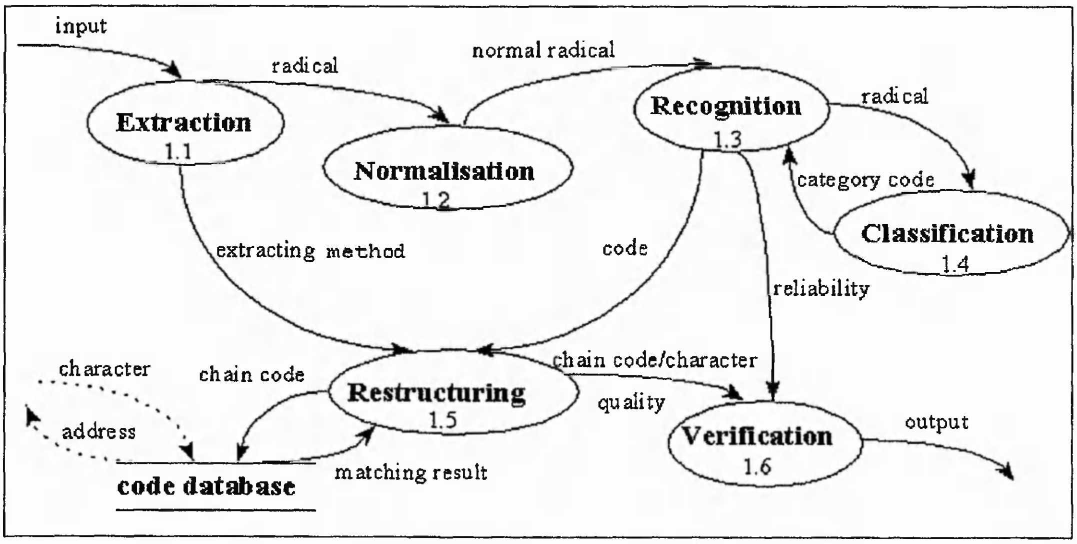


Figure 4.3 Further dataflow diagram of CCRS

iii) Function Analysis

Generally, the logical structure in function analysis of a system is for mapping problem domains into functions and subfunctions of the system. According to the processors in Figure 4.3, five major functions of CCRS are determined as extraction, normalisation, recognition, restructuring and verification [Ren95b]. **Extraction** takes a radical from a character. **Normalisation** is to normalise a radical for passing it to recognition. **Recognition** includes classification of radicals, identifying a radical, and providing a code of a radical. **Restructuring** contains three subfunctions: organisation, formation and search. The organisation enables to build up a code database. The formation combines codes of radicals in a chain code. The search is able to match up the chain code with one in a code database. **Verification** possesses the capability of detecting, correcting and rejecting error in a chain code. Figure 4.4 shows the function analysis diagram.

iv) Data Analysis

The data analysis enables further validation of a system in the way of composting its data structure. The major methods are data flow, data structure, and language-based. The data flow method focuses on an assessment of how data moves through the information domains and provides isolating data content and processing function. The data structure

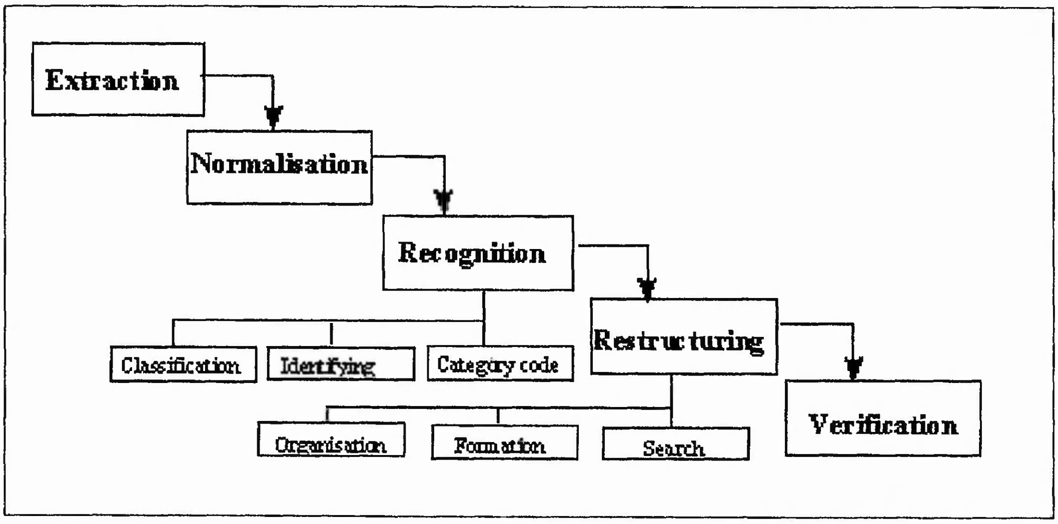


Figure 4.4 Function analysis diagram

method identifies information items, actions (processes) and models in terms of the information hierarchy. The language-based method applies a formal specification language to model data, which enables automated processing to uncover inconsistency, omissions, and other errors of the system [Pre87].

The data structure method used in the CCRS focuses on its 2-D features in the three layers. Data in CCRS, whether a character, radical or stroke, are written in dots being filled in a bitmap. This bitmap can be converted into digitised format or image representation shown in Figure 4.5 for processing convenience. The digitised format decreases the complexity of data structure. The image format is easily observed.

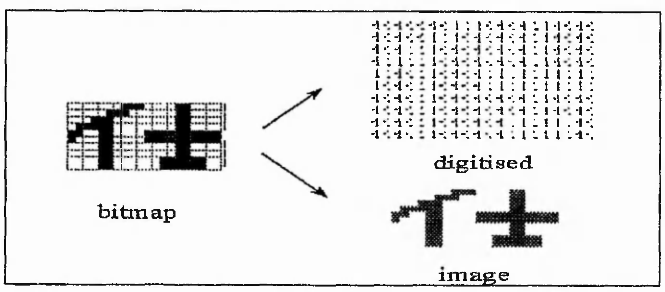


Figure 4.5 Data features of CCRS

4.2.2 Design Method

The system design concerns program modules and interactions of these modules that implement specifications of the system analysis [You89]. It includes (a) the logical design that transforms the logical specification into details of both data and processes of the system, and (b) the physical design that creates physical specifications, database and program specifications from the logical specification [Cut91].

The logical structure of the framework is organised as logical design of CCRS. The data model and program design are put into its physical design. The data model concerns the definition of different data formats through the whole system. The program design undertakes mapping functions of the system into the environment of computer performance.

i) Framework

In the design of the system framework, all functions are logically organised in the top-down integration approach [Pre87]. A main system and several subsystems are designed for co-operating with the overall recognising process. The main system takes responsibility on control of running every subsystem, monitoring time, and displaying outcomes of recognition. A subsystem is designed for carrying out only one function of CCRS. Additional subsystems provide services for CCRS. Figure 4.6 has shown the structure.

ii) Main System

The main system of CCRS performs these tasks: maintaining an input character, implementing subsystems and additional subsystems, monitoring performance of subsystems, giving statistics data and charts, and managing recognition of more characters (maximum 5) at the same time. Each task is defined as an individual option in processing.

iii) Subsystems

The five subsystems are termed Preprocessing, Normalisation, Recognition, Restructuring, and Postprocessing, for carrying out a part for recognition of Chinese characters.

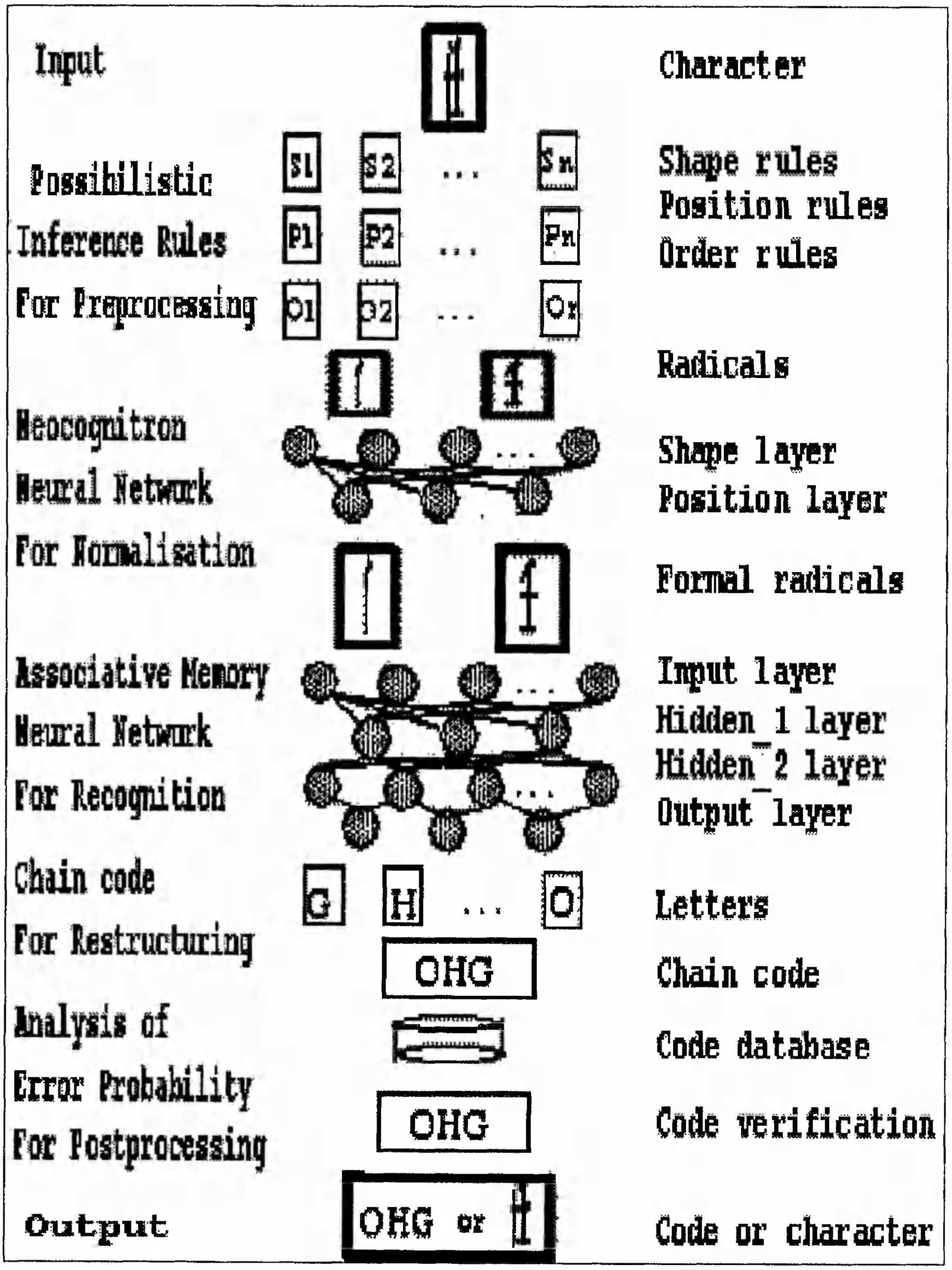


Figure 4.6 The structure of Chinese Character Recognition System (CCRS)

a) Preprocessing undertakes the extraction function, i.e. extracting radicals from an input character. Extracting techniques are mainly focused on the position, order and shape of radicals in a character.

b) Normalisation carries out normalising the position and shape of a radical after it has been deconstructed from a character.

c) Recognition deals with classification and recognition of a radical using two phases: learning and recognition. In the learning phase, all normal radicals are classified into different categories and learnt. Radicals passed from the Normalisation process are recognised prior to returning its code in the recognising phase.

d) Restructuring combines codes of radicals into a chain code in terms of their extracting rules in the Preprocessing and checks it using a code database. The organisation of the code database is one task of the subsystem as well.

e) Postprocessing is able to verify error in a chain code using strategies of detection, correction, and rejection cues.

iv) Other Subsystems

There are two other subsystems: Introduction and Management. **Introduction** subsystem gives a brief documentation to present the historical development of Chinese characters, the *Cang-Jie* method and the three-layer hierarchy, paradigm of CCRS process, and publications of the project. **Management** subsystem is used to maintain CCRS, for instance, profiles of programs, files, tools and modified records.

v) Data Model

The data model of CCRS is mainly applied to map the 2-D features of data into the environment of computer process. Four models are used to represent a Chinese character in different formats: bitmap, digitised, chain code and display. The **bitmap** model is a graphic model for coping with the input of Chinese characters. The **digitised** model uses digits, 1, -1 and 0 for dealing with processing characters. This model has merits of processing complex data using less memory. The **chain code** model employs a series of alphabetic characters abstractedly to stand for a Chinese character. The **display** model shows data on a screen for monitoring outcomes, demonstrating the final output and a standard character from an external database. Figure 4.7 illustrates the four data models.

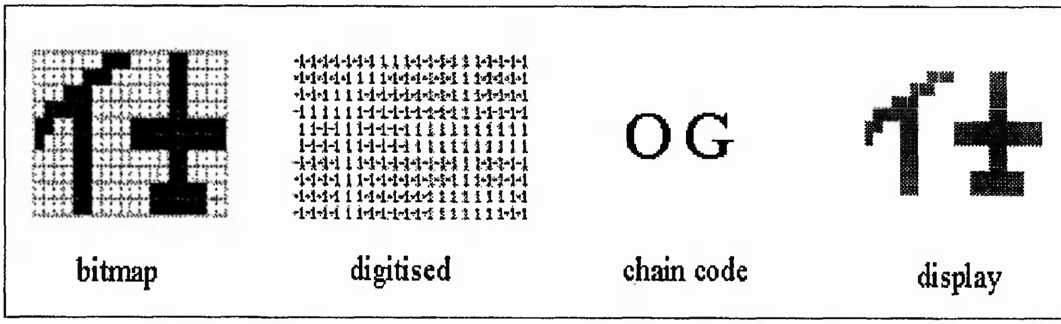


Figure 4.7 Four data models for representing a character

In addition, data in a code database are defined as a list of chain codes and their reference address. A chain code consists of less than five English characters from A to Z, which stand for categories of radicals in a character. A reference address is an address corresponding to a standard database of Chinese characters from outside of the system.

vi) Program Design

Program design transforms functions of the system into a structure chart of modules. In the chart, every module is indicated to what actions or requirements it should respond and in which communication relationship a module has been involved. Following the function analysis diagram (Figure 4.4), the structure chart of CCRS modules as program logical design is shown in Figure 4.8. In this figure, the **diamond** symbol stands for styles of executing modules; i.e. only one of these modules at a level can be processed at the same time. The sequence of executing the modules under the symbol is optional.

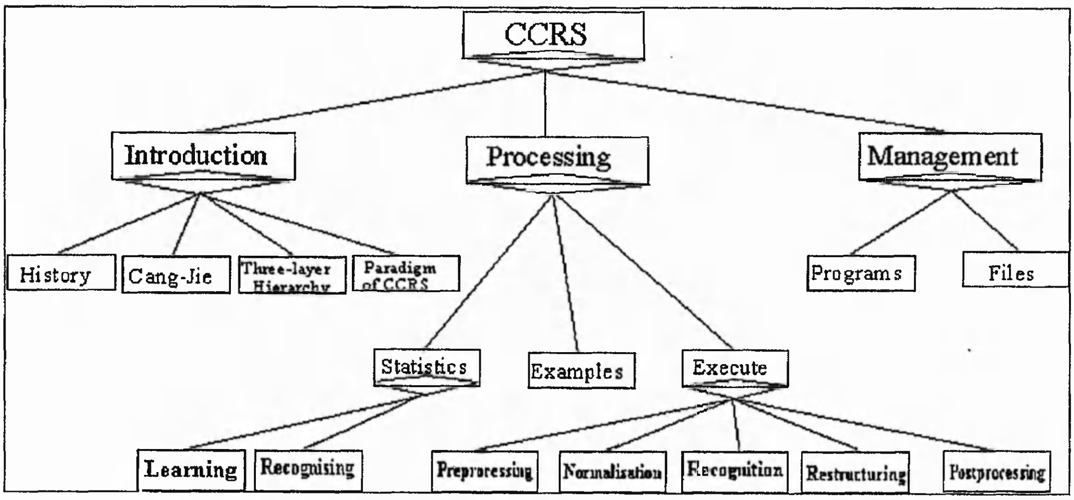


Figure 4.8 The structure chart of modules in CCRS

4.2.3 Implementation

The implementation is a process of translating the design specification into an operational system. This includes integration of models, programming, performing methods and maintaining the system [Cut91]. The strategy of CCRS implementation applies the top-down structure, i.e. inspecting a problem to be solved and refining it into sub-problems till each sub-problem is sufficiently well defined [Lei87]. Details of the CCRS implementation to be discussed centre on software application, implementing models, monitoring strategy and five subsystems.

i) Software Application

Software application applies knowledge of physical environment, tool and capability of software for supporting the system implementation. The implementation of CCRS is supported by the SunOS Release 4.1 workstation with UNIX operating systems. Techniques of the implementation include X windows, X view functions, C programming language, common user access (CUA) with programmable workstation window descriptions and standards, and a graphical package on Graphical Kernel System (GKS). More details of the techniques have been put in Appendix A: Support Tools.

ii) Implementing Models

CCRS has two alternative models for its implementation: separating and integrating.

a) The **separating** model implements every subsystem independently. In this model, testing data, various processing parameters and outcomes of a subsystem are examined. The model is also used for recognising a single character.

b) The **integrating** model is a complete implementation for integrating five subsystems of CCRS. The model provides a statistics utility for monitoring the whole process. Statistics charts give an illustration and comparison of different processing parameters in implementing these subsystems. In addition, the model can support recognition of less than five characters at the same time.

iii) Monitoring Strategy

The monitoring program collects parameters of process, variables and data; and also groups them together into a special package. Two monitoring strategies are used in CCRS for checking out and comparing with the processing results: statistics charts for monitoring the implementing time of CCRS, and viewing parameters for a guarantee of communications between these subsystems.

a) Statistics charts are produced by the monitoring program while CCRS is implemented in the integrating model for recognition of one or more characters. As a necessary tool to improve and validate CCRS, these graphical charts illustrate the recognising time for all subsystems.

b) Viewing parameters provide an opportunity for checking some important parameters reserved by the subsystems to ensure correct outcomes and communication between the subsystems. These parameters include status of running subsystems, processing variables and data, results and their qualities.

4.3 Preprocessing

The Preprocessing subsystem carries out the extraction of radicals from characters using fuzzy possibilistic inference rules to cope with the vagueness on the structure of characters [Ren97b]. Two representation methods are used: situation-rules to describe the position and extracting order of a radical, and shape-rules to deal with the shape domain of radicals, possibility distribution and measure techniques.

4.3.1 Vagueness of Radicals

According to the definition of the three-layer hierarchy method, radicals are seen as the basis of recognising a character. Essentially, a radical possesses pictographic features of uncertain position, shape and extracting order. The **position** means that a radical can be located at any place in a character, for instance, bottom, left or outside as shown in Figure 4.9 (a). A radical can be within a rectangle, square, u or y **shape** as illustrated in Figure 4.9 (b). The definition of shape is referred to Figure 3.3 in Section 3.2.2. The extracting **order** indicates the sequence of radicals extracted from a character. For instance, a radical located at the top of a character is extracted first, as examples shown in Figure 4.9 (c).

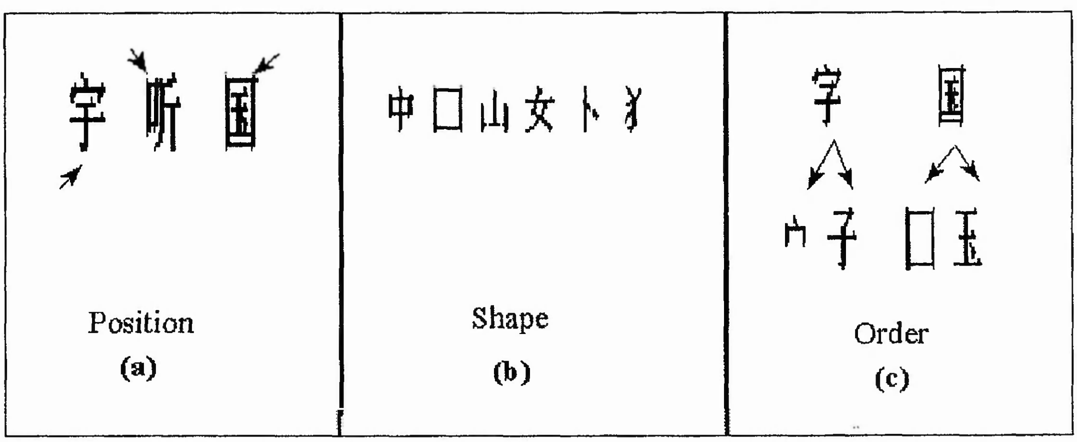


Figure 4.9 Vagueness of radicals

The relations between a character and radicals and between a radical and strokes are compressed and interconnected with each other. Basically, the relation between a character and radicals determines the position and extracting order of the radicals in the character. The relation between a radical and strokes decides the shape of the radical. In the real application, uncertain radicals in a character have compressed features of interdependence and mutual attraction. Figure 4.10 illustrates the features and their relations.

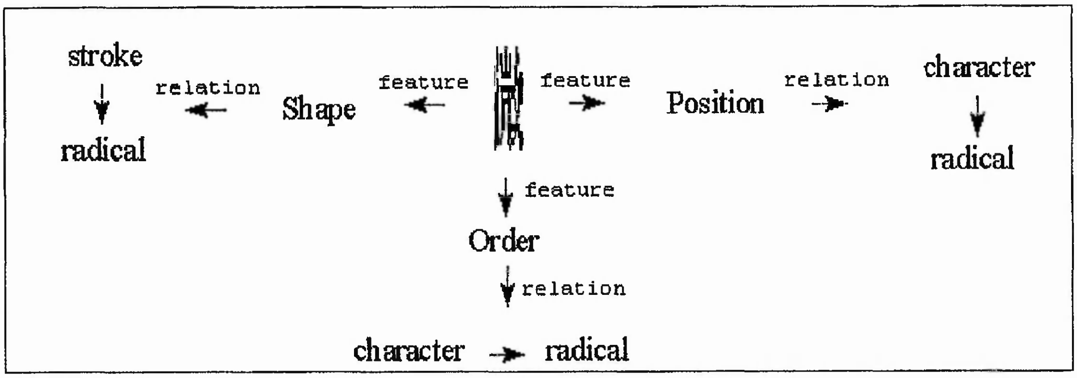


Figure 4.10 Compressed features of radicals in a character

4.3.2 Situation

The situation representation uses inference rules defined by the above interpretation for determining radicals in a character. The representation focuses descriptions on (a) the position of a radical in a character, and (b) the order of extracting a radical from a character.

According to the fuzzy possibilistic reasoning theory described in Section 3.3.5 in Chapter 3, let the notation P and O stand for two domains of the position of radicals in a character and the order of extracting a radical from a character respectively. Their possibility distributions can be defined as $\pi(P)$ and $\pi(O)$. The possibility measures are given by the notations $Poss_\pi$ (P) for $\pi(P)$, and $Poss_\pi$ (O) for $\pi(O)$. The possibilistic inference rules are represented by the notation $\Re^{(PO)}$. If p and o denote variables with the domains P and O respectively, the $Poss_\pi$ (p) is the possibility measure of p on $\pi(P)$; similarly, $Poss_\pi$ (o) for o on $\pi(O)$.

i) Position Variance

The investigation of position variance of radicals in a character is based on their features of a two-dimensional picture and a rectangular appearance, one of the major characteristics in the structure of Chinese characters. The domain of position variable is defined by

P = {***width, length***}.

Because a radical may keep an independent position in a character, the possibility distribution of position variance of a radical on the domain P, shown in Figure 4.11, is defined by

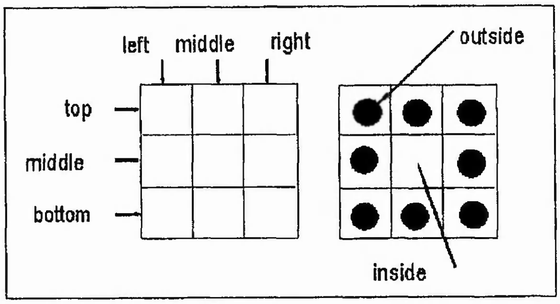


Figure 4.11 Possible position of a radical

π(P) = {***outside, inside, top, bottom, left, right, middle***}.

Poss$_\pi$ (P) for π(P) is defined by, for instance,

Poss$_\pi$ *(left)* = {***width*** $\leq$ ***2/3 width of P, length = length of P***}.

ii) Extraction Order

The extraction order indicates the sequence of radicals extracted from a character that might consist of two or more radicals. The domain of extraction order is expressed by

O = {***first, last***}.

The possibility distribution of extraction order on the domain O is represented by

π(O) = {***outside*** $\rightarrow$ ***inside, inside*** $\rightarrow$ ***outside, top*** $\rightarrow$ ***(middle*** $\rightarrow$ ***bottom), top*** $\rightarrow$ ***bottom, left*** $\rightarrow$ ***(middle*** $\rightarrow$ ***right), left*** $\rightarrow$ ***right***}.

The notation '$\rightarrow$' stands for the sequence from the first to the latter. Distinction of some distribution representations, such as, 'outside $\rightarrow$ inside' and 'inside $\rightarrow$ outside', will depend on inference rules between order, position and shape mentioned in the next section.

After developing such basic possibility distribution of order π(O) above, a complex distribution could be derived, for instance,

π(O) $^{(1)}$ *(top* $\rightarrow$ *bottom (left* $\rightarrow$ *right))* = {***top*** $\rightarrow$ ***bottom left*** $\rightarrow$ ***bottom right***}.

$Poss_\pi$ (O) for π(O) is defined by, for instance,

$Poss_\pi$ (*top* $\rightarrow$ *bottom*) = {***there are two rectangles***}.

Now, possibilistic inference rules $\Re^{(PO)}$ might be established for representing relations between the position and order of a radical. As examples, several rules are shown as follows.

$\Re^{(PO)}{}_{(1)}$: **IF** position **is** top **THEN** order **is** first,

$\Re^{(PO)}{}_{(2)}$: **IF** position **is** bottom **THEN** order **is** last.

4.3.3 Shape

The shape representation method centres on the shape domain of radicals, their possibility distributions and measure technique. Inference rules are established for the representation of radicals' relationships between their shape, position and order.

i) Possibility Distributions and Measures

The domain of radicals is defined as a rectangle in different sizes in terms of features of combined strokes. The shape domain of radicals is expressed by

S = {***rectangle***}.

Different combinations of basic strokes are assigned as the possibility distributions on the domain S, where the validity of the combinations is checked. The models of combinations are classified as connection and disconnection. The possibility distributions are represented by

π(S) = {***combination of basic strokes, basic strokes***}.

In order to determine the shape of a radical, possibility measures are based on evaluation of a continuous line, direction of a line connecting with other lines, priority of such direction and disconnecting distance. For instance, one of the possibility measures $Poss_\pi$ (S) for π(S) is defined as follows:

Poss_π *(priority of up → down)* = {***up → down, up → down left, right → left, up → down right***}.

ii) Shape Vagueness and Possibilistic Inference Rules

To produce a general concept of forming a radical, the shape vagueness of radicals is investigated for expressing the relation of combining two strokes. The relations can be classified as angle, location, continuous, distance and discontinuous.

The angle relation indicates a contour expression of two connected strokes. For example, it is defined as a contour if two connected strokes form an angle. Figure 4.12 shows three different types of angles from two connected strokes.

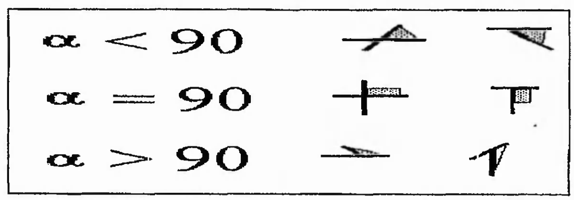


Figure 4.12 Angle of strokes connected

The location relation stands for the intersection point of two connected strokes. Figure 4.13 gives several examples to show the location relation.

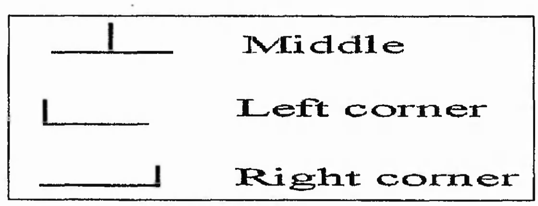


Figure 4.13 Location of strokes connected

The continuous relation expresses the possibility of a contour as part of a radical. A continuous contour is defined if a contour is formed with an angle.

The distance relation is to measure a scope of two disconnected strokes.

The discontinuous relation implies the possibility of a contour that may be broken down into two radicals. The

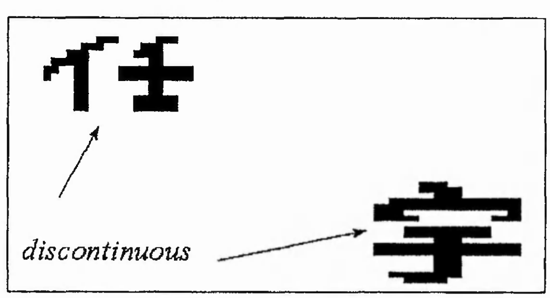


Figure 4.14 Discontinuous contour

distance of two disconnected strokes decides a discontinuous contour. Figure 4.14 gives two examples for showing the discontinuous contour.

Possibilistic inference rules are established by representations of relations between shape vagueness denoted by $\Re^{(S)}$; between shape and position by $\Re^{(SP)}$; and between shape, position and order by $\Re^{(SPO)}$. For example, the inference rules shown below are defined to divide a character into two parts: c1 and c2 from the inside to outside.

$\Re^{(S)}{}_{(1)}$: **IF** contour of c1 **is** square **AND** c2 **is** continuous contour of c1
AND angle of c1 connecting with c2 **is** 90
AND location of c2 **is** on the top middle of c1
THEN shape **is** combination of c1 and c2 (c1+c2).

$\Re^{(SP)}{}_{(2)}$: **IF** shape **is** c1+c2 **THEN** c1 position **is** outside.

$\Re^{(SP)}{}_{(3)}$: **IF** shape **is** c1+c2 **THEN** c2 position **is** inside.

$\Re^{(SPO)}{}_{(4)}$: **IF** shape **is** c1+c2 **AND** position **is** outside **THEN** order **is** last.

$\Re^{(SPO)}{}_{(5)}$: **IF** shape **is** c1+c2 **AND** position **is** inside **THEN** order **is** first.

4.4 Normalisation

Normalisation of an image or a picture is to make an adapted version of the original. This includes noise removal from an area, median filtering, thinning, position normalisation [Kov92] and shape recognition. Problems in position normalisation and shape recognition include: (a) that the position or shape information to be learned and recognised is difficult quantified; (b) that it is unaware of which shapes are important to visual tasks such as classification and discrimination in recognition; and (c) the lack of an appropriate language for position and shape discrimination and analysis [Bal91].

The Normalisation subsystem is dealing with variations of the position, size and shape of radicals extracted from a character using the Neocognitron neural network technique.

4.4.1 Variable Radicals

The variability of radicals can be investigated in three aspects: position variance, size variation, and shape transformation.

Position variance of a radical means that it could be located at different positions in a character when it is extracted. Figure 4.15 shows several examples.

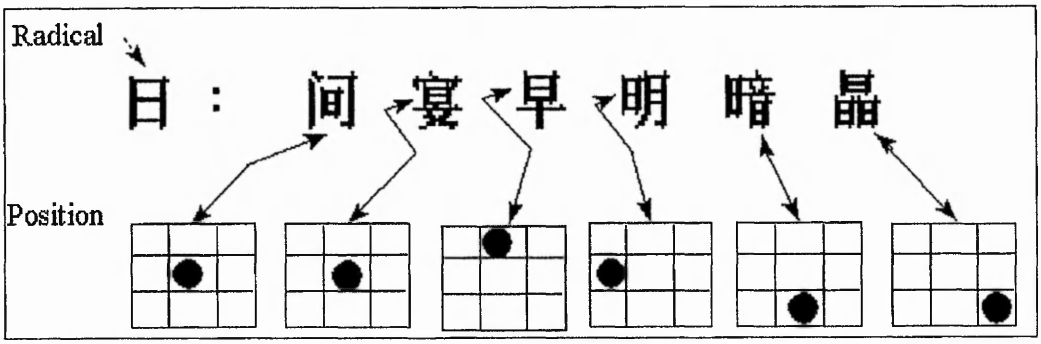


Figure 4.15 Position variance of a radical

Size variation of a radical is caused by the fact (a) that a character containing radicals is set in different sizes as input, and (b) that keeping a balance of writing a character in a square or a rectangular shape changes the sizes of radicals in it. In the first case, the same radical with different sizes is shown in Figure 4.16 (a) where a character in different sizes is input. In the second case, a radical is in a normal size if it is acting as a character. However, the radical as a component of a character has reduced its size as shown in Figure 4.16 (b).

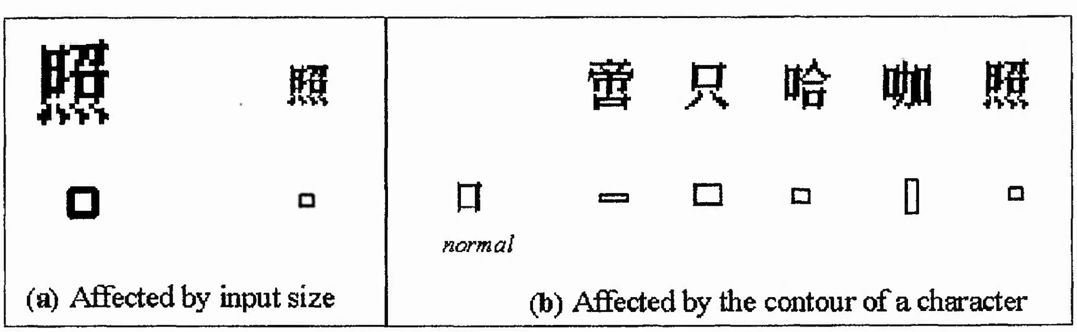


Figure 4.16 Size variation

In general, **shape transformation** of a radical implies its contour having been changed. The transformation of a radical can be partial or complete as illustrated in Figure 4.17.

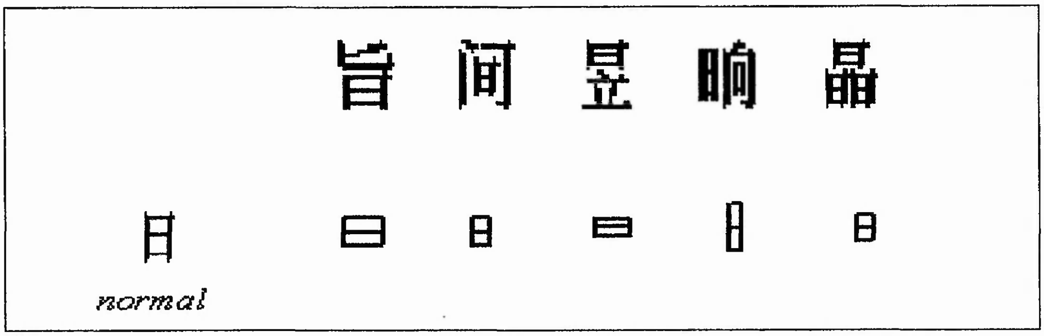


Figure 4.17 Shape transformation

Sometimes, transformation of radicals' shape leads to **ambiguity** of its shape discrimination. For instance, a radical in Figure 4.18 (a) can be seen as shape transformation of one in Figure 4.18 (b) so that a confusion in distinguishing them might occur.

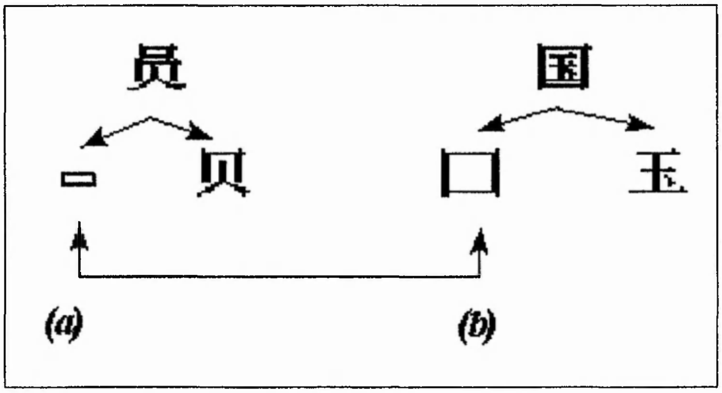


Figure 4.18 Ambiguity of shape discrimination

4.4.2 Normalisation Subsystem with Neocognitron Structure

Applying algorithms of the Neocognitron neural network, the subsystem has been designed as a network that includes learning and training phases. In the learning phase, formal patterns with different sizes and positions have to be learnt. In the training phases, the Neocognitron network can be trained in either supervised or unsupervised.

The unsupervised network behaves less successful but is more biologically plausible. In the supervised case, the network is trained from the first hidden layer upwards. Each S-plane in the hidden layers has to be identified by the teacher with a visual feature it must try to learn. When the training is applied to radicals, several different training patterns have to be used in different hidden layers, as shown in Figure 4.19.

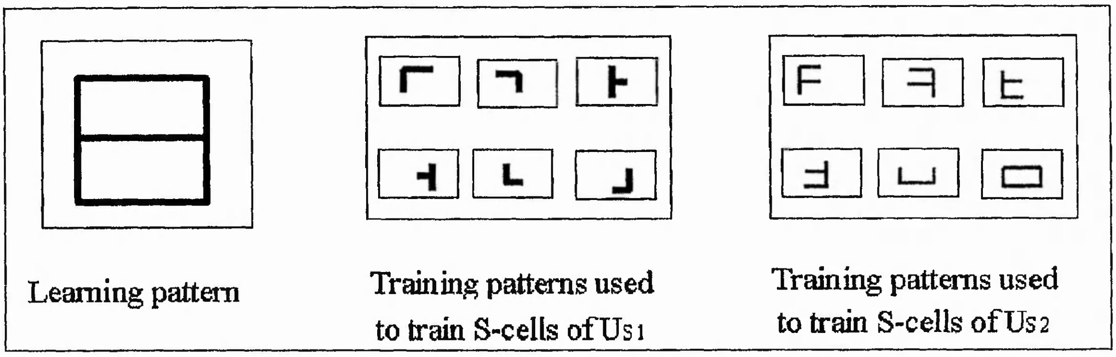


Figure 4.19 Some training patterns in the hidden layers

The basic principles of dealing with varying patterns in the network are: (a) tolerance to small position errors; (b) local feature extraction for gradually integrating into more global features; and (c) all features respond to one at the highest stage by each C-cell of the recognition layer.

4.5 Recognition

Basically, the Recognition subsystem is a four-layer neural network, composed of multi sub-nets and based on architecture of associative memory function [Ald95]. Sub-nets are developed to reduce intra connectivity of the network and to deal with radicals in a category. The associative memory function offers an important advantage of recalling a stored pattern from its partial or noisy input [Fre92].

4.5.1 Discussion of the *Cang-Jie* Method

In the *Cang-Jie* method, radicals and token-radicals are organised into 4 groups with 24 categories and 2 special categories. Although some radicals and token-radicals in characters are quite difficult to match into a category even by experts, these groups and categories have special advantages for illustrating topological structure and traditional culture customs. Assigning 26 English letters A to Z to these categories allows the translation of a radical from its pictorial to a letter format, i.e. from a 2-D structure to a single letter [Ren94].

4.5.2 Development of Classification of Radicals

Since radicals have been determined as major objects for recognition, the policy of classifying radicals has to be considered carefully. Three basic principles of determining categories are developed: (a) a member in a category should have the physical properties of the category and major features of the group to which the category belongs; (b) each member in a category may be a radical or a token-radical or some combinations of basic strokes; and (c) combinations are allowed between a token-radical and basic strokes to form a new integrated radical. The policy has some benefits in transforming knowledge of a radical identified abstractly by human analysis into its shape recognised by computer.

4.5.3 Architecture of the Associative Memory Neural Network

The associative memory neural network in the subsystem consists of four layers: input, hidden-1, hidden-2 and output. The hidden-1 layer consists of multi sub-nets where each sub-net deals with radicals in a category. The number of neurones in each sub-net is decided by the learning patterns in the category. The connectivity from the input to the hidden-1 layer is static. Neurones in the hidden-2 layer are created by the results gained from the hidden-1 layer. The connectivity between the two hidden layers is dynamic. The design of the hidden-2 layer with a dynamic structure is used to further enhance convergence on global minimum of the associative algorithms. Figure 4.20 shows the architecture of the network.

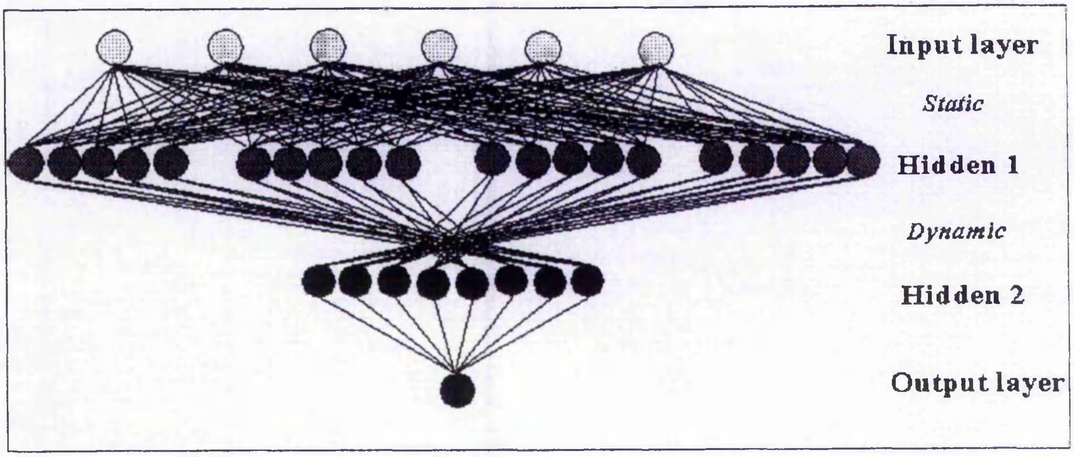


Figure 4.20 Architecture of the associative memory neural network in the Recognition subsystem

In the learning phase, radicals are classified into categories. Each is represented by a sub-network that is used for reducing the connectivity of the whole network and for using shared weights. The 26 different sub-nets are composed of a whole neural network with the associative memory function. The major task in the learning phase is to learn formal radicals and to form inter-connectivity for training a pattern.

In the training phase, sub-nets in the hidden-1 layer are trained to converge to local minima. The hidden-2 layer is generalised by re-learning these patterns of local minima. Eventually, the global minimum will be converged to the output layer.

When the network is connected as a whole, its inter-connectivity is low but intra-connectivity is high [Ben93], while the inter-connectivity is connectivity of neurones between layers, and the intra-connectivity is ones at the same layer, shown in Figure 4.21. The intra-connectivity can be reduced while the architecture of sub-nets is being used.

Weights in the network can be shared for saving space and time, i.e. weights that are associated with different (e.g. translated) input features may be shared, and weights that are associated with different times may be shared. In the current network, weights shared in feature space are considered, but not in time.

4.5.4 Modification of the Network

The modification of the associative memory function in the network aims to enhance convergence to a global minimum [Ren95a]. This modification has been made in both of the learning and training phases. In the learning phase, the modification centres on changing reasonable parameters for Hebbian rules shown in equation (3.7) in Section 3.4.2 of Chapter 3, so that the convergence is ideally forced to search for all patterns. Two assumptions are made:

$$\theta_i = \varphi_j = 0 \quad or \quad \theta_i = \varphi_j = \frac{1}{2}\sum W_{ij} \qquad \ldots\ldots \qquad (4.1)$$

In the training phase, the modification is concentrated on how to enhance local minima to converge to a global one. The enhancement is dynamically formed in the hidden-2 layer, as shown in Figure 4.21.

The converged quality, indicated by the parameter 'reliability rate', is also taken into account for a global comparison of different results when referring to learning patterns. There are three possibilities of results: recognition, mis-recognition and failure. The reliability rate produced depends on the matching quality and training quantity of a pattern [Ren97a].

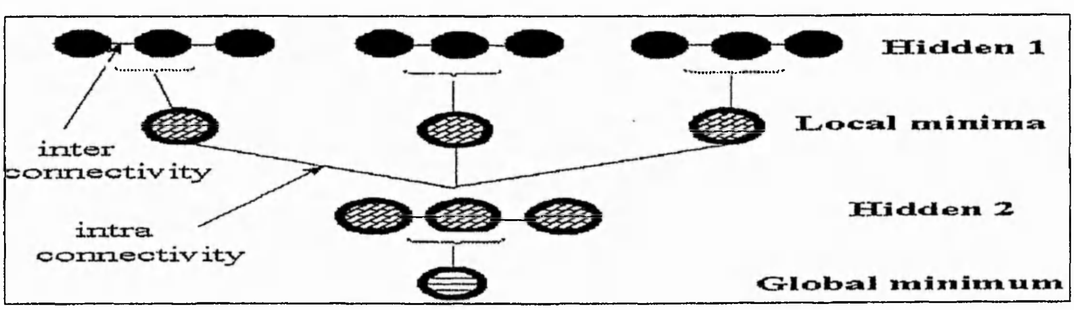


Figure 4.21 Function of the hidden-2 layer

4.6 Restructuring

The restructuring of a character is able to reorganise radicals from their letters into a chain code for verification at the character level as illustrated in Figure 4.22. A subsystem called Restructuring is developed for carrying out the process using the chain code method. The architecture of the subsystem is shown in Figure 4.23.

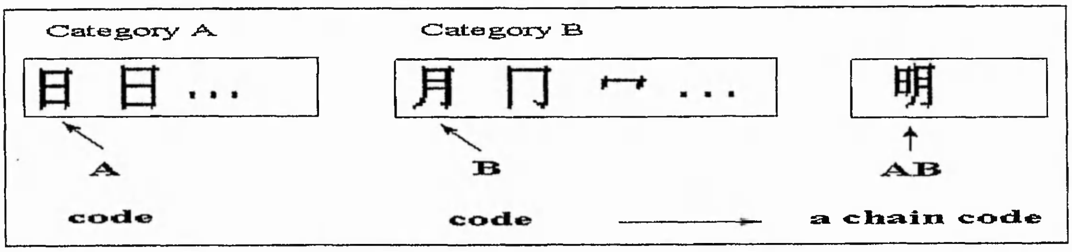


Figure 4.22 Code data

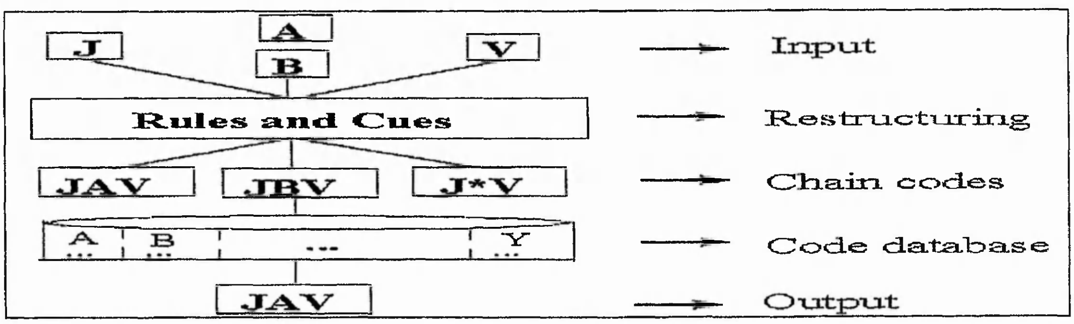


Figure 4.23 Architecture of the Restructuring subsystem

4.6.1 Restructuring Cues

The chain code method uses restructuring cues to form a chain that contains several codes and to verify it using a code database. Restructuring cues are some specified rules for successful formation of a chain code showed in Figure 4.24. Some of these rules are listed as follows.

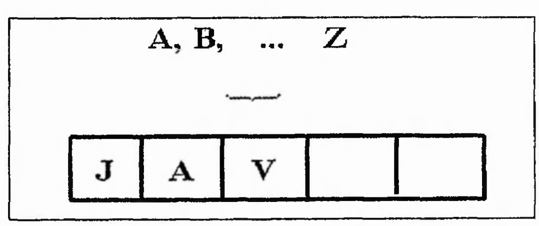


Figure 4.24 Format of a chain code

i) Filling Order

The filling order is to guarantee the accuracy of a chain code. It relies on extracting rules while a radical is extracted from a character. The priority of the filling order is defined in Figure 4.25 (a). One of its applications is shown in Figure 4.25 (b).

Priority	Order
1	*outside* → inside
2	*inside* → outside
3	*top* → middle → bottom
4	*top* → *bottom*
5	*left* → middle → right
6	*left* → *right*

(a) Priority

outside →

inside top → inside bottom →

inside bottom left → inside bottom right

(b) Application of the priority

Figure 4.25 Filling order

ii) Reliability

The reliability indicates the quality of a chain code and it has three possibilities: recognition, mis-recognition and failure for a code. A code or a chain code that has been recognised is independent of adjacent codes and can be matched to one in a code database. It is possible for mis-recognised codes to be matched if codes can be found in the database. Recognition of a failed code will depend on successful matching of its neighbours in a chain code. In this case, different approximate estimates have to be accepted as results for further exploration.

iii) Importance of the First Code

The first code in a chain code is very important in encoding because it stands for the category and affects the filling order of a chain code. Because the order of a chain code in the code database follows the one of its first code, the first code with a good reliability will greatly reduce the possibility of multi chain codes corresponding to one character. In addition, the uncertainty of the first code will bring undefined priority into the filling order.

iv) Omitted Codes

At least one redundant code has to be omitted when the total number of codes in a chain code is more than 5. In other words, the structure of the 5 codes or less has ensured the identity of a chain code. The priority of omitted codes is defined as using a discontinuous selection in the last part of a chain code, as shown in Figure 4.26. The discontinuous selection is considered easier to be interpolated if the complete structure of a chain code is required.

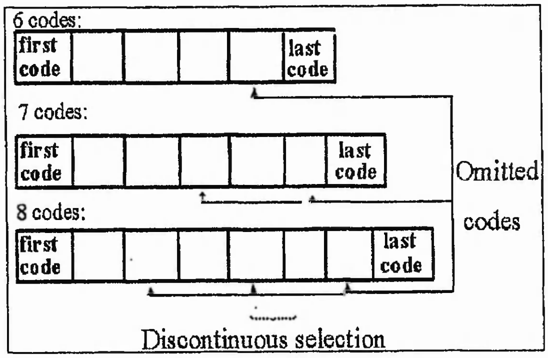


Figure 4.26 Selection of omitted codes

v) Necessity of the Last Code

The last code sometimes plays a necessary role while a chain code is formed by complex rules. In this case, the last code is a referring point to any omitting code; i.e. an omitted code will be located at one before the last code.

vi) Possible Code

A possible code is regarded as the replacement of a code with a low reliability by guess. In general, a better result can be obtained if a possible code is located in the middle of a chain code. The quality of the overall chain code will be affected when a replacement is put in its first code.

vii) Impossible Position of a Code

An impossible position of a code means that the code is put at an incorrect position so that the order of a chain code is contrary to the structure of Chinese characters. Figure 4.27 gives an example of an impossible position to show that the code O is only allowed in the left rather than in the right of a character. In this case, the code has to be regarded as failure.

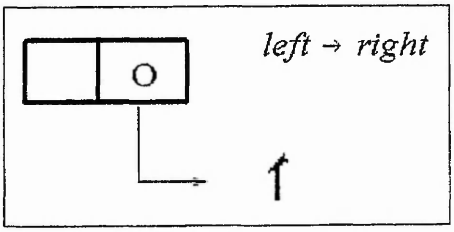


Figure 4.27 Impossible position

viii) Quality of a Chain Code

The quality of a chain code is a combination of three factors: high reliability, result of matching it to the code database, and success of filling order. Figure 4.28 lists relations of the quality and the three factors.

Quality	Code reliability	Matching to database	Filling order
* * *			
order	0 recognition	0 match	0 success
matching	1 mis-recognition	1 possible	1 failure
reliability	2 failure	2 failure	

Figure 4.28 The quality of a chain code

4.6.2 Code Database

A code database is used for verifying a chain code, and is referring to an external database in support of displaying Chinese characters in a pictorial format. The advantages of using the code database are: (a) a guarantee of character recognition in CCRS; (b) a simple process of verifying 1-D chain codes instead of 2-D characters; (c) easy checking which code in a chain code has not been recognised properly; (d) a simple method of connecting it to an external database of Chinese characters that occupies a large space but displays a character intuitively.

The structure of the code database is designed in two levels: control and entity. The control level deals with categories that follow the order of the first code of chain codes. The entity level includes the context of chain codes and reference address to an

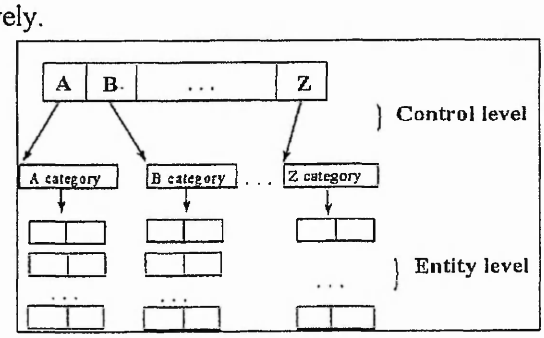


Figure 4.29 Structure of the code database

external database. Each chain code is composed of one to five English letters from A to Z and arranged in the alphabetical order. Figure 4.29 shows the structure of the code database.

An interface program is allowed to access the code database. The program is able to search possible solutions as results from the database when the quality of a code is low. Figure 4.30 shows an example of the possible solutions.

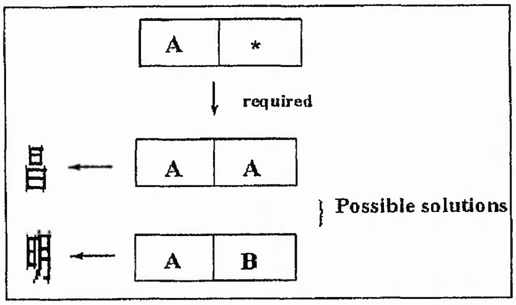


Figure 4.30 An example of possible solutions

4.7 Postprocessing

Postprocessing subsystem deals with the assessment of errors in chain codes using the analysis of error probability method. The architecture of the subsystem is shown in Figure 4.31. The assessment aims to (a) correct errors that happened in recognising characters so that an optimum result can be obtained; and (b) reject an incorrect character when its failure in recognition has been confirmed.

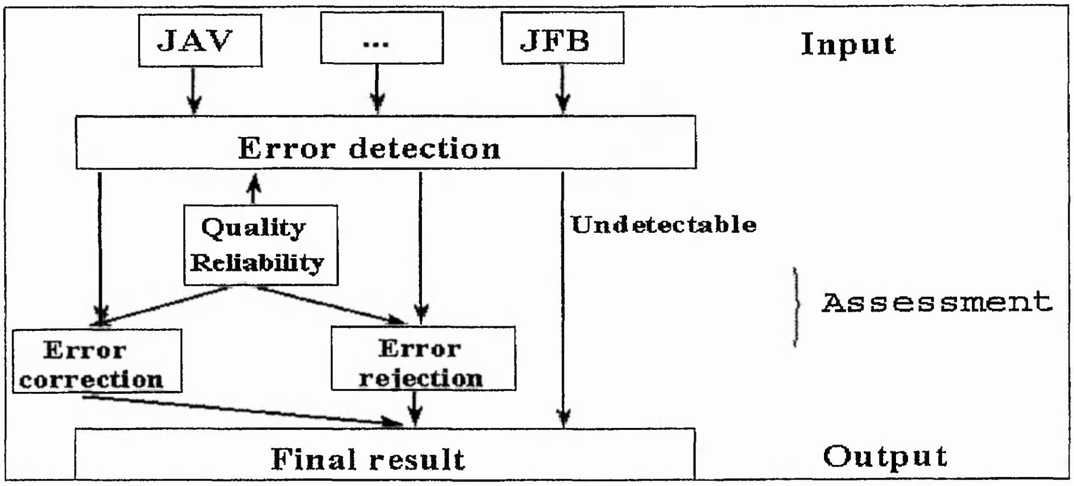


Figure 4.31 Architecture of Postprocessing subsystem

4.7.1 Assessment Scheme

The assessment scheme that applies the analysis method relies on two important factors: quality rate from Restructuring, and reliability rate from Recognition, to detect errors for correction and rejection [Ald95].

The quality rate is a series of figures that indicate measurement of a chain code on its code reliability, matching it to a database and filling order on recognition. The reliability rate gives more details of convergent iterations and training times of every code in a chain code. Comparatively, the quality rate is an integrating measurement on a chain code. The reliability rate is more concentrated on statistics of each code in a chain code.

The scheme includes three parts: error detection, error correction and rejection. The error detection gives a measurable criterion for checking errors. Error correction and rejection are assessment actions.

4.7.2 Error Detection

Error detection is able to discover an error or an ambiguous case in a chain code. Four measurable criteria are used: undetected, unaffected, affected, and error. In the undetected case, a chain code is thought of as a good one, and no action is taken. In the unaffected case, an error in a chain code is discovered but it may not affect the code too much. Consequently, the assessment action can be ignored as in an undetected case. A chain code in the affected case needs to be examined by an action. In the error case, an error detected affects the property and quality of the code. Assessment action has to be taken. The measurable criteria for error detection with related information are listed in Table 4.1.

Error detection	Quality	Reliability	Action
Undetected	Good	Good	None
Unaffected	Fine	Good/Fine	None
Affected	Uncertain	Uncertain	Correction / Rejection
Error	Uncertain	Uncertain	Correction / Rejection

Table 4.1 Measurable criteria for error detection

4.7.3 Error Correction

Error correction is an assessment action of correcting an error when it has affected the property or quality of a chain code. However, the quality and reliability rates of a chain code provide some cues for correcting an error unless it belongs to a case of rejection. In general, the reliability rate is referred to a further comparison of two codes for a better solution. The quality rate is for determining the correction. Some general cues are listed as follows.

i) Good Quality Replaces a Worse One

A chain code on a good quality must replace a worse one when a choice between them is demanded. The standard of good quality is measured with the standard of recognised, complete matching to one in a database, and success of filling order.

ii) Priority of the Quality Rate

When a correction is required, the priority of elements in the quality rate is defined as code reliability firstly, then matching, and finally filling order. A code with a good reliability could be firstly used to replace one with the worse reliability, but in reverse, a code with the worse reliability has to be used last after consideration of its matching ability and filling order.

iii) Integrating vs. Partial Consideration

The integrating and partial considerations of the quality rate take place between a whole chain code and its internal codes. Basic rules define that a chain code with half codes on good quality can replace one with all codes on the bad quality. This means that partial recognition of a character gives a high probability to identify a whole character via matching it to a database.

iv) Priority of Reliability

The highest priority in the reliability rate is given to a code with faster iteration, fewer training times, and a good reliability rate. Meanwhile, the faster iteration has a higher priority than the training times because fast convergence means that the neural network has matched up some major features of a pattern.

v) Unsuccessful Correction

Unsuccessful correction of a chain code implies that assistance of human expert knowledge will be needed.

4.7.4 Rejection

Rejection is an assessment action of dismissing a chain code in terms of its errors. Several cues are considered as follows for this examination.

i) Quality Rate on Failure of Filling Order and Matching

A chain code fails on its filling order and matching means that it cannot be corrected any more.

ii) Quality Rate with Failure of Reliability

If a chain code fails on its reliability, it will be an unsuccessful recognition. In this case, there is no chance to save the code. Therefore, a rejection has to be taken.

iii) Quality Rate with Failure of Reliability and Matching

It is regarded as hopeless if a chain code fails on its quality, reliability, and matching. The chain must be rejected.

Chapter 5

Results and Discussions

5.1 Introduction
5.2 Performance of CCRS
5.3 Extraction of Radicals
5.4 Radical Normalisation
5.5 Classification and Recognition of Radicals
5.6 Restructuring of Chain Codes
5.7 Error Assessment

5.1 Introduction

In this chapter, a variety of experimental outcomes derived from the CCRS are illustrated and discussed. The experiments are focused on the application and evaluation of the three-layer hierarchy method and different functions of the CCRS. Discussions are on the basis of detecting problems or difficulties existed in the current system, evaluation of new features, and assessment of the system performance for future development.

The performance of the CCRS to be presented in section 5.2 has the emphasis on the overall structure, evaluation of time utilisation, and bottlenecks in the CCRS. Character deconstruction achieved by the Preprocessing subsystem in 5.3 is concentrated on deconstructing a character into different radicals in different shapes, positions and orders. Radical normalisation in 5.4 gives results of dealing with radicals in variable position and shape transformation carried out by the Normalisation subsystem. Classification and recognition of radicals performed by the Recognition subsystem in 5.5 are focused on assessment of architecture and effectiveness of neural networks. Restructuring of chain codes processed by the Restructuring subsystem in 5.6 is centred on analysis of chain codes and their formation in alternative cases. Error examination derived by the Postprocessing subsystem in 5.7 has a focus on correction and rejection of errors discovered from a chain code. Further details of experiments and outcomes from the CCRS have been listed in Appendix C: Experimental data and outcomes.

5.2 Performance of CCRS

The performance of CCRS is laid out on windows with the pop-down menus. Every function of the system is given as an option in a certain menu. The entire performance is supported in an option-choice manner. More details about the structure of the menus are given in Appendix B: Developed Software.

5.2.1 Structure of the Main System and Subsystems

The structure of the CCRS is divided into the main system and subsystems. The main system, as illustrated in Figure 5.1, controls the performance tasks of the subsystems, translations of data, statistics of time resources utilisation, and management of the system.

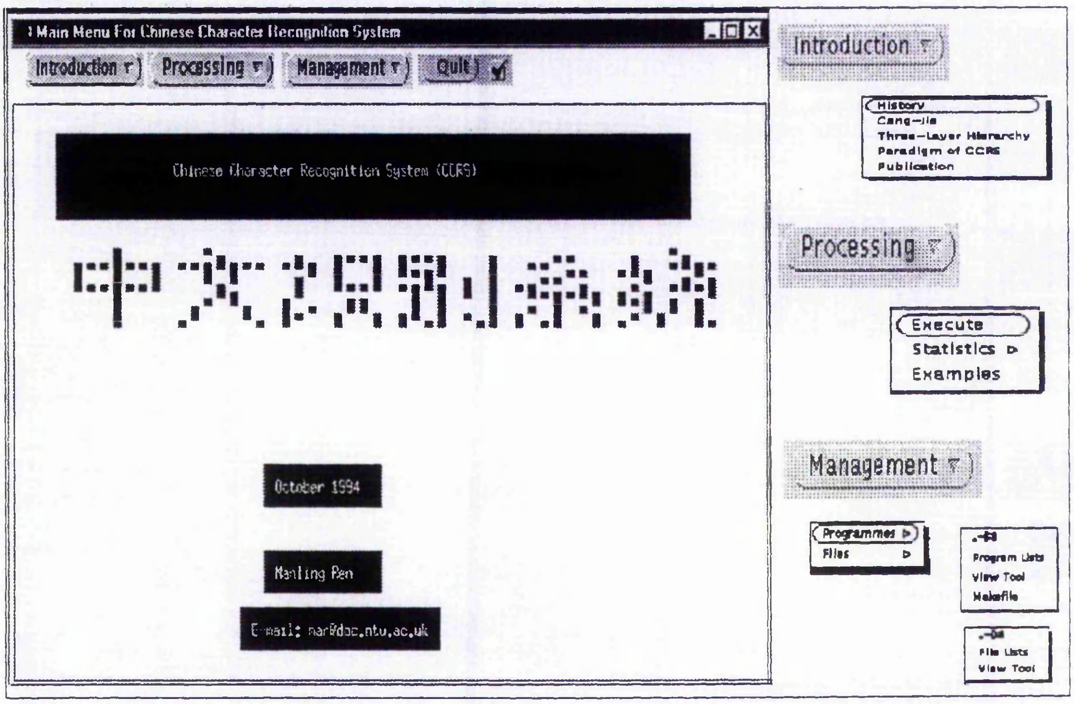


Figure 5.1 The performance structure of the main system

The performance structure of subsystems is based on the model of option-choice and program-opacity. The model is allowed to select an option from a menu, run a program of the subsystem, observe inputs and outcomes afterwards. The structure is displayed in Figure 5.2.

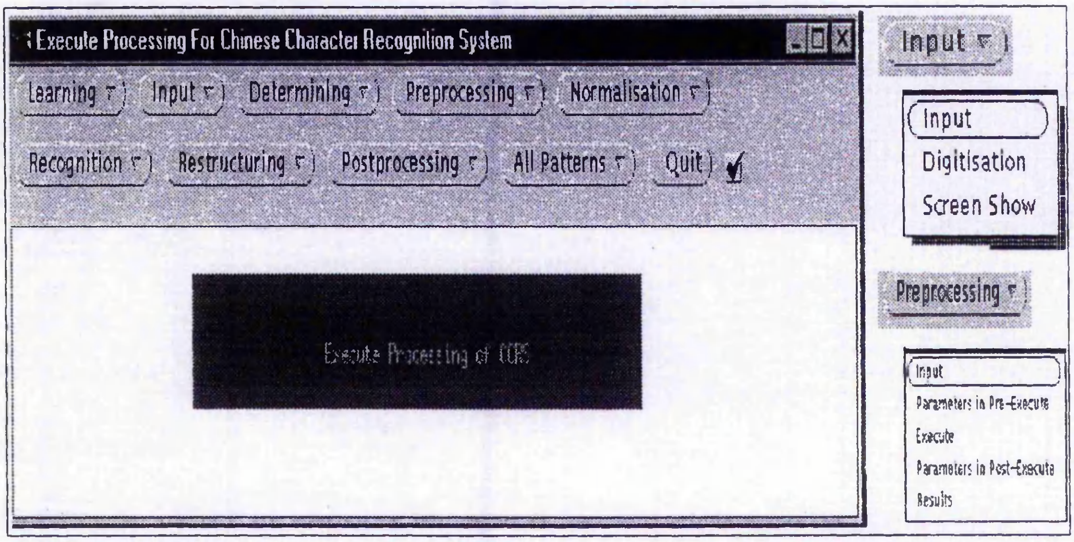


Figure 5.2 The performance structure of the subsystems

5.2.2 Performance Data

The data used in the performance of the CCRS has four models: bitmap as shown in Figure 5.3 (a), digitised as shown in Figure 5.3 (b), chain code shown in Figure 5.3 (c), and display shown in Figure 5.3 (d). Data in the bitmap model is used as input characters, and in the display model is for demonstrating a character on screen. Both of bitmap and display models are used as options in the main systems. Data in digitised and chain code models is generated by executing programs of the subsystems.

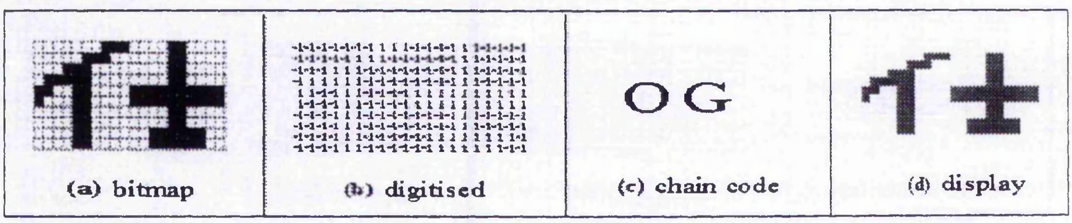
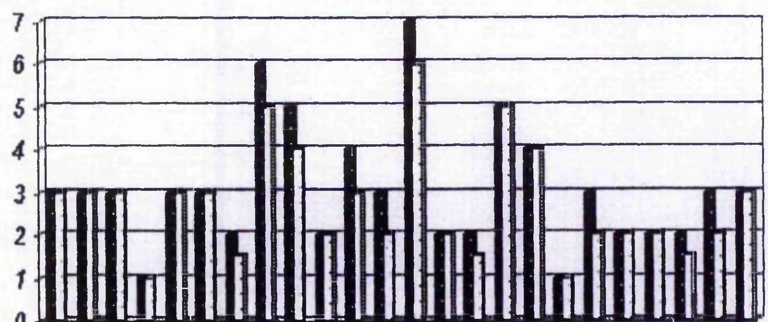


Figure 5.3 Performance data

5.2.3 Time Utilisation

The performance time used by the CCRS includes monitoring the time of learning patterns and executing five subsystems, i.e. the process time of recognising a character. Both of these times can be displayed by running the monitoring program of the CCRS. An example of monitoring the time for learning formal radicals is shown in the following three figures, where the entire time is in Figure 5.4 (a), the user time only in Figure 5.4 (b), and the system time in Figure 5.4 (c). In these figures, the x-axis (horizontal) stands for categories of radicals; the y-axis (vertical) is for the number of radicals in a category and the time of learning these radicals is in seconds.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	Y
■ *Radicals*	3	3	3	1	3	3	2	6	5	2	4	3	7	2	2	5	4	1	3	2	2	2	3	3
□ *Time in seconds*	3	3	3	1	3	3	2	5	4	2	3	2	6	2	2	5	4	1	2	2	2	2	2	3

Figure 5.4 (a) The monitoring time for learning formal radicals

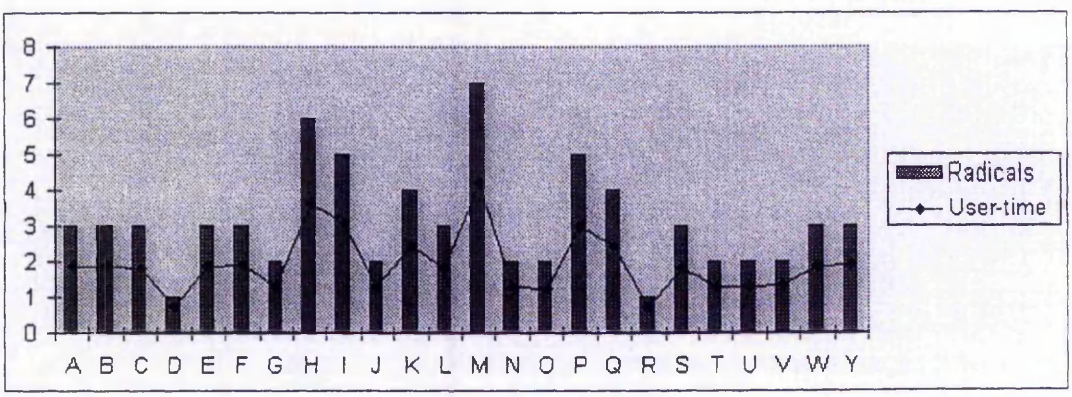


Figure 5.4 (b) The user time

Analysing the system time spent on learning radicals in different categories shown in Figure 5.4 (c), a flat line very near to zero can be observed. This explains that the I/O process doesn't take much time on the learning process of the CCRS.

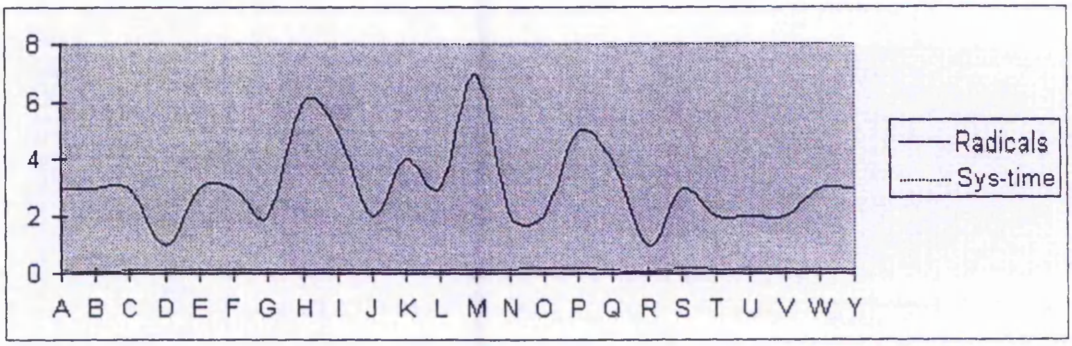


Figure 5.4 (c) The system time

Figures 5.5 (a) and (b) show the entire execution time of the CCRS in recognising a character. In these pictures, the colours are used to symbolise the time spent on the five subsystems and waiting time. In Figure 5.5 (a), the x-axis (horizontal) stands for the number of characters to be recognised; and the y-axis (vertical) is the process time in seconds. In Figure 5.5 (b), the numbers with solid and dash lines indicate time utilisation on recognising a character and on the I/O, respectively.

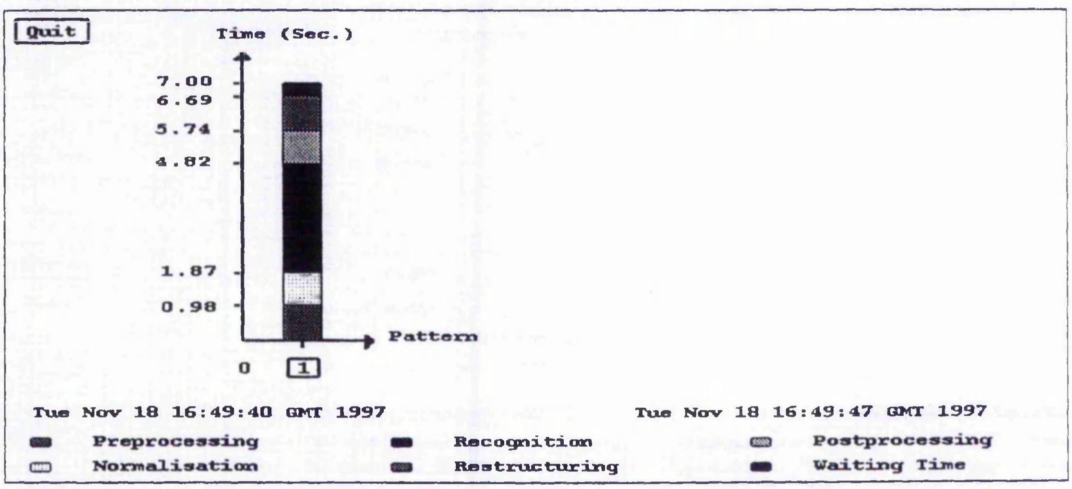


Figure 5.5 (a) An example of the time statistics

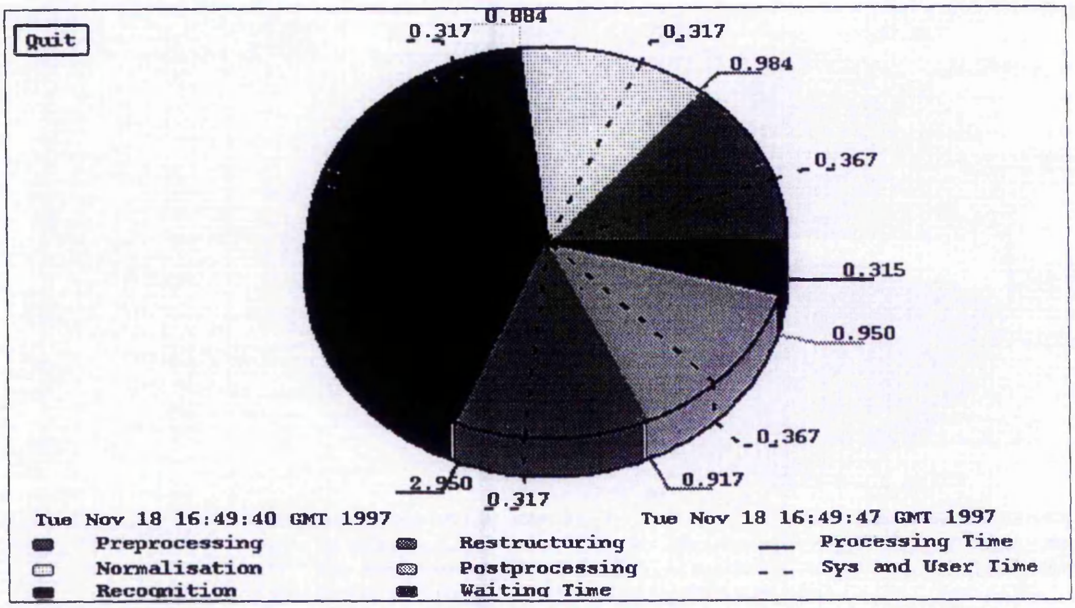


Figure 5.5 (b) Details of the executing time

5.2.4 Discussion on Performance Time

Although the performance time of the CCRS has not become a bottleneck problem at the current stage, a suggestion can be proposed for future development. Altering the process model from serial (i.e. to process the sub-nets one by one) to parallel (multi-processors to deal with the sub-nets in parallel) in the hidden-1 layer of the network in the Recognition subsystem can reduce its processing time.

5.3 Extraction of Radicals

In order to test the fuzzy possibilistic inference rules run by the Preprocessing subsystem for extracting radicals from a character effectively, some experiments have been conducted. In the extraction processing, the experimentation focused on different structures of radicals in a character and the extraction technique.

50 standard writing characters with different structures, which can be extended to more characters with the same structure, were used to examine the extraction rules, shown in Figure 5.6. These characters were chosen with representation of radicals in different

positions, shapes and orders. The correct radicals have been extracted from 48 out of 50 test characters, i.e. a 96% success rate. In the implementation of the preprocessing, a dynamic scheme was employed for updating possibilistic inference rules while they were being evolved around a sample set of special cases.

Figure 5.6 Test characters

5.3.1 Different Positions

The implementation of extracting radicals that are located at different places in characters has examined the capability of rules to cope with various structures of characters.

5.3.2 Different Shapes

Examining different shapes of radicals centred on dealing with basic shapes and complex shapes. The basic shapes include the shapes of basic strokes, and complex shapes are combination of basic strokes, for instance, shapes of a rectangle, square, cross, y or u shape, which can be referred to the definitions in Section 3.2.2. Figure 5.7 shows some results of the implementation.

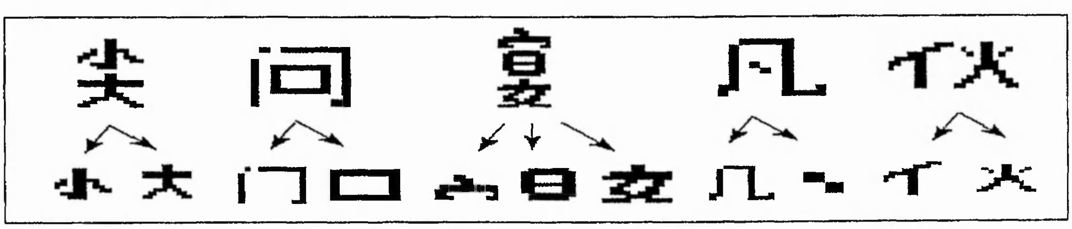


Figure 5.7 Different shapes of radicals

5.3.3 Different Orders

The extraction order is able to decide which radical has to be extracted first in the case of a character being composed of more than one radical. Examples shown in Figure 5.8 are some results of testing order rules.

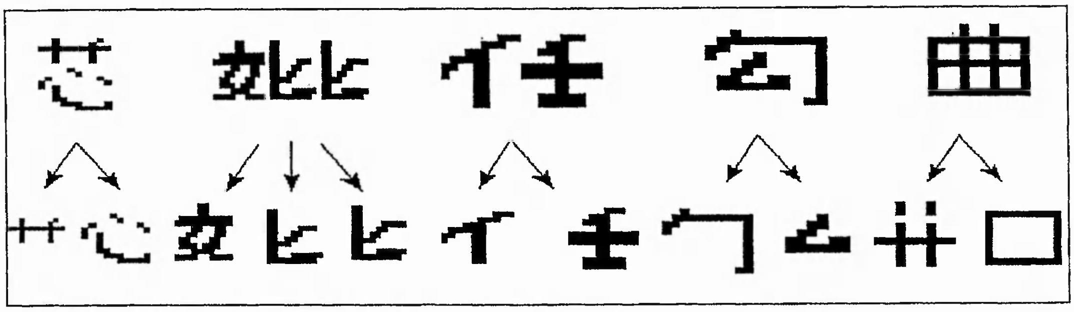


Figure 5.8 Results of different orders

5.3.4 Expansibility of Rules

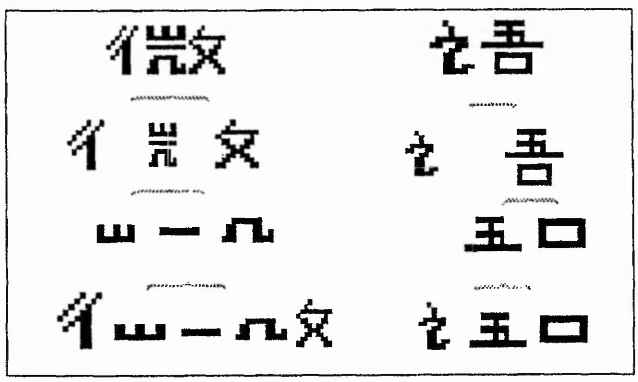


Figure 5.9 Examples for expansibility of rules

In addition to dividing a character as a picture into several radicals, i.e. sub-pictures, some sub-pictures might be further segmented into smaller sub-pictures, i.e. sub-sub-pictures, where the sub-pictures are formed from two or more radicals. The segmentation is termed expansibility of possibilistic rules, described as the complex distribution in Section 4.3.2. Examples in Figure 5.9 shows the implementation results of the expansibility of rules.

5.3.5 Special Cases of Extraction

In analysing the above results, two types of incorrect results that appeared in the extraction process need to be carefully investigated. Incorrect results can arise in two special cases where (a) an individual radical has a discontinuous shape, and (b) the shape of two radicals connected together is without a discontinuous part, shown in Figure 5.10, for example.

For the first case, a complete radical is divided into two parts by the current inference rules. There are two methods to deal with this problem according to human analysis. One method is to have special rules, probably against existing rules, to deal with these special radicals. The other is to attach new rules that could examine a rectangular area occupied by a radical. If the area is small enough only for a stroke rather than a radical, the extraction in this case will be invalid or it will be treated as a single radical without extraction. However, it needs a statistic value to decide a minimum area for tolerating a radical because some of the strokes can be treated as radicals as shown in Figure 5.11 (b), but some cannot be as shown in Figure 5.11 (a).

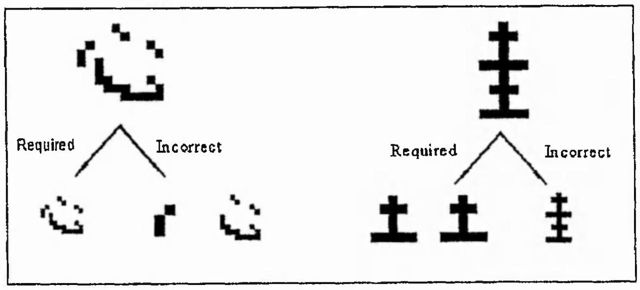


Figure 5.10 Two special cases

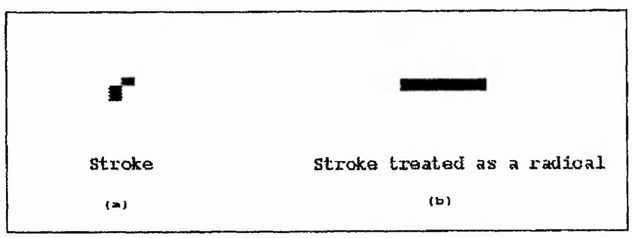


Figure 5.11 A case of strokes treated as a radical

In the second case, two radicals are connected, or overlaid together, and form a continuous shape. This is very difficult to deal with by only applying inference rules. Other methods should be further investigated for exploring these radicals. Currently, such radicals are treated as difficult ones.

5.4 Radical Normalisation

Normalisation deals with radicals to cope with position invariance and shape transformation. Experiments were carried out with the results from the Preprocessing stage. By using the Neocognitron neural network, the experimentation was focused on position adjustment and selection of different shapes.

5.4.1 Position Adjustment

120 radicals with different positions in characters were used to examine the function of position adjustment in the Normalisation process. All radicals have been adjusted to a formal position. Examples in Figure 5.12 show some outcomes of the process.

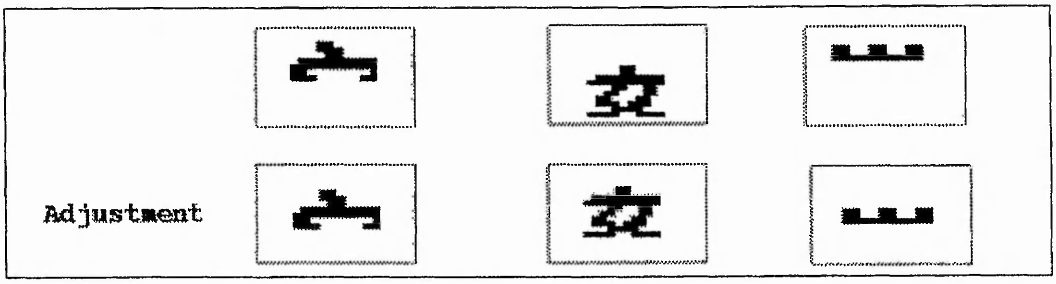


Figure 5.12 Position adjustment

5.4.2 Shape Selection

Examining shape transformation centred on the selection of tried and tested shapes as training patterns. 12 different shapes in 2 groups were trained for a learning pattern. 4 test patterns in different shapes were used to examine the network. 3 out of 4 test patterns have been normalised into a formal one. These test data are shown in Figure 5.13.

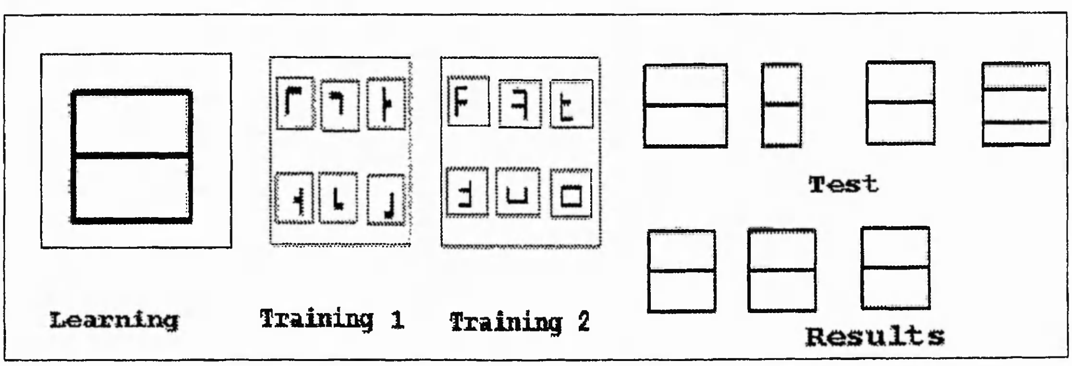


Figure 5.13 Shape selection and normalisation

5.4.3 Discussion on Fixed Patterns

Although the network possesses the strength of dealing with the shape transformation of a pattern, every training pattern was only used as a fixed pattern. For instance, patterns in Training 1 and 2 in Figure 5.13 can be only reserved for testing a pattern that has the same structure of its learning pattern. Otherwise, training patterns have to be re-selected. In this case, there will be a number of different training patterns being put in the network when a new learning pattern is added. This limitation will certainly increase the cost and redundancy of the network and will be against the characteristics of the topological structure in Chinese characters.

5.5 Classification and Recognition of Radicals

The experiments for classification and recognition of radicals followed on from results of the Normalisation subsystem. In the process of using the associative memory neural network with sub-net structure, the experimentation was focused on different classification, recognition and modification of the network.

The implementation of recognising radicals included two phases: learning and training. The learning phase was concentrated on the effect of different weights and error tolerance of the sub-nets. The training phase examined optimal architectures and convergence of the network.

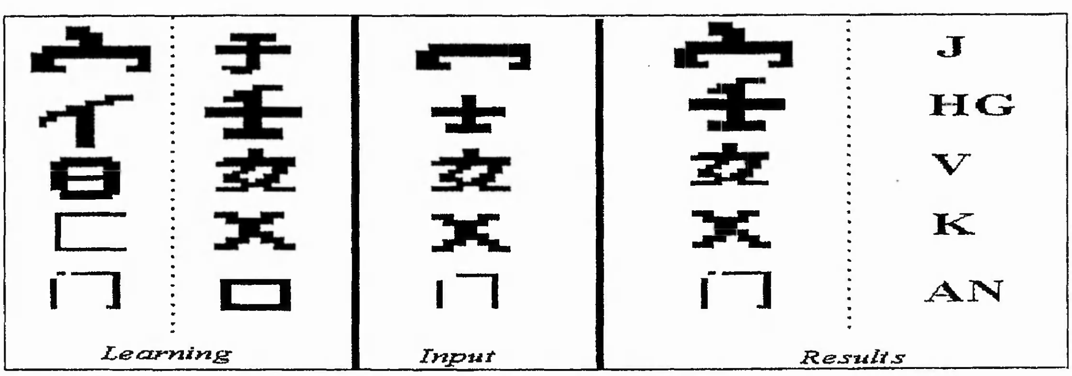


Figure 5.14 Recognition of different radicals

120 radicals covering 24 categories have been used to examine functions of classification, recognition and translation of the Recognition subsystem. Some of these are shown in Figure 5.14. Some ambiguous cases will be discussed in Section 5.5.5.

5.5.1 Classification of Radicals

According to standards of classification, test radicals were divided into the categories shown in Figure 5.15.

Group: Philosophy

Category	Name	Standard Radical	Radicals/ Token-radicals
A	Sun	日	门(N) 日
B	Moon	月	冂 冖
C	Metal	金	儿 钅
D	Wood	木	
E	Water	水	氵 又
F	Fire	火	小 ⺍
G	Soil	土	士

(a)

Group: Stroke Combination

Category	Name	Standard Radical	Radicals/ Token-radicals
H	Left-diagonal	竹	几(H) 王(R) 彳(H) 月(A) 壬(H)
I	Dot	戈	丶 厶 之(I) 扌(L)
J	Cross	十	宀
K	X connection	大	乂 力(S) 攵
L	Vertical	中	由(W) 卩
M	Horizontal	一	五(R) 石(I) 于(O) 歹(H) 工 乃(B)
N	Hook/Turning	弓	刂

(b)

Group: Physical Symbol

Category	Name	Standard Radical	Radicals/ Token-radicals
O	Person	人	亻
P	Heart	心	勹 七 毛 匕
Q	Hand	手	扌 龵 手(J)
R	Mouth	口	
X	Difficult		
Z	New		

(c)

Group: Shape Similarity

Category	Name	Standard Radical	Radicals/ Token-radicals
S	Flanking open	尸	丁 匚
T	Abreast balance	廿	艹
U	U shape	山	凵
V	Twisting shape	女	幺(F)
W	Square	田	囗 四(L)
Y	Y shape	卜	文(K) 亠(R)

(d)

Figure 5.15 Classification of test radicals

Within these categories, a radical with a tag represents a combination of radicals or token-radicals that are independent in different categories in the *Cang-Jie* method. The tag is used for referring to a database of Chinese characters in the *Cang-Jie* method, instead of building up a new one.

5.5.2 Learning Phase

Radicals in each category were learnt by the learning phase of the network to form connectivity schemes among its sub-nets in hidden-1 layer. Some learning patterns are shown in Figure 5.16.

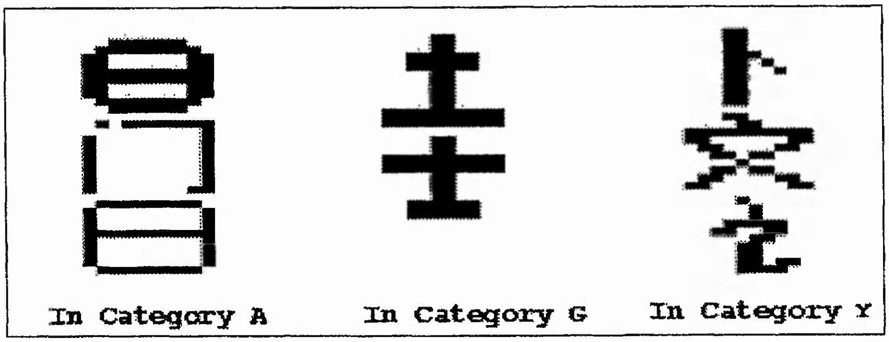


Figure 5.16 Some learning patterns

Different parameters for u =1, ... M in Equation (3.7) in Section 3.4.2 were chosen for the structure of neurones in the hidden-1 layer, so that weights of the network can achieve better results. Partial **weights**, W (i, j), are shown in the horizontal axis in Figure 5.17 when M is equal to 2, 3, or 4 shown in the vertical axis. Figure 5.18 shows how weights can affect the recognition of test radicals.

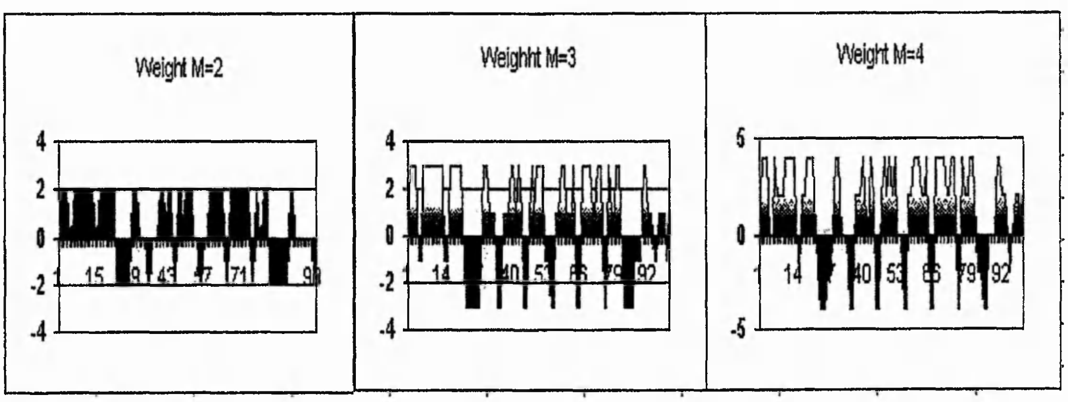


Figure 5.17 Different weights

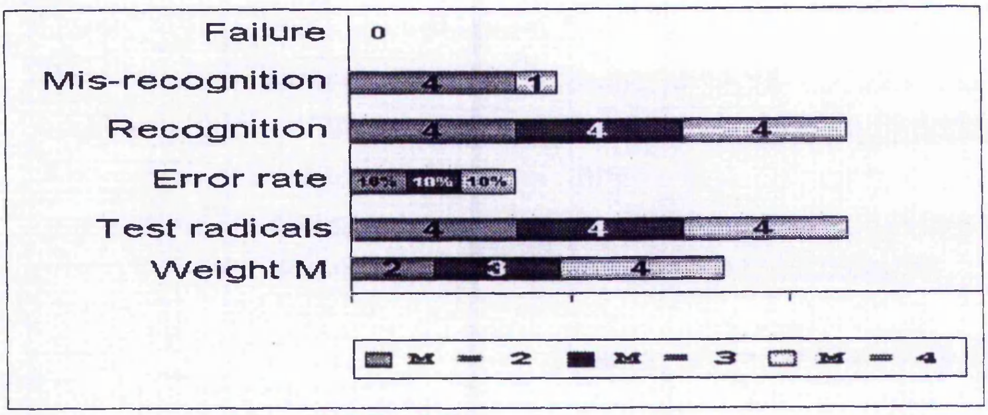


Figure 5.18 Effects of different weights

Three groups of test data with different error rates were used for examining the **error tolerance**. The error rates were 10%, 20% and 60%. The percentage of error rate indicates the scale of noise occurring in test data. Figure 5.19 gives the recognition results for this trial.

The results in Figures 5.18 and 5.19 show that a better recognition is achieved when the weight parameter M is 3.

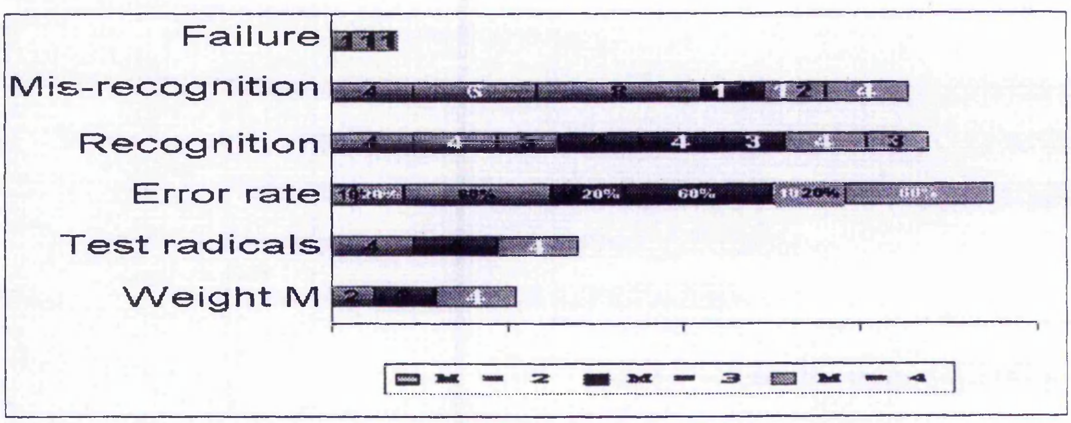


Figure 5.19 Results of test data with different error rates

Using the better weight parameter, i.e. M is equal to 3, 74 radicals in 24 categories were learnt and formed the 24 sub-nets in the hidden-1 layer of the associative memory neural network in the Recognition subsystem. The partial weights of the sub-nets are shown in Figure 5.20.

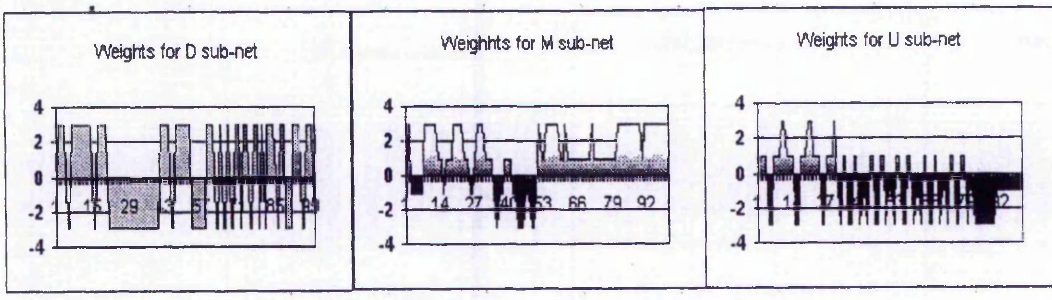


Figure 5.20 Weights for some sub-nets

5.5.3 Training Phase

Training the network was centered on the structure of neurons in the hidden-2 layer. Figure 5.21 shows three test groups of patterns in **different orders**.

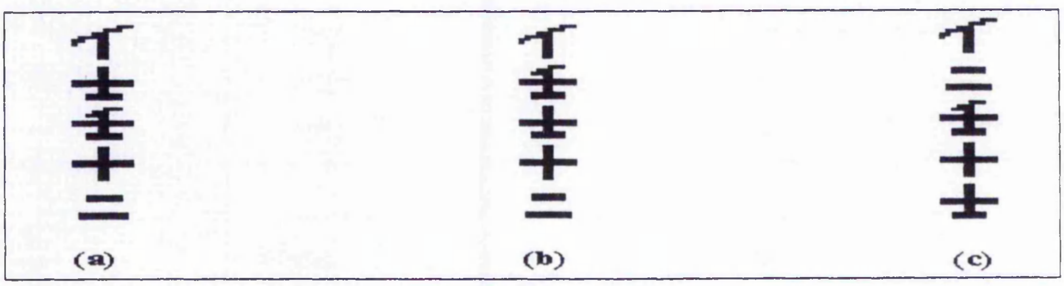


Figure 5.21 Patterns in different orders

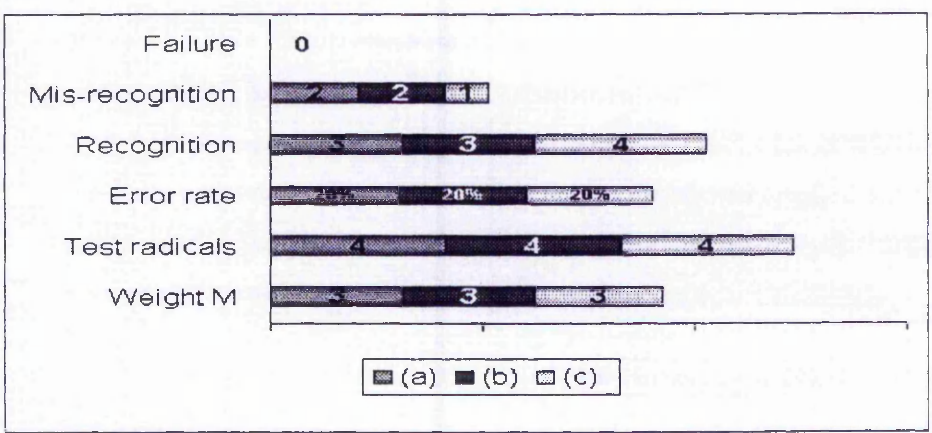


Figure 5.22 Results of different neuron structures

Figure 5.22 gives results of recognition when different structures of neurons in Figure 5.21 were used separately.

It is clear that there is an optimal scheme, such as the order of patterns in (c) in this case, for the neuron structure, even if the difference between the three groups of learning patterns is quite small.

5.5.4 Modification of the Network

Modification of the network was centred on the structure of neurons and improvement of global convergence. Results in Figure 5.23 show the convergence of local minima to a global minimum of radicals from the hidden-1 layer when parameters θ_i and φ_j in Equation (3.8) are set to 0.

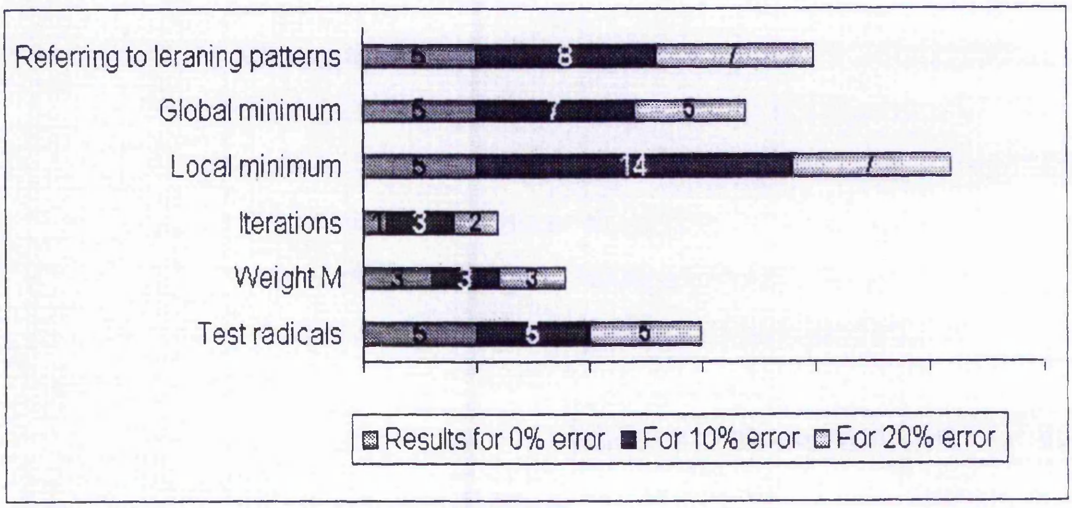


Figure 5.23 Test results with setting the parameters θ_i and φ_j to 0

After modifying the parameters θ_i and φ_j to $\frac{1}{2}\sum W_{ij}$, the mis-recognition rates of these tests were reduced. Figure 5.24 shows the results of the enhancement.

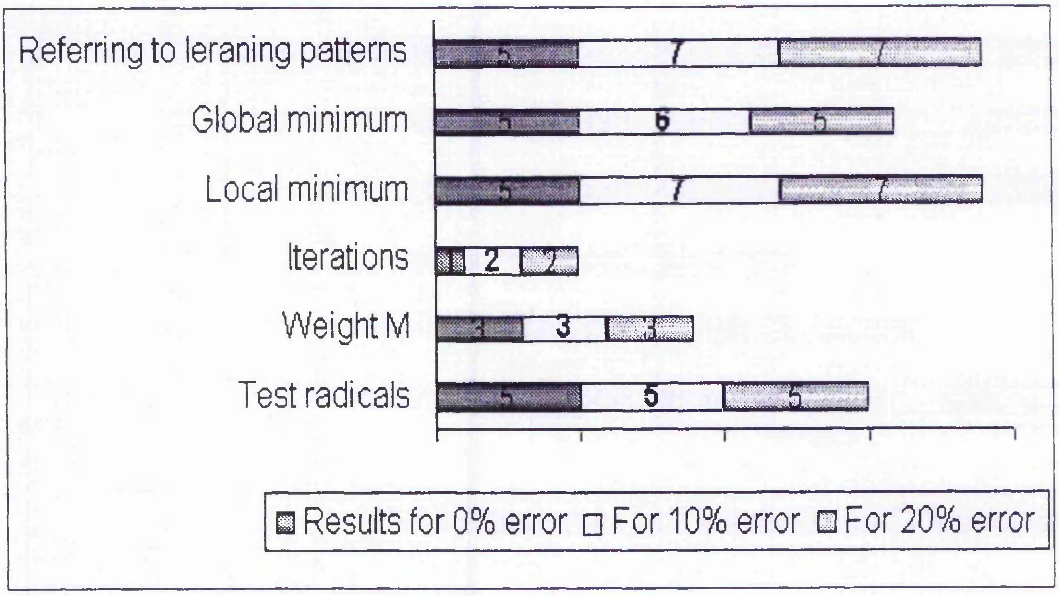


Figure 5.24 Test results with modifying the parameters θ_i and φ_j to $\frac{1}{2}\ \Sigma W_{ij}$

Compared to Figure 5.23, the convergence to a global minimum in Figure 5.24 has been improved and the number of iterations is reduced as well.

112 out of 120 radicals have been recognised by using the structure of the network, where 74 radicals were different from each other. Figure 5.25 shows the **recognition rate** of these **radicals** (both numbers indicated by the y-axis (vertical)) in different **categories** indicated by the x-axis (horizontal)). This figure was used for examining in which category radicals have been recognised successfully. Figure 5.26 shows the recognition frequency of the radicals in the categories (the number for frequency is indicated by y-axis (vertical), each radical in a category indicated by the x-axis (horizontal). The results in Figure 5.26 were used to test which radical is the most common radical appeared in characters.

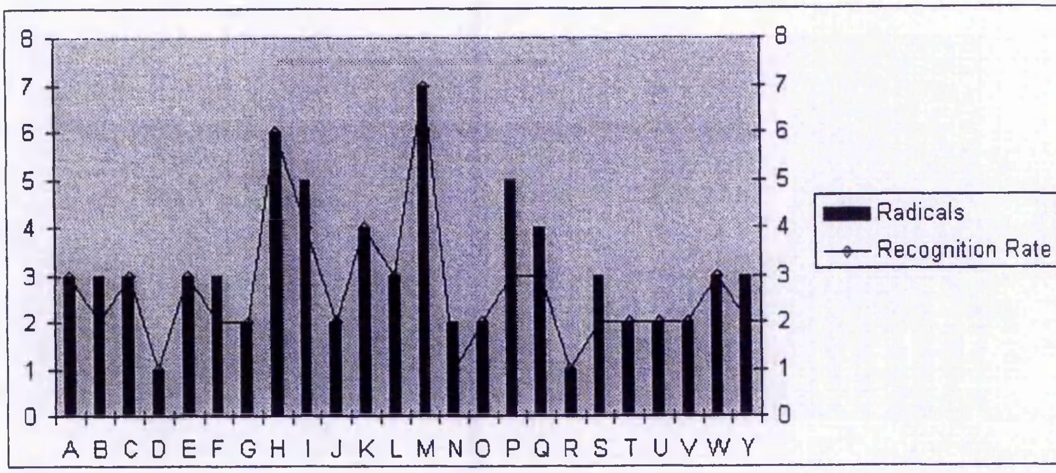


Figure 5.25 Recognition rate of radicals in different categories

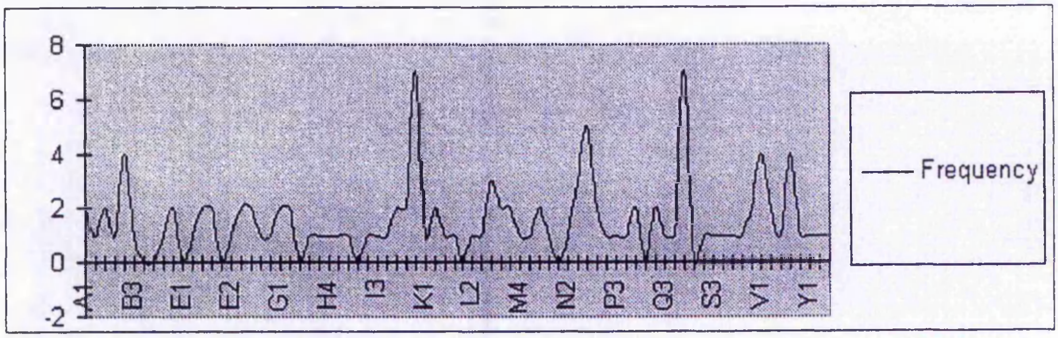


Figure 5.26 Recognition frequency of the radicals in the categories

5.5.5 Discussion on Recognition

The associative memory algorithm has advantages of quicker convergence speed and the capability of recognising error patterns. According to the statistics of outcomes from the above trials, the number of iterations is less than 4 for reaching recognition. The capability of recognising the error patterns can be high, up to as much as 60%.

The ambiguity of recognising patterns is basically caused by converging to a local minimum, especially when two patterns had the same reliability rate. There is also another case of ambiguity as shown in Figure 5.27. The difference between patterns (a) and (b) might be treated as noise causing ambiguity of patterns to occur.

In the Recognition subsystem, dividing the whole network into sub-nets has solved the limitation of associative memory. However, convergence to local minima can still occur in the hidden-1 layer.

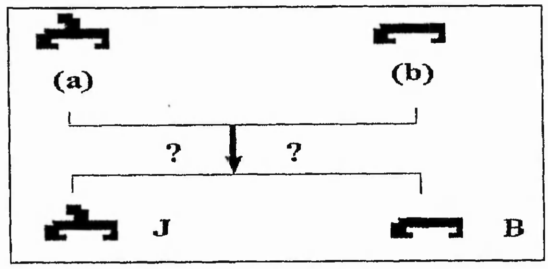


Figure 5.27 A case of ambiguity

5.6 Restructuring of Chain Codes

In order to ensure correctness of characters recognised, restructuring of chain codes and verification at the character level were carried out. Test data were focused on the applications of restructuring cues and ambiguous solutions in verification.

120 codes, i.e. radicals, have been used to examine the functions of restructuring and verification in the Restructuring subsystem. 46 chain codes have been successfully restructured and verified. Some of these are shown in Figure 5.28.

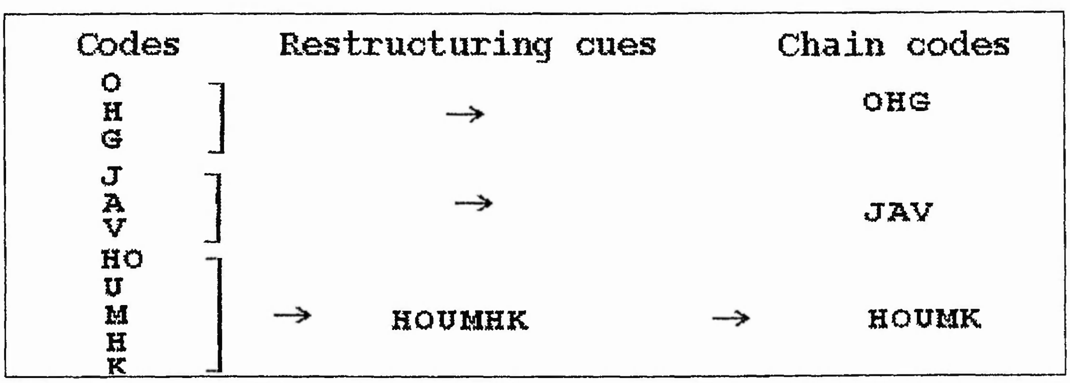


Figure 5.28 Restructuring of chain codes

5.6.1 Application of Restructuring Cues

When codes are required to be restructured into a chain code, all possible solutions for forming a chain code should be considered for the final verification.

Combination of codes allows different codes corresponding to a radical to be restructured into different chain codes. For instance, a character consists of three radicals and each radical has been given two different codes. In this case, 8 chain codes can be restructured. An example is illustrated in Figure 5.29.

Code1	Code2	Code3		Chain codes
B	A	K	→	BAK BAV BWK BWV
J	W	V		JAK JAV JWK JWV

$$^2C_1 * {}^2C_1 * {}^2C_1 = 8$$

Figure 5.29 Combinations of chain codes

Inspection of **illegal positions** of some radicals is another technique to enhance the final verification of characters. Radicals at illegal positions shown in Figure 5.30 were examined.

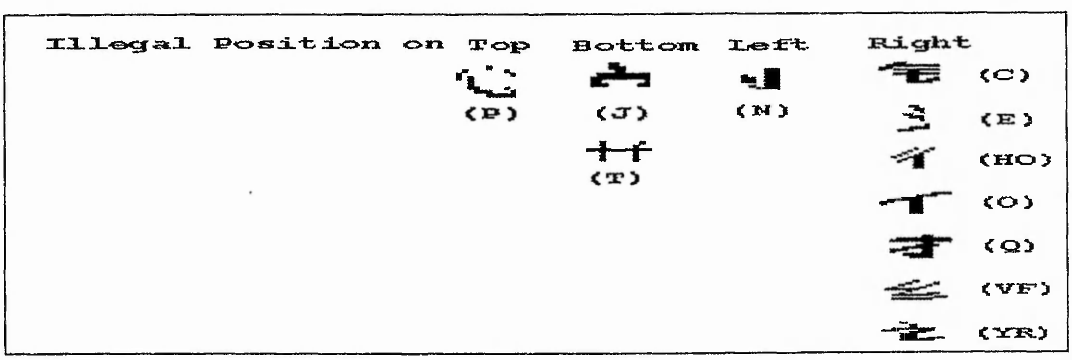


Figure 5.30 Illegal positions of some radicals

Examination of redundant codes ensures that a chain code only contains 5 codes or less. An example of a **code omitted** from a chain code is shown in Figure 5.31.

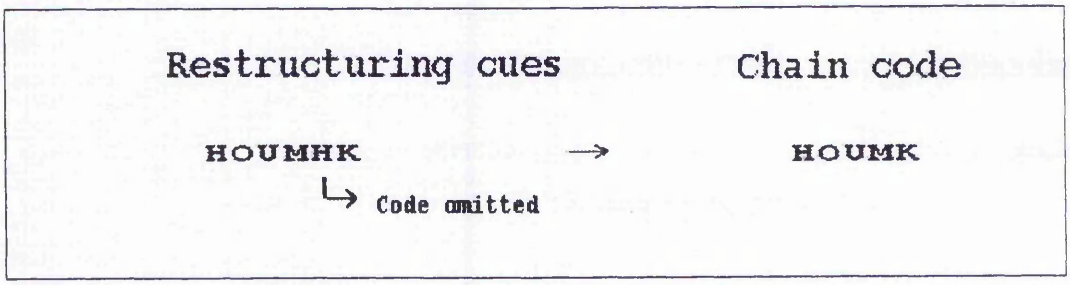


Figure 5.31 An example of codes omitted

5.6.2 Verification

The verification of a chain code includes matching the code to the code database and providing its quality rate.

75 codes with different errors for combining to 10 chain codes were used to test the verification. 63 out of 75 test codes have been successfully verified. The verification of chain codes using the code database could easily fail if a code in a chain code was not in a correct position, especially if the first code in a chain code was wrong. Some test results in Figure 5.32 show in which position a wrong code is most likely to cause a failure.

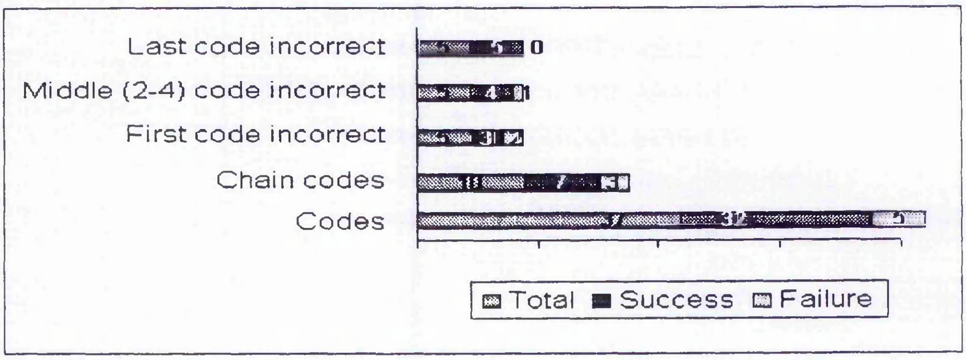


Figure 5.32 Test results on verifying chain codes

The quality rate of a chain code was used as a standard for evaluating the chain with respect to its reliability, matching ability to database and filling order. Some quality rates are shown in Figure 5.33.

Quality

0 0 0	1 0 0	1 2 1
good quality	mis-recognition	failure in order failure in matching

Figure 5.33 Results of the quality rate

5.6.3 Discussion on Restructuring

Apart from some mis-recognition results caused by the Recognition process, the Restructuring process has the capability of forming a chain code for verifying a character. Although the quality of a chain code has provided a reference for searching for possible solutions as results for matching to the database, the ambiguity of solutions occurred often, especially if the solutions were for the first code in a chain code. More human knowledge can be considered to enhance the restructuring cues.

5.7 Error Assessment

The experiments for error assessment were concerned with the effectiveness of methods used in the Postprocessing subsystem for detection, correction and rejection of errors occurring in a chain code. In the process of using the analysis of error probability, the experimentation was focused on the global examination of chain codes and assessment action.

The implementation of global assessment included comparison of quality and reliability rates, which came from the whole recognition process, of chain codes and making decisions for the assessment action when more than one chain referred to the same character. The assessment action was for correction and rejection of a chain code.

52 chain codes with different errors were used to examine the functions of global assessment and the actions of the Postprocessing subsystem. A success rate of 98% has been achieved. The test results are shown in Figure 5.34.

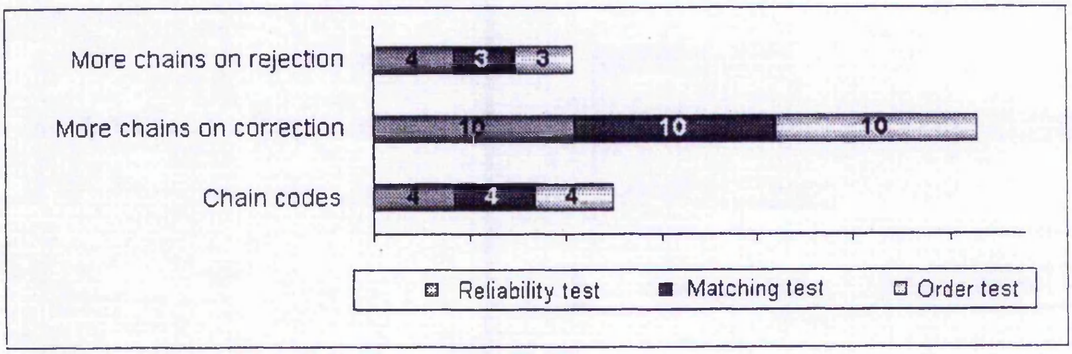


Figure 5.34 Results of error assessment

5.7.1 Global Assessment

The global assessment of chain codes was depended on the analysis of two rates: quality and reliability as illustrated in Figure 5.35. There were two stages involved: a chain code and more chain codes corresponding to a character. In the first stage, the assessment was only referring to the quality rate of a chain code. Two results could be derived: recognition or failure. All test results shown in Figure 5.36 had passed the assessment of the first stage.

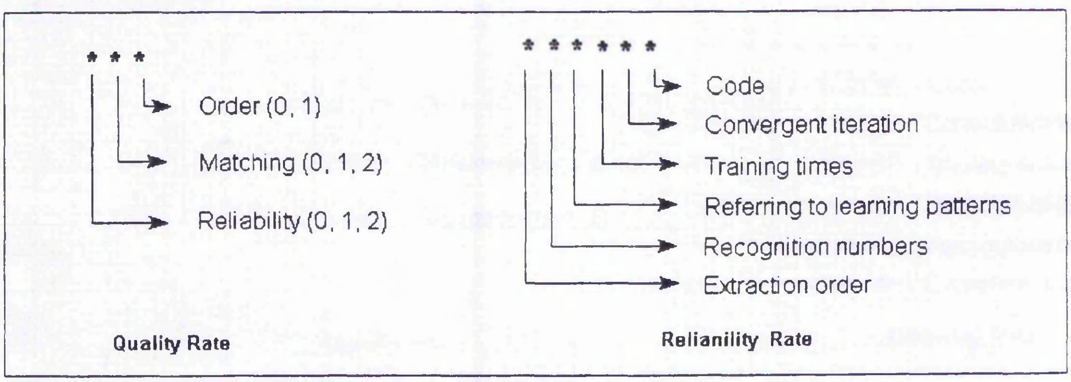


Figure 5.35 The quality and reliability rates

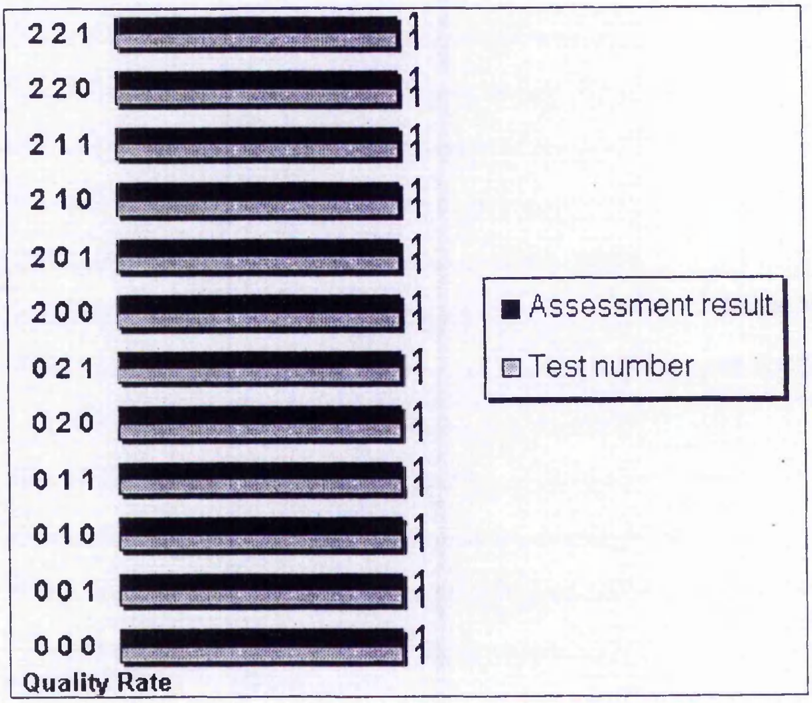


Figure 5.36 Results of a chain code corresponding to a character

In the second stage, the assessment had to take into account the quality and reliability rates together for comparison. The three possible outcomes were recognition, recognition with action, and failure. For the test results, the option of recognition with action was implemented by the assessment action.

5.7.2 Assessment Action

The implementation of assessment action was focused on correction and rejection of chain codes when these chain codes referred to only a character. The correction was able to get the best chain code. The rejection involved the decision to remove the worst one. 30 chain codes with different quality and reliability rates for the correction and 10 for the rejection were used for the assessment action. All test results had passed the assessment apart from one failure in the correction section. The test results and assessment rate are shown in Figure 5.37. The failed case will be discussed in Section

5.7.3.

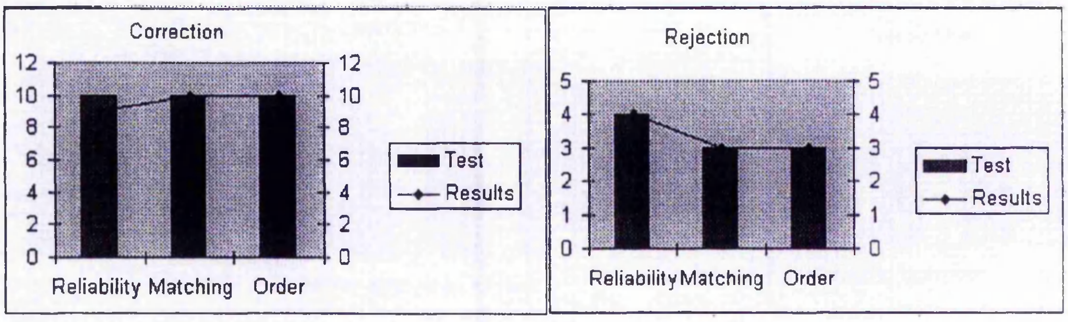


Figure 5.37 Results for the assessment action

5.7.3 Discussion on Correction

Analysing results in the error assessment stage, a failed result is caused by making a correction decision between two chain codes with the same probability, i.e. two codes in chain codes had the same quality and reliability rates, as shown in Figure 5.38. The action taken by the current process followed the policy of 'first come first served', which clearly meant that the code which came first was always taken. However, other methods can be further investigated even if such a case is quite rare.

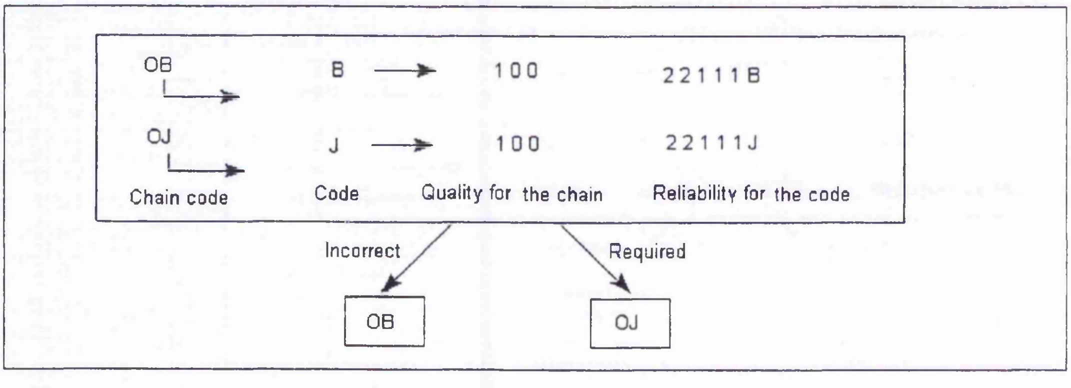


Figure 5.38 The failed case in the assessment action

Chapter 6
Conclusions and Future Work

6.1 Conclusions
6.2 Future Work

6.1 Conclusions

The work described in this thesis represents a significant advance towards the aims of the project, and the objectives of the project has been successfully achieved. As an original contribution, the method of three-layer hierarchy character-radical-stroke has been developed for the representation of the structure of Chinese characters. Another new achievement, inspired by the *Cang-Jie* method and greatly simplified the computer recognition process, is to use the process character-radical-chain code to translate a character from a 2-D pictorial format to a chain code for verification. These two new methods applied to a system called Chinese Character Recognition System have formed a solid base for potential research.

Compared to the existing methods of Chinese character recognition, the three-layer hierarchy offers the advantages: (a) having the capability of processing Chinese characters with a similar structure; (b) a more systematic representation of the internal topological structure of a character; (c) reducing the vocabulary of characters learnt by machine; and (d) using a chain code instead of a character to simplify the recognition process. The three-layer hierarchy method has a good structure representation in the classification of characters. The ambiguity of classified characters in the three-layer hierarchy is much lower than with the four-corner method. Furthermore, the method forms a solid base of further recognition work, such as hand-written Chinese characters in the *Caoshu* and *Xingshu* styles as distinct from the printed and standard script used in this project.

Having been investigated different knowledge representation techniques, two methods of fuzzy syntactic and fuzzy possibilistic reasoning have been applied to the CCRS. The latter method was also successfully applied to the recognition with 96% successful extraction rate.

In order to cope with the variability of radicals in a character, a Neocognitron neural network has been developed. Algorithms, advantages and disadvantages of the network have been investigated. Outcomes have shown that the network has strengths in the recognition of a shape for a fixed pattern rather than flexible one.

Recognising radicals was centred on the application of a neural network with associative memory function. The network possessed capabilities of learning different features of radicals, and then recognising them. The enhancement of the network at several stages has improved its recognition rate to 96%. Outcomes of the execution time and recognition rate have shown that the network was quite successful.

Restructuring of chain codes was a partial application of the *Cang-Jie* method to verify a character and to simplify the recognition process. Restructuring cues have been defined and a code database for supporting the verification was established. Outcomes have shown that the chain code method can easily and effectively be used for verification at the character level.

The postprocessing of a character created in the course of the described work was used for error assessment in recognition. An assessment scheme using the analysis of error probability has been investigated and implemented. Findings and outcomes of the application of the analysis method have shown its positive effects on the assessment of a character.

6.2 Future Work

The future work recommended here follows on from the difficulties to explore further improvements of the project.

Although work in the preprocessing stage has classified positions of radicals in a character, a case that allows omitted and difficult radicals in a character has not been considered yet. Basically, a character in such a case has a very complex structure and it is written in the complex style. Applying fuzzy possibilistic rules to such characters and more complex characters can be investigated in future development.

Normalisation of different radicals could focus on the question of dealing with variability and flexibility of a character. The method of Neocognitron neural networks has dealt with position invariance and shape recognition of a radical for a fixed pattern. Other advanced methods can be investigated to extend the capability of dealing with shapes from fixed to more flexible ones as the network is progressively developed.

The recognition of radicals can be further developed to reduce its mis-recognition cases. Investigating other methods for combining with the network can be considered to keep the strengths of learning features of a pattern while reducing the probability of convergence to a local minimum.

References

[Ald95] Al-Dabass, D., Ren, M. and Su, D., *An associative memory artificial neural networks system with a combined-radicals method for Chinese character recognition*, IEEE-ICONIP'95-Beijing: Proceedings of International Conference on Neural Network Information Processing, Beijing, China, 1995, Vol.2 of 2, pp.857-860.

[Bal91] Bala, J. And Wechsler, H., *Shape analysis using morphological processing and genetic algorithms*, Proceedings of Third International Conference on Tools for Artificial Intelligence TAI'91, 1991, pp.130-137.

[Ben93] Bengio, Y.A., *Connectionist approach to speech recognition*, Advances in Pattern Recognition Systems Using Neural Network Technologies, 1993, World Scientific, USA, pp.3-24.

[Che85] Cheung, Y.S. and Leung, C.H., *Chain-code transform for Chinese character recognition*, IEEE 1985 Proceedings of International Conference on Cybernetics Society, Tucson, USA, 1985, pp.42-45.

[Cho95] Cho, W., Lee, S.-W. and Kim, J. H., *Modelling and recognition of cursive words with hidden markov models*, Pattern Recognition, 1995, Vol.28, No.12, pp.1941-1953.

[Cow90] Cowell, J. R., *Character recognition in unconstrained environments*, PhD Thesis, The Nottingham Trent University, UK, 1990.

[Cox94] Cox, C., Blesser, B. and Eden, M., *The application of type font analysis to automatic character recognition*, Proceedings of 2nd International Joint Conference on Pattern Recognition, Copenhagen, 1994, pp.266-232.

[Cut91] Cutts, G., Structured Systems Analysis and Design Methodology (second edition), 1991, Blackwell Scientific Publications, UK.

[Day90] Dayhoff, J. E., Neural Network Architectures: An Introduction, 1990, Van Nostrand Reinhold, USA, pp.37-57.

[Dei90] Deitel, H. M., An Introduction to Operating Systems (second edition), 1991, Addison-Wesley Publishing Company Inc., USA.

[Dow88] Down, E., Clare, P. and Coe, I., Structured Systems Analysis and Design Method: Application and Context, 1988, Prentice Hall International Ltd., UK.

[Fre92] Freeman, J. A. and Skapura, D. M., Neural Networks: Algorithms, Applications, and Programming Techniques, 1992, Addison-Wesley Publishing Company, pp.373-393.

[Fuk82] Fukushima, K. and Miyake, S., *Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position*, Pattern Recognition, 1982, Vol.15, No.6, pp.455-469.

[Fuk88] Fukushima, K., *A neural network for visual pattern recognition*, Computer, 1988, Vol.21, Iss.3, pp.65-75.

[Fuk91] Fukushima, K. And Wake, N., *Handwritten alphanumeric character recognition by the neocognitron*, IEEE Transactions on Neural Networks, 1991, Vol.2, Iss.3, pp.355-365.

[Gon78] Gonzalez, R. C. and Thomason, M. G., Syntactic Pattern Recognition: Introduction, 1978, Addition-Wesley Publishing Company Inc., USA, pp. 12-17.

[Gov90] Govindan, V. K. and Shivaprasad, A. P., *Character recognition: a review*, Pattern Recognition, 1990, Vol.23, No.7, pp.671-683.

[Gue93] Guerfali, W. and Plamondon, R., *Normalising and restoring on-line handwriting*, Pattern Recognition, 1993, Vol.26, No.3, pp.419-431.

[Guo86] Guo, C. and Xuan, G., *Automatic recognition of printed Chinese character by four corner codes*, ICPR, 1986, Vol.8, pp.1013-1015.

[Ham92] Hamilton, R. J., Pringle, R.D. and Grant, P.M., *Syntactic techniques for pattern recognition on sampled data signals*, IEEE Proceedings E: Computers and Digital Techniques, 1992, Vol.139, Iss.2, pp.156-164.

[Hil93] Hildebrandt, T. H. and Liu, W., *Optical recognition of handwritten Chinese characters: advances since 1980*, Pattern Recognition, 1993, Vol.26, No.2, pp.205-225.

[Hop82] Hopfield, J., *Neural networks and physical systems with emergent collective computational abilities*, Proc. Ntl. Acad. Sci. USA, 1982, Vol.79, pp.2554-2558.

[Hsi92a] Hsieh, C.-C. and Lee, H.-J., *A probabilistic stroke-based viterbi algorithm for handwritten Chinese characters recognition*, Proceedings of 11th IAPR International Conference on Pattern Recognition, Conference B: Pattern Recognition Methodology and Systems, 1992, Vol.II, pp.191-194.

[Hsi92b] Hsieh, C.-C. and Lee, H.-J., *Off-line recognition of handwriting Chinese characters by on-line model-guided matching*, Pattern Recognition, 1992, Vol.25, No.11, pp.1337-1352.

[Hua87] Huang, J. S. and Chung, M.-L., *Separating similar complex Chinese characters by Walsh transform*, ICCC, 1987, Vol.7, pp.187-191.

[Hun93] Hung, K.-W. and Chan, W.-C., *Stroke encoded Chinese handwriting input system based on back-propagation networks*, Proceedings of TENCON'93: 1993 IEEE Region 10 Conference on 'Computer, Communication, Control and Power Engineering', 1993, Vol.2, pp.1106-1109.

[Hym91] Hyman, S.D., Vogl, T.P., Blackwell, K.T., Barbour, G.S., Irvine, J.M. and Alkon, D.L., *Classification of Japanese Kanji using principal component analysis as a preprocessor to an artificial neural network*, IJCNN-91-Seattle: International Joint Conference on Neural Networks, 1991, Vol.1, pp.233-238.

[Imp91] Impedovo, S., Ottaviano, L. and Occhinegro, S., *Optical character recognition - a survey*, International Journal of Pattern Recognition and Artificial Intelligence, 1991, Vol.5, No.1&2, pp.1-24.

[Joh88] Johnson, K., Daniell, C. and Burman, J., *Feature extraction in the neocognitron*, IEEE International Conference on Neural Networks, 1988, Vol.2, pp.117-126.

[Ker88] Kernighan, B. W. And Ritchie, D. M., The C Programming Language (second edition), 1988, AT&T Bell Laboratories, Prentice Hall, USA.

[Kli88] Klir, G. J. and Folger, T. A., Fuzzy Sets, Uncertainty and Information, 1988, Prentice Hall, USA, pp.138-227.

[Kli95] Klir, G. J. and Yuan, B., Fuzzy Sets And Fuzzy Logic: Theory And Applications, 1995, Prentice Hall PTR, USA, pp.200-208, pp.369-374.

[Kos87] Kosko, B., *Adaptive bidirectional associative memories*, Applied Optics, 1987, 26(23), pp.4947-4960.

[Kos88] Kosko, B., *Bidirectional associative memories*, IEEE Transactions on Systems, Man, and Cybernetics, 1988, 18(1), pp.49-60.

[Kov92] Kovacs, Z. M., Guerrieri, R., *Computer recognition of hand-written characters using the distance transform*, Electronics Letters, 1992, Vol.28, Iss.19, pp.1825-1827.

[Kru94] Kruse, R., Gebhardt, J. and Klawonn, F. Foundations of Fuzzy Systems, 1994, John Wiley & Sons Ltd, UK, pp.81-155.

[Ku92] Ku, K.-M. and Chiu, P.P.K., *A combined method for stroke segmentation and feature extraction for handwritten characters*, Proceedings of TENCON'92: 'Technology Enabling Tomorrow' 1992 IEEE Region 10 International Conference on 'Computers, Communications and Automation towards the 21st Century', 1992, Vol.1, pp.277-281.

[Ku93] Ku, K.-M. and Chiu, P.P.K., *An expert system for recognising hand-written Chinese characters*, Proceedings of TENCON'93: 1993 IEEE Region 10 Conference on 'Computer, Communication, Control and Power Engineering', 1993, Vol.4, pp.510-513.

[Ku94] Ku, K.-M. and Chiu, P.P.K., *Fast stroke extraction method for handwritten Chinese characters by cross region analysis*, Electronics Letters, 1994, Vol.30, Iss.15, pp.1210-1212.

[Kuo94] Kuo, J. B. and Mao, M. W., *A radical-partitioned neural network system using a modified sigmoid function and a weight-dotted radical selector for large-volume Chinese characters recognition VLSI*, Proceedings of 1994 IEEE International Symposium on Circuits and Systems: Non-linear Circuits and Systems (NCS) Neural Systems (NEU), 1994, Vol.6 of 6, pp.331-334.

[Lam92] Lam, P., *On the standardisation of the radicals of Chinese characters*, Journal of MACRO Linguistics, 1992, No.2, Household World Publisher, pp.156-163.

[Lei87] Leigh, D., Hatter, D. and Newton, R., Software: Design, Implementation and Support, 1987, Paradigm, UK.

[Lin93] Lin, C.-K., Fan, K.-C. and Lee, F. T.-P., *On-line recognition by deviation-expansion model and dynamic programming matching*, Pattern Recognition, 1993, Vol.26, No.2, pp.259-268.

[Liu88] Liu, Y.J and Tai, J.W., *The theory and practice of on-line Chinese character recognition*, Journal of Chinese Information Processing, 1988, Vol.2, No.4, pp.1-13. (In Chinese)

[Liu93] Liu, Y.-M., Introduction of Cang-Jie Coding Method, 1993, Ru Lin Ltd., Taiwan. (In Chinese)

[Man86] Mantas, J., *An overview of character recognition methodologies*, Pattern Recognition, 1986, Vol.19, No.6, pp.425-430.

[Man87] Mantas, J., *Methodologies in pattern recognition and image analysis - a brief survey*, Pattern Recognition, 1987, Vol.20, No.1, pp.1-6.

[Mao92] Mao, M. W. and Kuo, J. B., *A coded block adaptive neural network system with a radical-partitioned structure for large-volume Chinese characters recognition*, Neural Networks, 1992, Vol.5, pp.835-841.

[Mcn94] McNeil, F. M. and Thro, E., Fuzzy Logic: A Practical Approach, 1994, Academic Press Inc., USA, pp.15-18.

[Min90] Minnix, J. I., McVey, E. S. and Inigo, R. M., *Modified neocognitron with position normalizing preprocessor for translation invariant shape recognition*, Proceedings of IJCNN International Joint Conference on Neural Networks, 1990, Vol.1, pp.395-400.

[Nad93] Nadal, C. and Sue, C. Y., *Applying human knowledge to improve machine recognition of confusing handwritten numerals*, Pattern Recognition, 1993, Vol.26, No.3, pp.381-389.

[Nag88] Nagy, G., *Chinese character recognition: a twenty-five-year retrospective*, Proceedings of 9th International Conference on Pattern Recognition, 1988, Vol.1, pp.163-167.

[Nak92] Nakayama, K. and Chigawa, W., *Japanese Kanji character recognition using cellular neural networks and modified self-organising feature map*, CNNA'92 Proceedings of Second International Workshop on Cellular Neural Networks and Their Application, 1992, pp.191-196.

[Nel91] Nelson, M. M. and Illingworth, W. T., A Practical Guide to Neural Nets, Addison Wesley Publishing Company, Inc., USA, 1991, pp.67-71.

[Nou90] Nouboud, F. and Plamondon, R., *On-line recognition of handprinted characters: survey and beta tests*, Pattern Recognition, 1990, Vol.23, No.9, pp.1031-1044.

[Nou91] Nouboud, F. and Plamondon, R., *A structural approach to on-line character recognition: system design and applications*, International Journal of Pattern Recognition and Artificial Intelligence, 1991, Vol.5, No.1&2, pp.311-335.

[Pal92] Pal, S. K., *Fuzziness image information and scene analysis*, An Introduction to Fuzzy Logic Applications in Intelligent Systems, edited by Ronald R.Yager and Lotfi A. Zadeh, 1992, Kluwer Academic Publishers, USA, pp.147-183.

[Par74] Parks, J.R. et al, *An articulate recognition procedure applied to handprinted numerals*, Proceedings of 2nd International Joint Conference on Pattern Recognition, Copenhagen, 1974, pp.416-420.

[Poo93] Poon, J.C.H., Man, G.M.T. and Chan, C.H., *The application of fuzzy grammar to handwritten character recognition*, Proceedings of TENCON'93: 1993 IEEE Region 10 Conference on 'Computer, Communication, Control and Power Engineering', 1993, Vol.4, pp.502-505.

[Pre87] Pressman, R. S., Software Engineering: A Practitioner's Approach (second edition), 1987, McGraw-Hill Book Company, Singapore, pp.164-208.

[Ren94] Ren, M., *An associative memory artificial neural network system with a combined-radical structure for Chinese character recognition*, 1994, internal report, Department of Computing, The Nottingham Trent University, UK.

[Ren95a] Ren, M., Su, D. and Al-Dabass, D., *An associative memory artificial neural network system*, ECAC'95-London: Proceedings of European Chinese Automation Conference, London, UK, 1995, pp.91-96.

[Ren95b] Ren, M., *Application of ANN to Chinese character recognition*, 1995, a report required by the committee of the MPhil/PhD degree registration in The Nottingham Trent University, UK, supervised by Dr David Al-Dabass, Dr Daizhong Su and Dr Lindsay L. Evett.

[Ren96a] Ren, M., Al-Dabass, D. and Su, D., *Using a syntactic method to partition a Chinese character*, The **Caledonian** International Engineering Journal, 1996, 96(1), pp.51-60.

[Ren96b] Ren, M., Al-Dabass, D. and Su, D., *A three-layer hierarchy for representing Chinese characters*, Research and Development in Expert Systems XIII: Proceedings of Expert Systems 96, the Sixteenth Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, UK, 1996, pp.137-146.

[Ren97a] Ren, M., Al-Dabass, D. and Su, D., *A fuzzy/possibilistic-reasoning AM-architecture for recognising radicals in Chinese characters*, ISFL'97-Zürich: Proceedings of Second International ICSC Symposium on Fuzzy Logic and Application, Zürich, Switzerland, 1997, pp.265-271.

[Ren97b] Ren, M., Al-Dabass, D. and Su, D., *A hierarchical approach to fuzzy possibilistic-reasoning for recognising Chinese characters*, International Journal of Computer Mathematics, 1997, Vol.64, No.1&2, pp.17-34.

[Ren97c] Ren, M., *Algorithms for off-line recognition of Chinese characters*, 1997, transfer report from MPhil to PhD, The Nottingham Trent University, UK.

[Rot93] Rothstein, M., Rosner, B., Senatore, M. and Mulligan, D., Structured Analysis and Design for the CASE User, Ranade, J. Consulting editor, 1993, McGraw-Hill Inc., USA.

[Rtt93] Real Time Telemetry Systems Group, *Nottingham Trent University Graphics (NTUG) on Graphical Kernel System (GKS)*, 1993, internal technical report, Department of Computing, The Nottingham Trent University, UK.

[Scu91] Scurfield, E., Chinese, 1991, Hodder and Stoughton, UK.

[Shr84] Shridhar, M. and Radreldin, A., *High accuracy character recognition algorithm using Fourier and topological descriptors*, Pattern Recognition, 1984, Vol.17, No.5, pp.515-524.

[Siy74] Siy, P. and Chen, C.S., *Fuzzy logic for handwritten numeral character recognition*, IEEE Transactions on Systems, Man, and Cybernetics, 1974, pp.570-575.

[Sun91] Sun Release 4.1, *User commands*, 1991.

[Tap90] Tappert, C. C., Suen, C. Y. and Wakahara, T., *The state of the art in on-line handwriting recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, Vol.12, No.8, pp.787-808.

[Tse92] Tseng, L.Y. and Chuang, C.T., *An efficient knowledge-based stroke extraction method for multi-font Chinese characters*, Pattern Recognition, 1992, Vol.25, No.12, pp.1445-1458.

[Wak92] Wakahara, T., Murase, H. and Odaka, K., *On-line handwriting recognition*, Proceedings of The IEEE, 1992, Vol.80, No.7, pp.1181-1194.

[Wan93] Wang, Y. F., Cruz, J. B. Jr. and Mulligan, J. H. Jr., *Multiple training concept for back-propagation neural networks for use in associative memories*, Neural Networks, 1993, Vol.6, pp.1169-1175.

[Wan94] Wang, T., *Improving recall in associative memories by dynamic threshold*, Neural Networks, 1994, Vol.7, No.9, pp.1379-1385.

[War88] Ward, J.R. and Kuklinski, T., *A model for variability effects in handprinted with implications for the design of handwriting character recognition systems*, IEEE Transactions on Systems, Man, and Cybernetics, 1988, Vol.18, pp.438-451.

[Wil93] Wilson, C.L., *Evaluation of character recognition systems*, Neural Networks for Processing III Proceedings of the 1993 IEEE-SP Workshop, 1993, pp.485-496.

[Woo96] Wood, J., *Invariant pattern recognition: a review*, Pattern Recognition, 1996, Vol.29, No.1, pp.1-17.

[You88] Young, Y., *Handprinted Chinese character recognition via neural networks*, Pattern Recognition Letter, 7, 1988, pp.19-25.

[You89] Yourdon, E., Modern Structured Analysis, 1989, Prentice Hall Inc., USA.

[Zad65] Zadeh, L.A., *Fuzzy sets*, Information and Control, 1965, Vol.8, pp.338-353.

[Zad92] Zadeh, L. A., *Knowledge representation in fuzzy logic*, An Introduction to Fuzzy Logic Applications in Intelligent Systems, edited by Ronald R. Yager and Lotfi A. Zadeh, 1992, Kluwer Academic Publishers, USA, pp.1-25.

[Zho86] Zhou, Y., *Modernisation of the Chinese language*, International Journal of the Sociology of Language, 1986, Vol.59, Mouton de Gruyter, Amsterdam, pp.7-23.

[Zho92] Zhou, Y., *Language planning in China: understanding and misunderstanding*, Journal of MACRO Linguistics, 1992, No.1, Household World Publisher, pp.57-64.

[Zim87] Zimmermann, H.-J., Fuzzy Sets, Decision Making, and Expert Systems, 1987, Kluwer Academic Publishers, USA, pp.235-237.

Appendix A. Supporting Tools

UNIX, as a trademark of American Telephone and Telegraph Company (AT&T) Bell Laboratories and from Uniplexed Information and Computing Services (UNICS) at its early stage of development, is a collection of user and system programs, i.e. an operating system, and was designed in the late 1960s and early 1970s [Dei90]. UNIX system can provide a friendly environment for program development and text processing, such as easy to combine programs with one another, encouraging a modular, tool-oriented building-block approach to program design, files management, supporting different developments. The SunOS Release 4.1 lint library is a super-set of the 4.3 BSD lint library. It includes all of the 4.3 BSD functionality; most of System V release 3.2 functionality; as well as extensive additional functionality in the networking and file system areas.

X window is a network transparent window system, developed at Massachusetts Institute of Technology (MIT), which runs in a wide range of computing and graphics machines. The X window system servers run on computers with bitmap displays. The server distributes user input and accepts output requests from various client programs through a variety of different inter-process communication channels. It also supports overlapping hierarchical sun windows, text and graphics operations, on both monochrome and colour display [Sun91].

X view (X window-system-based visual/integrated environment for workstations) is an Open Look user-interface toolkit that supports development of interactive, graphics-based applications running under the X window system [Sun91].

C programming language, which was developed in 1972 and published its first standard by the American National Standards Institute (ANSI) in 1989, is an unambiguous, machine-independent and general-purpose programming computer language that features economy of expression, modern control flow and data structure, and a rich set of operators [Ker88].

CUA is a set of standards for formatting terminal screen output. The standards cover the placement of specific types of data characteristics on terminal and workstation screens to increase consistency and intuitiveness [Rot93]. The standards used in CCRS include pull-down menu with the action bar, pop-up windows, panel body and radio buttons and so on.

GKS is an international standard for computer graphics designed by International Standards Organisation (ISO) in 1985. The standard defines a graphical system that supports a wide range of drawing functions independent of programming language or hardware. Nottingham Trent University Graphics package (NTUG) is a package of graphics library developed on GKS. The library includes 35 functions with two language versions, C and FORTRAN, for drawing and displaying graphics on X terminal clients [Rtt93].

Appendix B. Developed Software

This appendix briefly describes the software developed in the course of the reported work. The software has been managed by CCRS Management subsystem using the option of viewing programs or files in the main menu of CCRS. Profiles of the programs and files are listed and their source programs can be modified in the circumstances.

B.1 Programs

The source programs are stored on the disk of the departmental UNIX file server in the directory **~mar/xview/source/ccrs**. Executing programs are put in the working area: **~/mar/xview**.

Figure B.1 shows the structure of programs in the main system of CCRS.

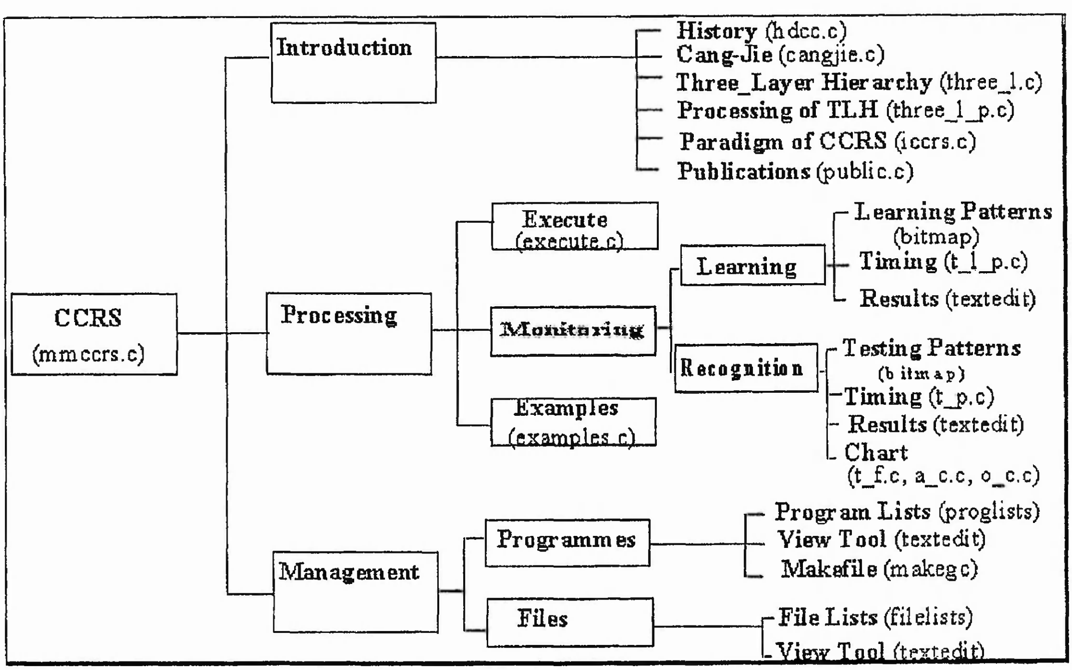


Figure B.1 The structure of CCRS programs in the main system

Figure B.2 shows the structure of CCRS executing problems in the five subsystems.

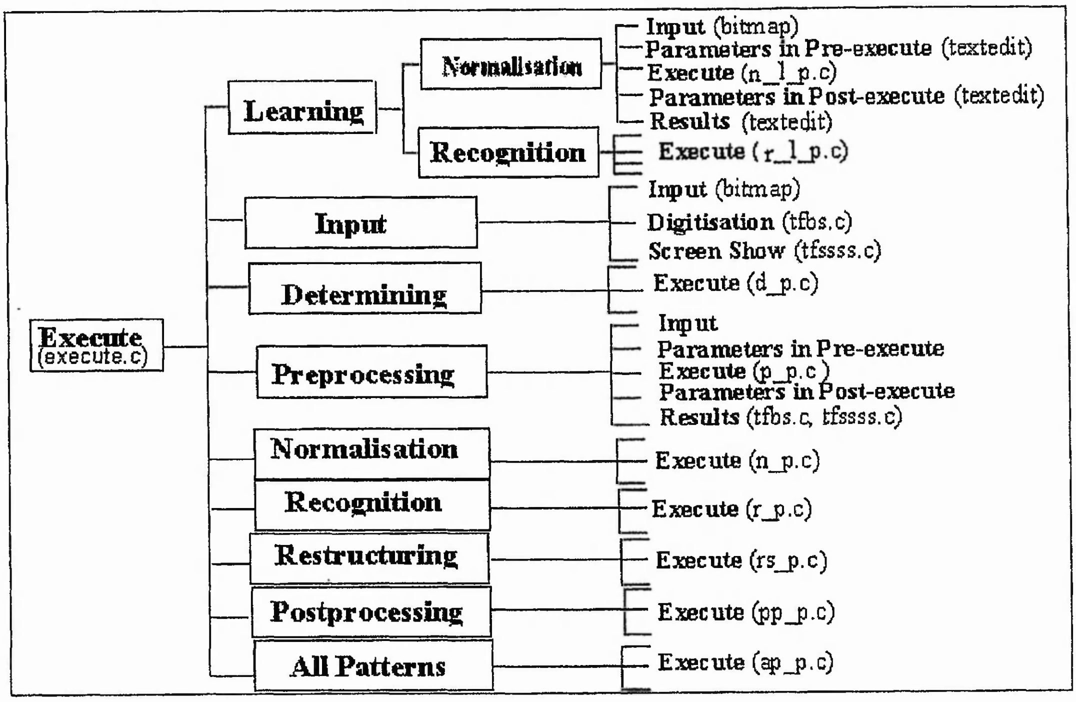


Figure B.2 The structure of executing programs for the five subsystems

Figure B.3 shows the structure of CCRS examples.

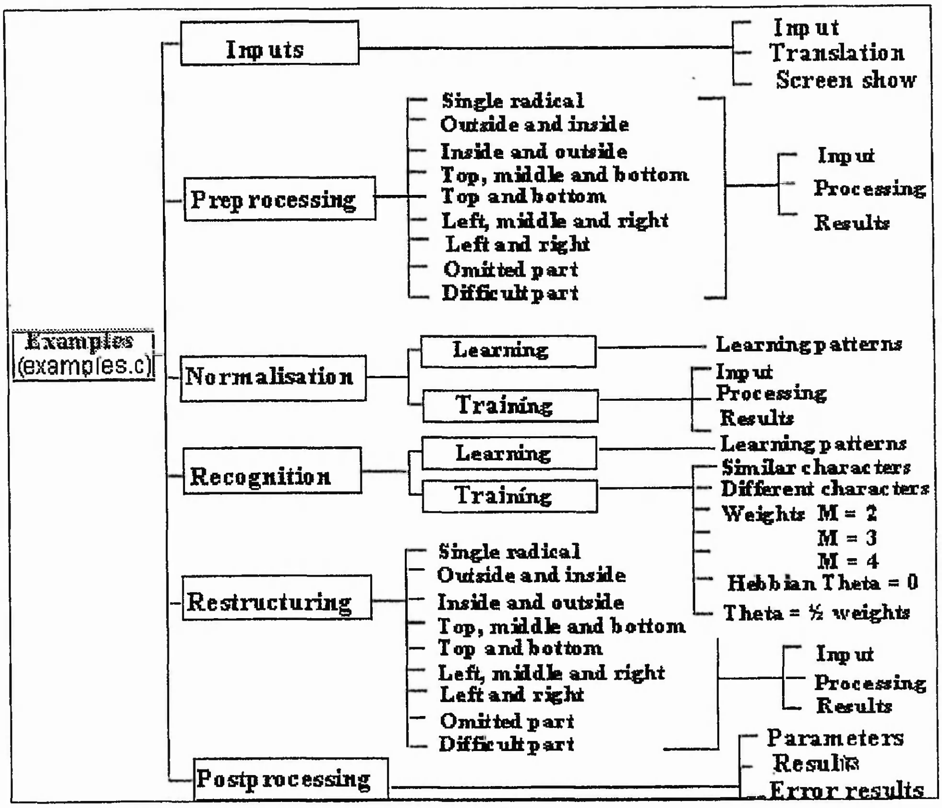


Figure B.3 The structure of CCRS examples

The definition of programs is described as follows.

a_c.c:	drawing time chart for integrating recognition of characters;
ap_p.c:	integrating recognition;
cangjie.c	illustration of the *Cang-Jie* method;
d_p.c:	determining a pattern;
examples.c	examples of CCRS;
execute.c:	executing program of CCRS;
filelists:	lists of CCRS files;
hdcc.c:	historical development of Chinese characters;

iccrs.c introduction of CCRS processing;

makegc: makefile of CCRS;

mmccrs.c: the main system program of CCRS;

n_l_p.c learning patterns in the Normalisation process;

n_p.c: normalisation of radicals;

o_c.c: a chart for time utilisation of recognising a character;

p_p.c: preprocessing for extraction;

pp_p.c: postprocessing for error assessment;

proglists: lists of CCRS programs;

public.c publications of CCRS;

r_l_p.c: learning patterns in the Recognition process;

r_p.c: recognition of radicals;

rs_p.c: restructuring of chain codes;

t_f.c: translating data for drawing time chart;

t_l_p.c: time monitoring process for learning patterns;

t_p.c: time monitoring process for recognising characters;

tfbs.c translating data from bitmap to digitised format;

tfssss.c display a pattern on screen;

three_l.c introduction of three-layer hierarchy;

three_l_p.c: illustration of three-layer hierarchy process.

Some programs are listed as follows.

i) Building up a menu for Introduction of CCRS in mmccrs.c

```
/* Building up the main menu for Introduction of CCRS */
/* set a menu in horizontal */
    panel = (Panel)xv_create(frame, PANEL,
        PANEL_LAYOUT,            PANEL_HORIZONTAL,
        WIN_WIDTH,
        WIN_EXTEND_TO_EDGE,
        NULL);

   xv_set(canvas_paint_window(panel), NULL);


  /* Create the menul before the panel button  */
```

```
      menu1 = (Menu)xv_create(NULL, MENU,
      MENU_STRINGS,   "History", "Cang-Jie", "Three-Layer Hierarchy",
                            "Processing of TLH",
                            "Paradigm of CCRS", "Publications", NULL,
            MENU_NOTIFY_PROC,  introduction_notify_proc, NULL,
        NULL);

/* associate the menu to the panel window for easy retrival */
  xv_set(canvas_paint_window(panel),
      WIN_CONSUME_EVENTS,  WIN_MOUSE_BUTTONS, NULL,
      WIN_EVENT_PROC,      menu_event_proc,
      WIN_CLIENT_DATA,     menu1,
      NULL);

  /* create the panel button */
  xv_create(panel, PANEL_BUTTON,
      PANEL_LABEL_STRING,     "Introduction",
      PANEL_ITEM_MENU,         menu1,    /* attach menu to button */
      NULL);
```

ii) Partial possibilistic rules for checking the shape of a radical in the outside and inside case in p_p.c

```
/* Some rules for the outside-to-inside case in Preprocessing*/
    if ((top==1)&&(bottom==1)&&(left==1)&&(right==1))
      {code =0; goto partition_check;}

/* both cases: the top line is open */
    if ((top==0)&&(bottom==1)&&(left==1)&&(right==1))
      {code =11; goto partition_check;}

/* both cases: the bottom line is open */
    if ((top==1)&&(bottom==0)&&(left==1)&&(right==1))
      {code =12; goto partition_check;}
partition_check:
```

iii) Counting weights for the associative memory neural network in ccrspro3.h called by r_p.c

```
/* PX: array of input data  */
/* PY: array of output data */
/* PW: weights              */

/* count weights */
int Count_weights(PX, PY, PW)
   float PX[ISIZE][JSIZE], PY[ISIZE][JSIZE], PW[Wsize][Wsize];
{ int i, j, i1, j1;

   for(i1=0; i1<N1width; i1++)
    {for(j1=0; j1<N1height; j1++)
      {for (i=0; i<N1width; i++)
        {for (j=0; j<N1height; j++)
             {PW[i1*N1width+j1][i*N1height+j]=
         PW[i1*N1width+j1][i*N1height+j]+PX[i][j]*PY[i1][j1];}
        }
      }
    }
  return 0;
```

```
} /* end of Count_weights  */
```

B.2 Files

All files used by the CCRS are managed by the Management subsystem of CCRS, which lists their formats, usage and profiles. During the performance of CCRS, files created in executing area are entitled as **ccrs%.&&****, where **%** stands for the category of radicals for some files, **&&** indicate the property of files, ** stands for a series number of the order of radicals in a character. An illustration of files used in CCRS is shown in Figure B.4.

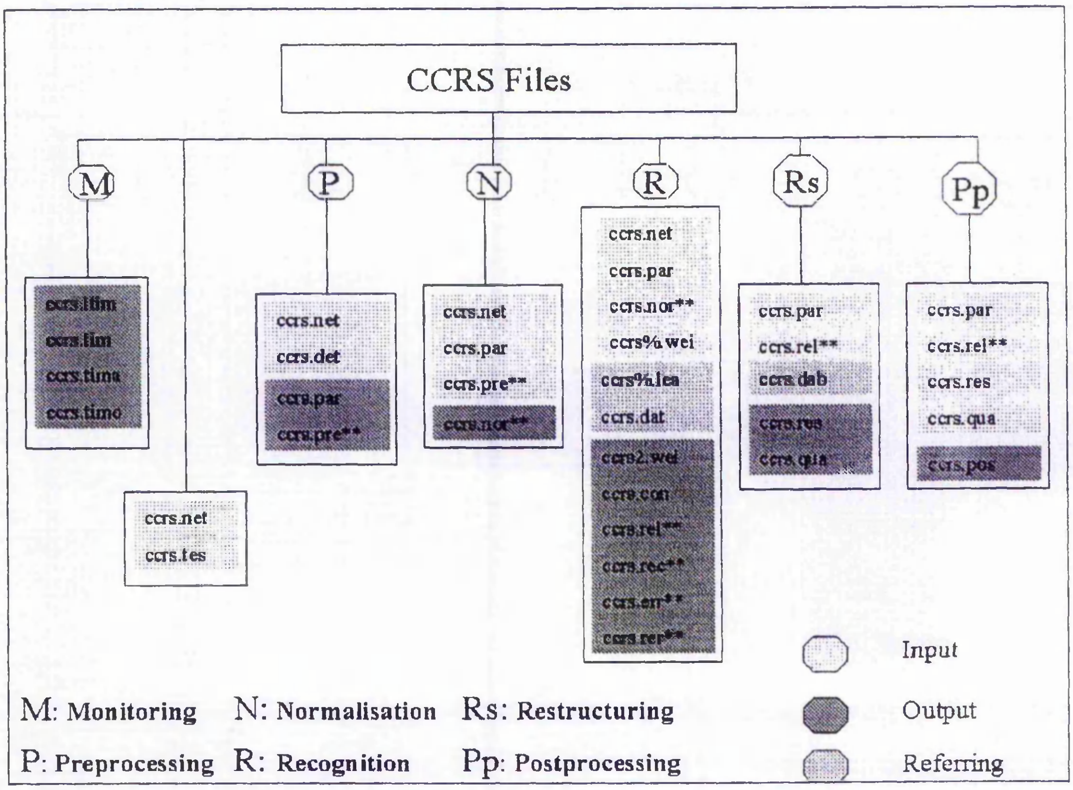


Figure B.4 Illustration of files in CCRS

ccrs.con:	recognised results from the hidden-1 layer in the Recognition subsystem;
ccrs.dab:	code database;
ccrs.dat:	translating radicals into codes;
ccrs.det:	a character determining as input for the Preprocessing;
ccrs.err**:	unrecognised patterns for a radical in the ** position of a character;
ccrs%.lea:	learning patterns, % can be from A to Z;
ccrs.ltim:	time statistics for learning patterns;
ccrs.net:	parameters of the CCRS system;
ccrs.nor**:	normalised radicals in the ** position of a character;
ccrs.par:	the method of deconstructed a character into radicals;
ccrs.pos:	recognised results in the Postprocessing subsystem;
ccrs.pre**:	extracted results in the Preprocessing;
ccrs.qua:	quality rate for chain codes;
ccrs.rec**:	recognised results in the Recognition subsystem;
ccrs.rel**:	reliability rate for a radical;
ccrs.rer**:	converged results in the Recognition subsystem.
ccrs.res:	restructuring results in the Restructuring subsystem;
ccrs.tes:	test characters;
ccrs.tim:	monitoring results of time statistics;
ccrs.tima:	the chart of time utilisation for all test characters.
ccrs.timo:	the chart of time utilisation for a certain test character;
ccrs%.wei:	weights for the Recognition subsystem, % can be from A to Z.
ccrs2.wei:	weights in hidden-2 layer for the Recognition subsystem.

B.3 Code Database

The code database is stored in the directory **~/mar/xview/p3data/c-database/ccrs.dab** as an ASCII text file. It is used in the Restructuring subsystem for searching chain codes. The structure of the database is presented in Figure B.5.

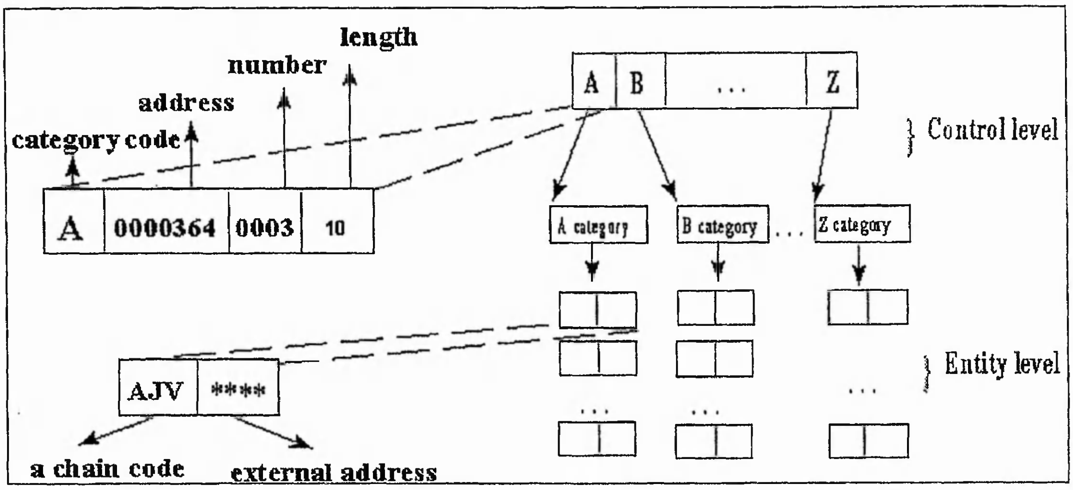


Figure B.5 The structure of the code database

Currently, chain codes included in the database are shown in Table B.1.

Category	Chain Code	Category	Chain Code	Category	Chain Code
A	AJV, ANR, AWLE	K		U	UK
B	BG, BMR, BQ	L		V	VFBB, VFQJL, VFYK, VPP
C	CJ	M	MCW, MGB, MGOII, MOB	W	WD, WHG, WKS, WMMR
D	DG	N		X	
E	EQHL	O	OB, OF, OG, OHG, OJ, OMNN	Y	YRBP, YRMMR
F	FG, FK	P	PI, PII, PU	Z	
G	GG, GP	Q	QM		

H	HGOII, HNI, HOUMK	R	RHG, RKSR, RVE		
I	ILIL	S	SK, SMC, SMR		
J	JAV, JFB, JLW, JMAM, JMD, JMIG	T	TP, TW		

Table B.1 Data in the code database

Appendix C. Experimental Data and Outcomes

This appendix provides a description of the data and test results in the course of this work. All data stored in bitmap format and their results in digitised or chain code format. They can be translated into different formats: bitmap, digitised and display, supported by the programs: **tfbs** and **tfssss**, mentioned in Appendix B: Developed Software.

C.1 Data

All original data are formatted by using bitmap tool and stored in a separate subdirectory according their features of structure, i.e. outside, inside, left, right and so on. At present, there are 9 different subdirectories for storing different data. The name of subdirectory will indicate the structure features of characters. Each character is put in a bitmap file and entitled by its '*pinyin*' of Chinese pronunciation. These subdirectories are **~mar/xview/p3data/@@/*.bitmap**, where * stands for the name of a file and @@ are:

outside:	characters in the structure from outside to inside;
inside:	characters in the structure from inside to outside;
top-m-b:	characters in the structure from top, to middle, then bottom;
top:	characters in the structure from top to bottom;
left-m-r:	characters in the structure from left, to middle, then right;
left:	characters in the structure from left to right;
one:	characters are radicals;
special:	special characters;
learning:	formal radicals learnt by neural networks.

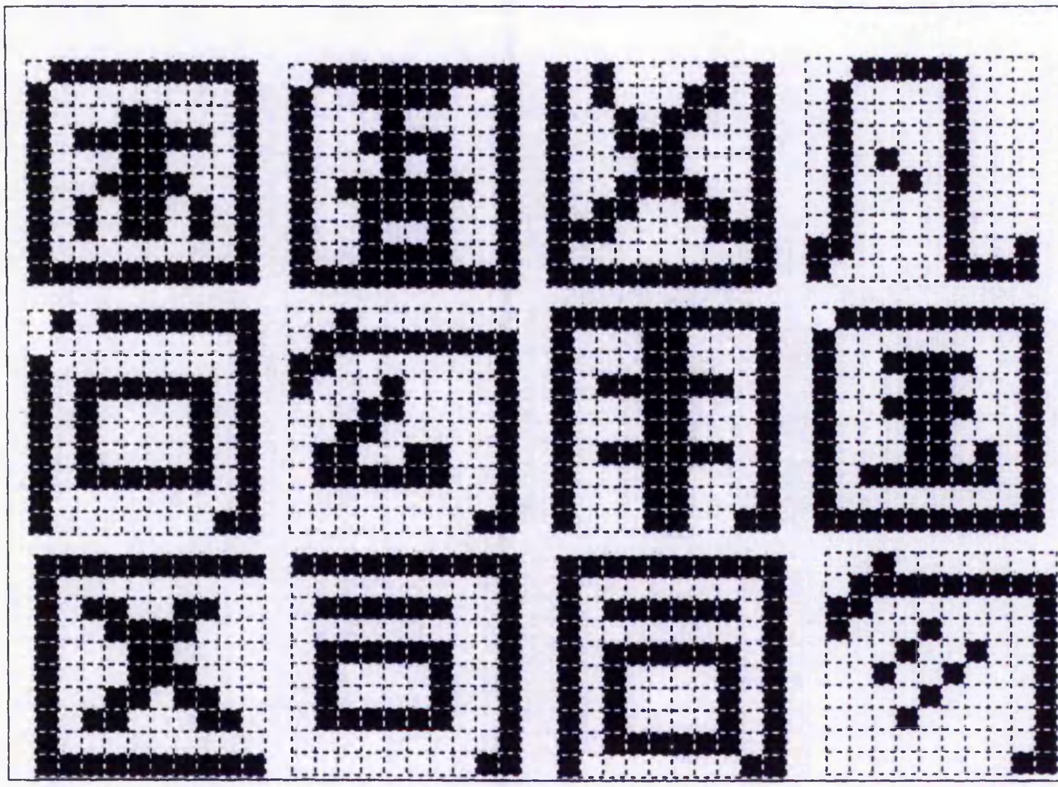


Figure C.1 Test data for outside and inside

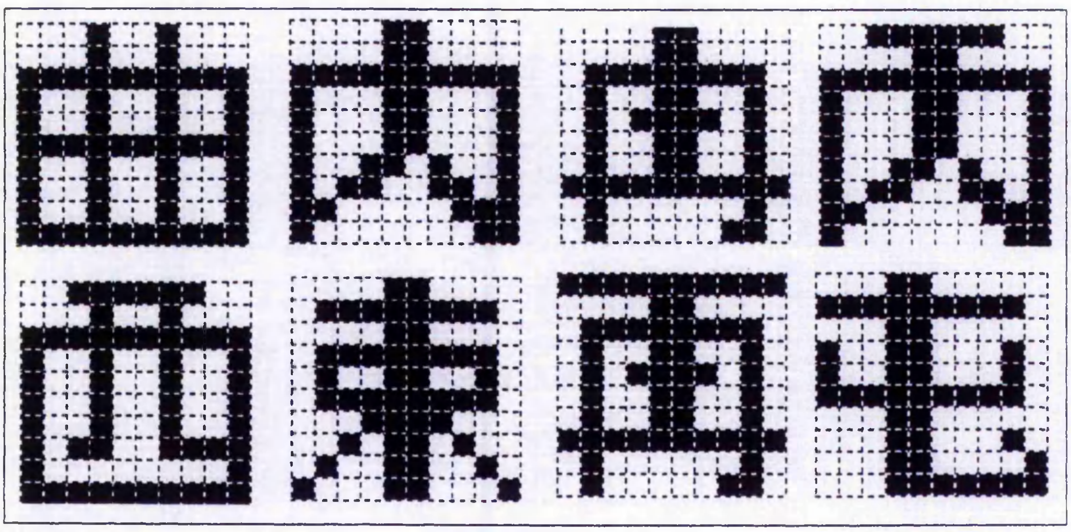


Figure C.2 Test data for inside and outside

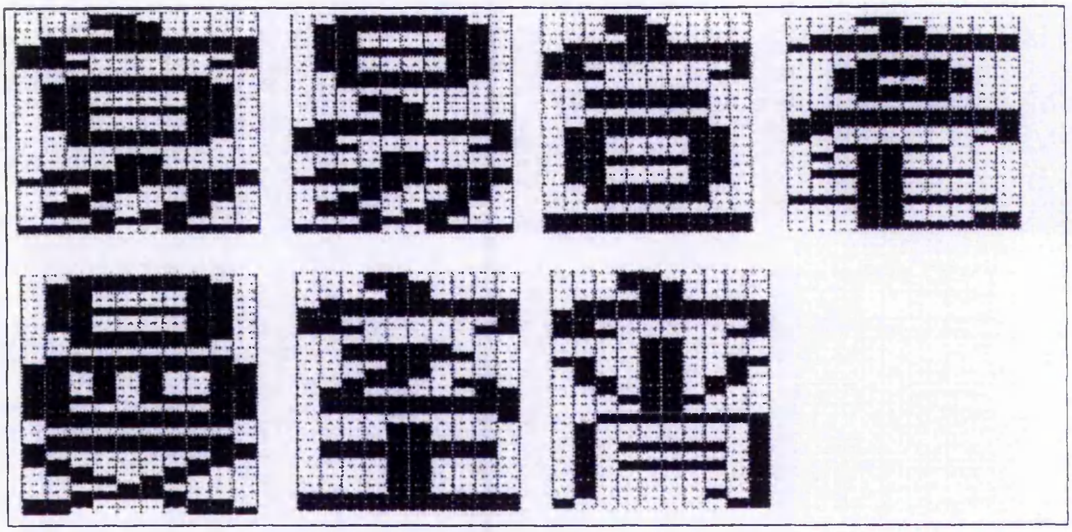


Figure C.3 Test data for top, middle and bottom

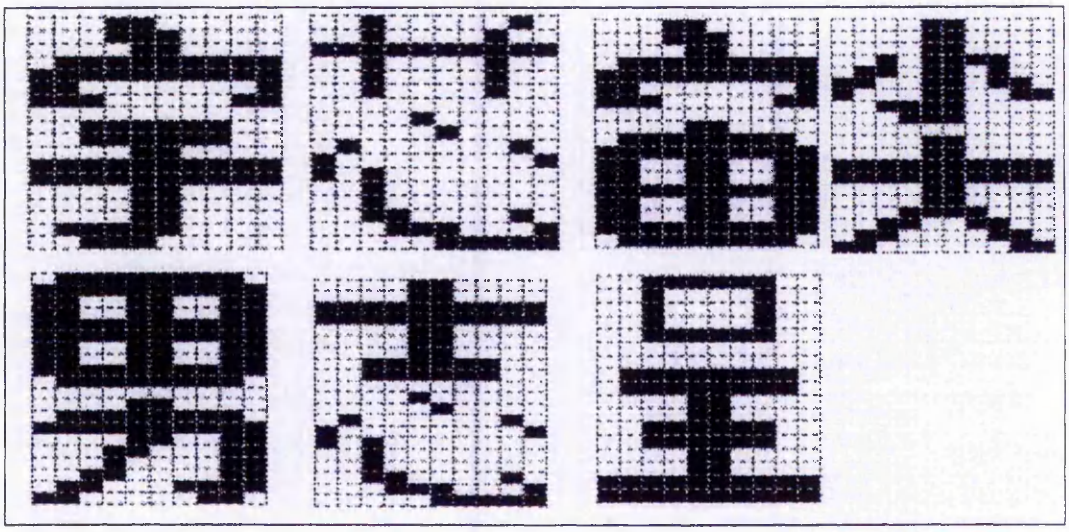


Figure C.4 Test data for top and bottom

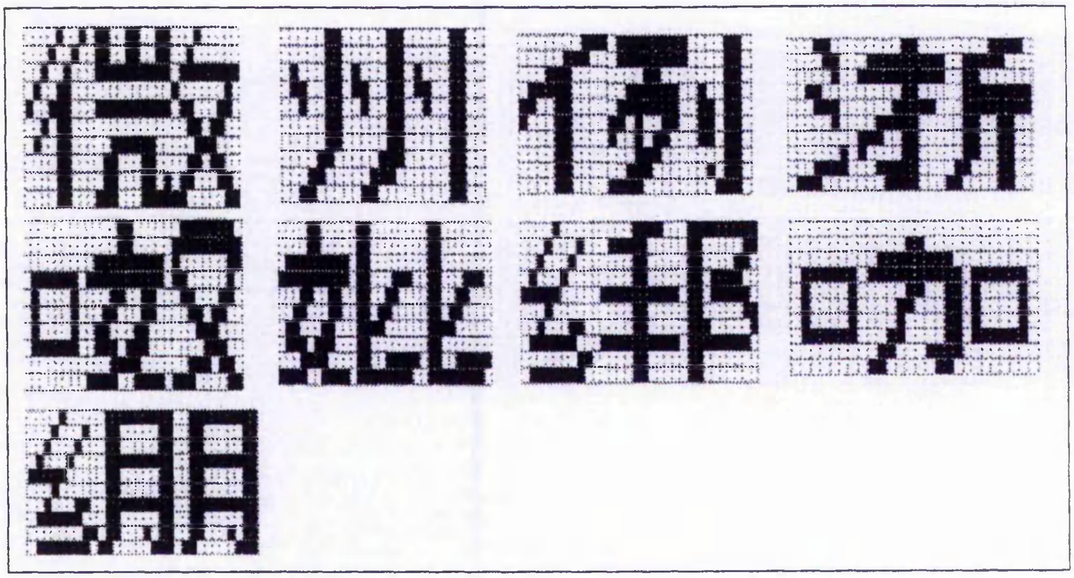


Figure C.5 Test data for left, middle and right

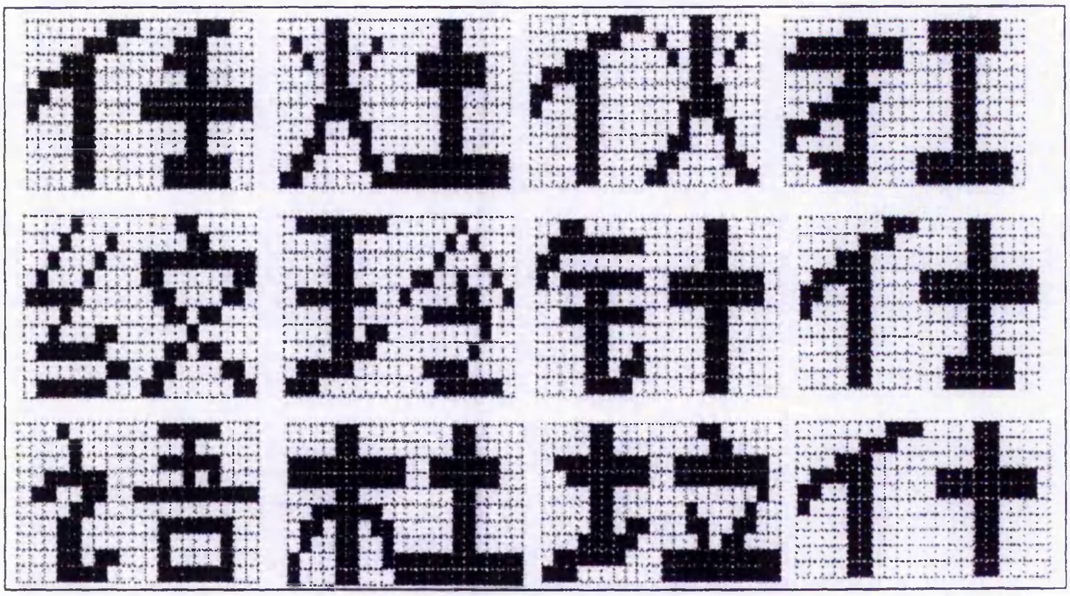


Figure C.6 Test data for left and right

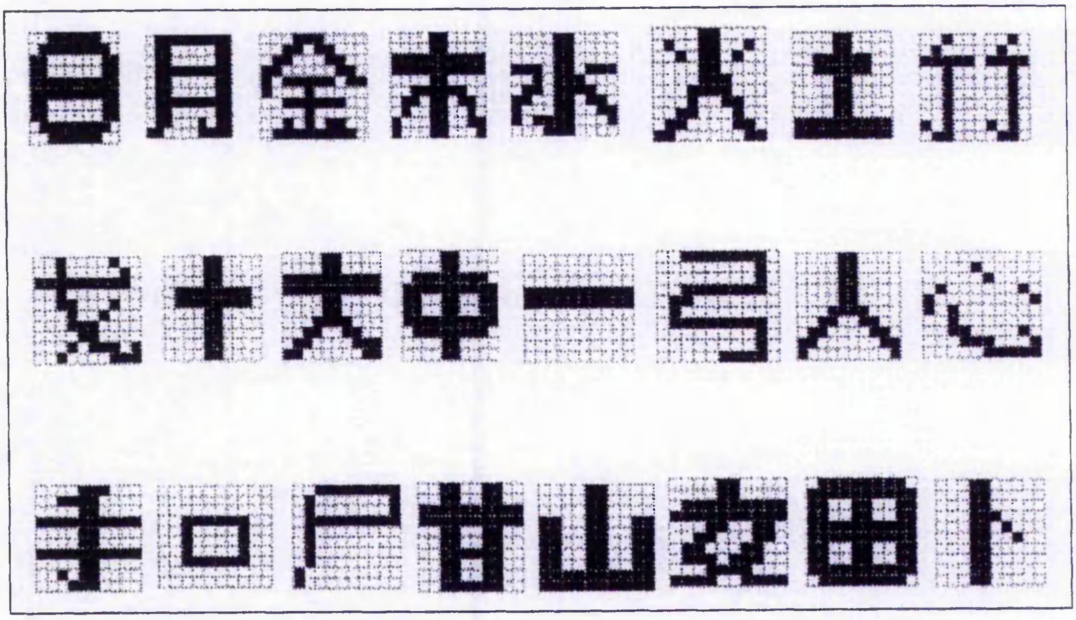


Figure C.7 Test data for radicals

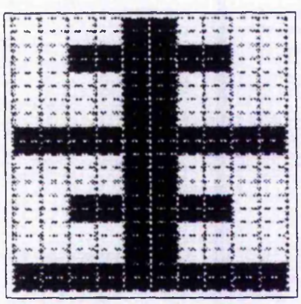


Figure C.8 Test data for special characters

C.2 Outcomes

All experimental outcomes are stored in ~mar/xview/outcomes/%/@@/*.&, where % stands for a process name, for instance, the Preprocessing is called as 'pre'; @@ and * are defined as the same as one in Section C.1; & stands for the property of the file. In general, a result file is named following its original file with a suffix of processing feature and stored at the same subdirectory as its original one.

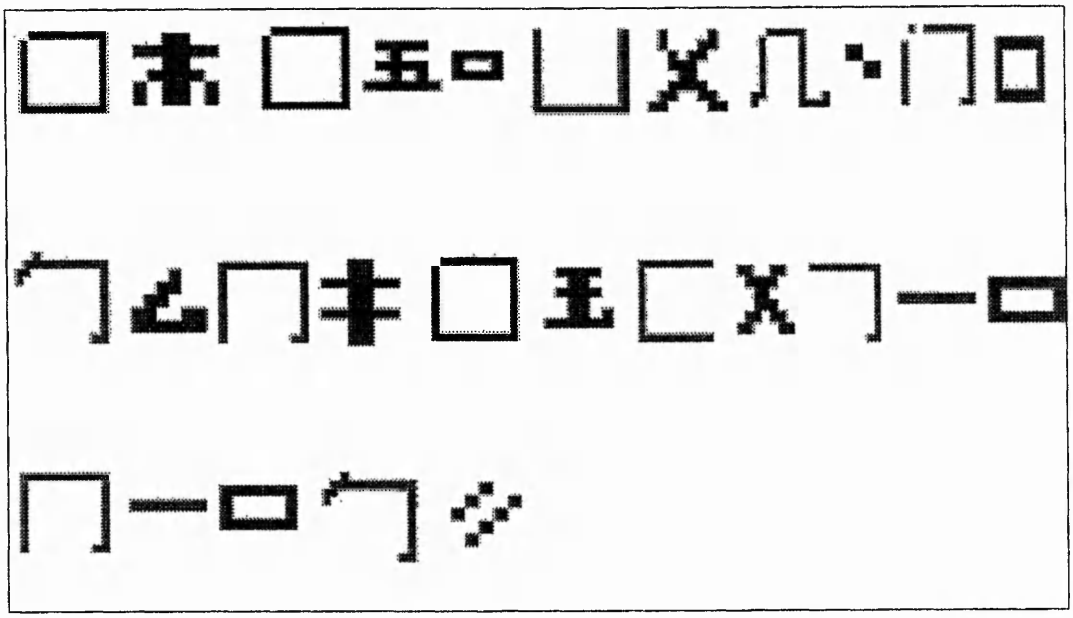


Figure C.9 Outcomes for outside and inside in Preprocessing

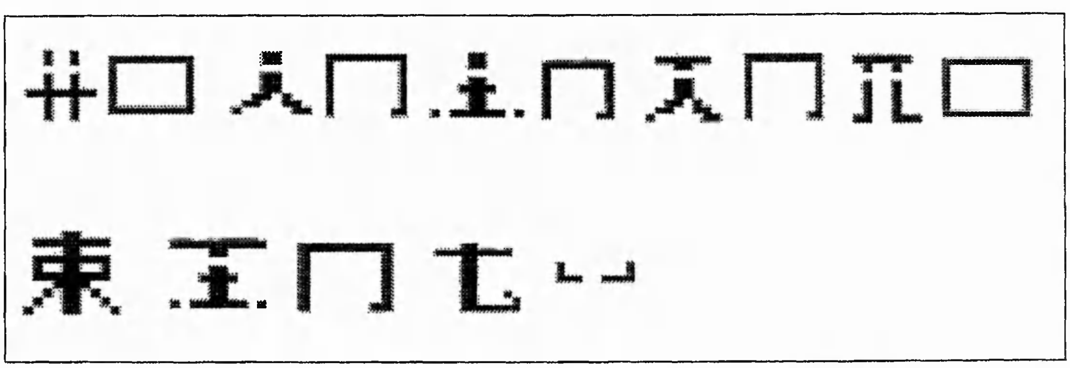


Figure C.10 Outcomes for inside and outside in Preprocessing

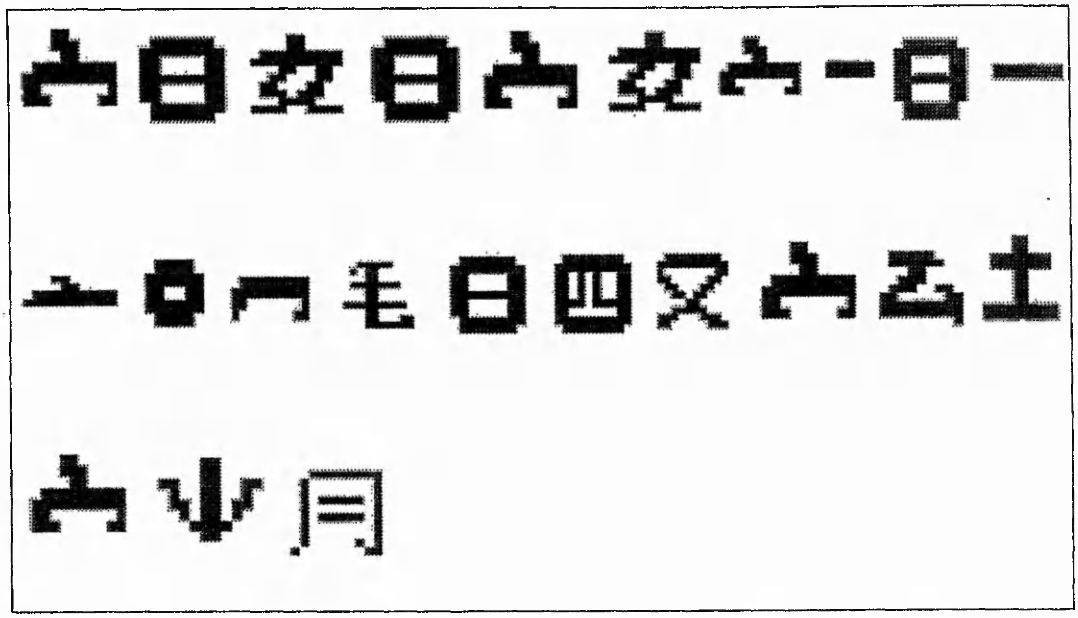


Figure C.11 Outcomes for top, middle and bottom in Preprocessing

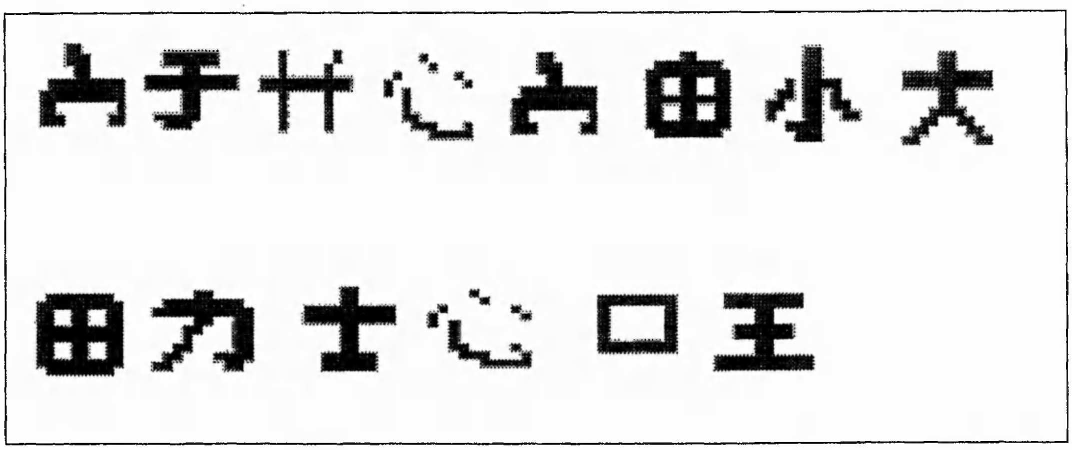


Figure C.12 Outcomes for top and bottom in Preprocessing

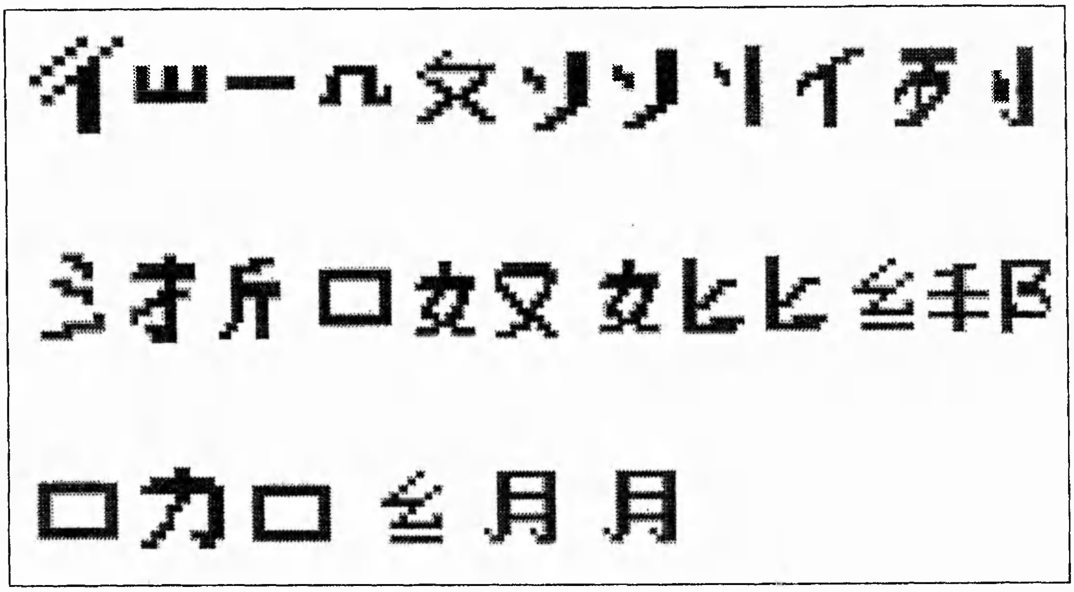


Figure C.13 Outcomes for left, middle and right in Preprocessing

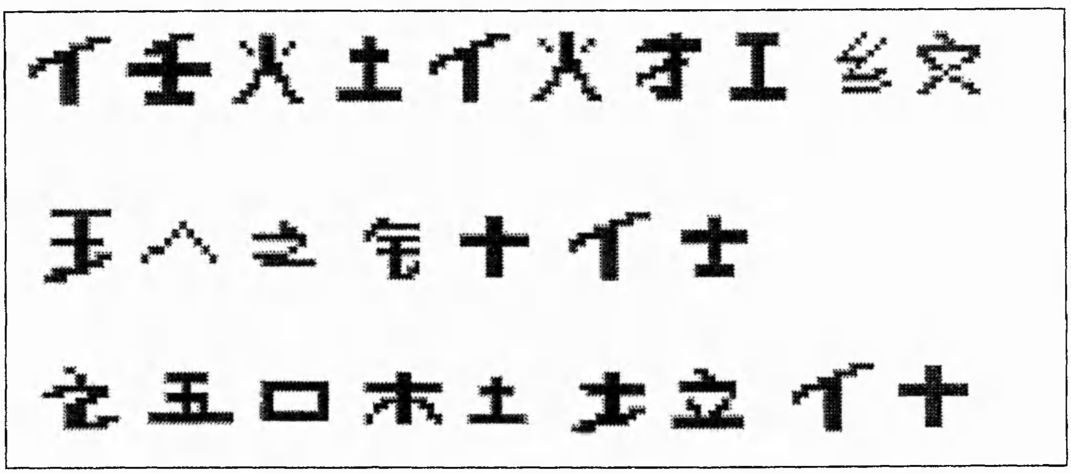


Figure C.14 Outcomes for left and right in Preprocessing

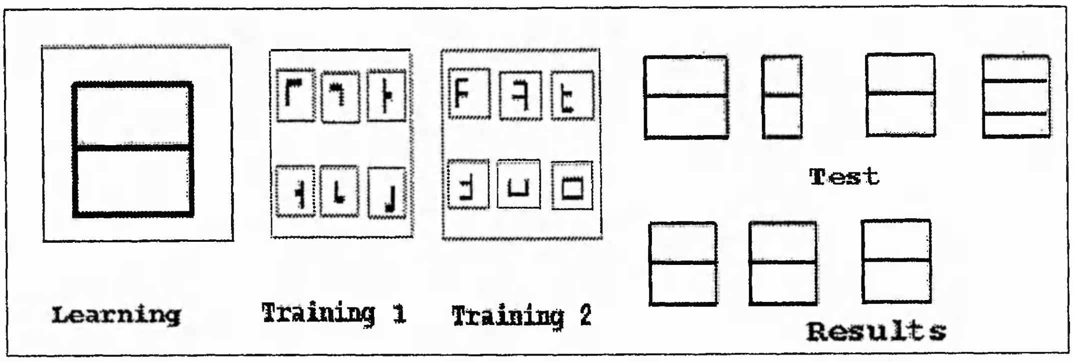


Figure C.15 Outcomes for shape selection in Normalisation

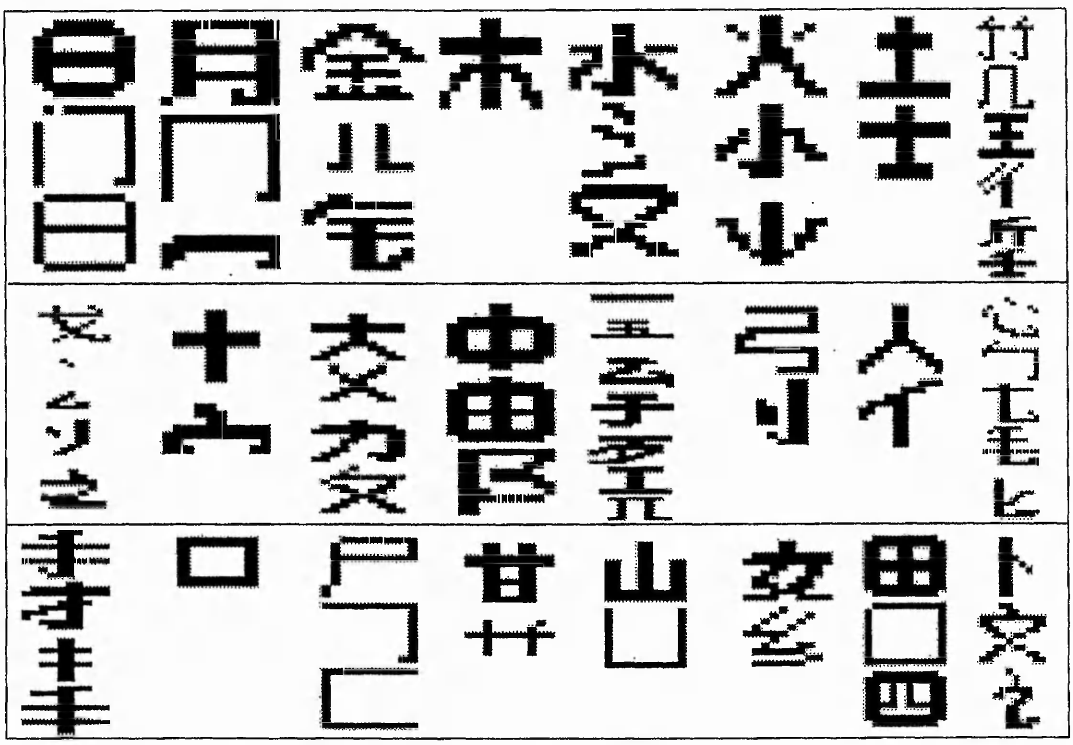
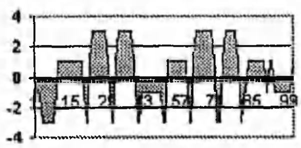
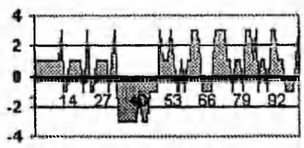


Figure C.16 Learning patterns in Recognition

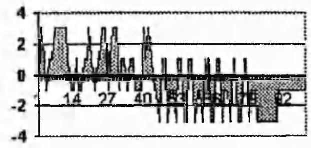
Weights for A sub-net



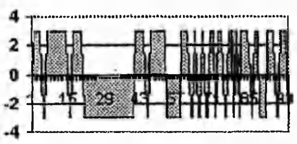
Weighhts for B sub-net



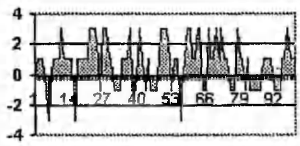
Weights for C sub-net



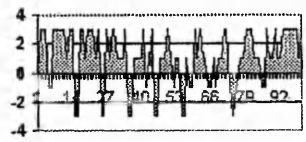
Weights for D sub-net



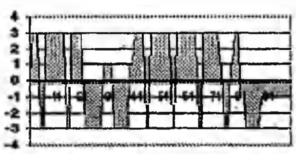
Weighhts for E sub-net



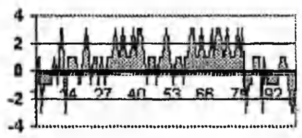
Weights for F sub-net



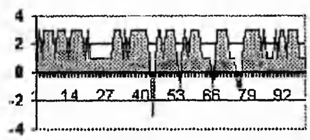
Weights for G sub-net



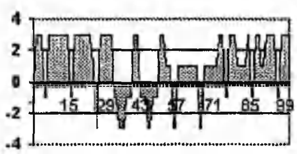
Weighhts for H sub-net



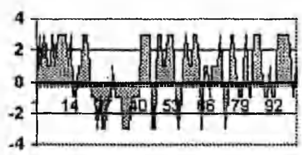
Weights for I sub-net



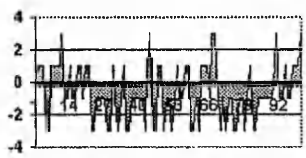
Weights for J sub-net



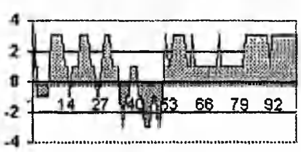
Weights for K sub-net



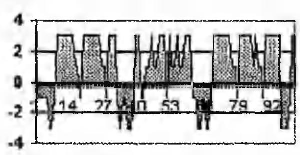
Weights for L sub-net



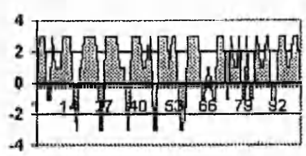
Weighhts for M sub-net



Weights for N sub-net



Weights for O sub-net



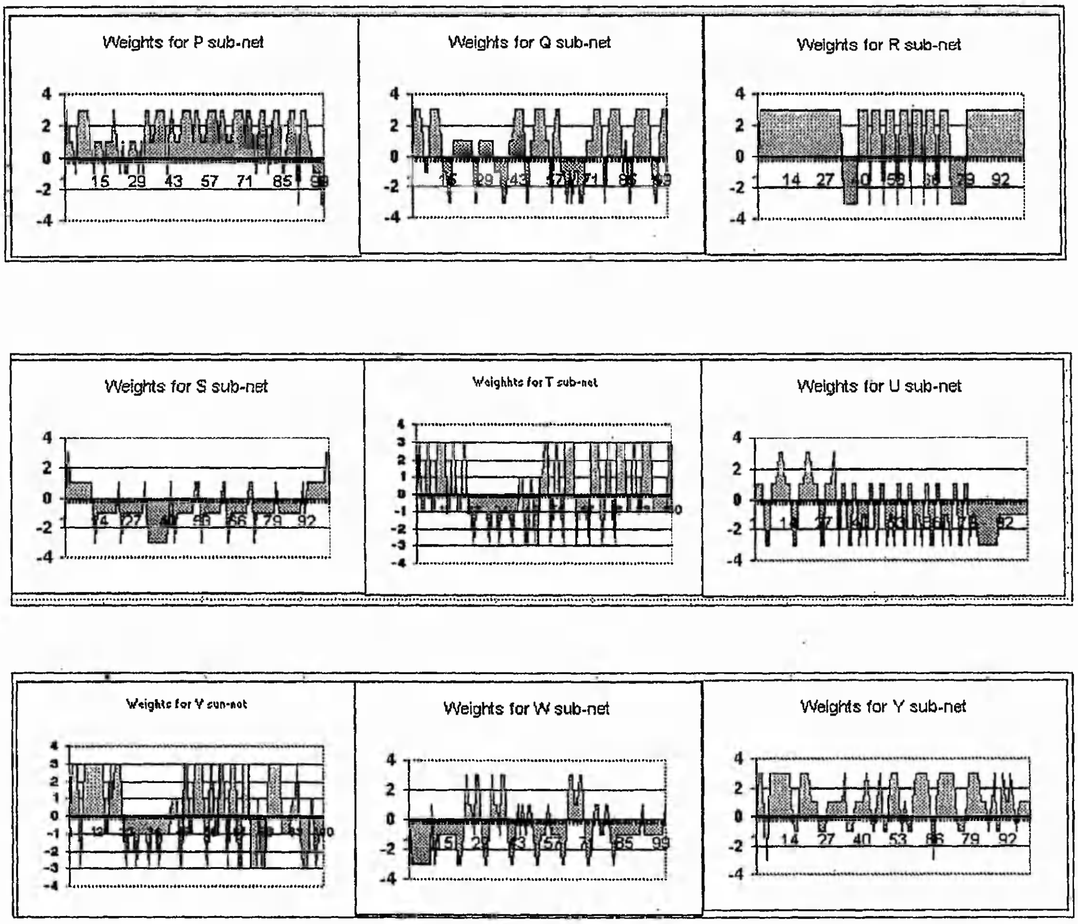


Figure C.17 Weights for sub-nets in Recognition

W D, W M M R, U K, HN I, AN R, P I, B Q,
W HG, S K, S M R, B M R, P I I

T W, O B, B G, MO B, MC W, M G B, P U

J A V, A J V, J M A M, Y R B P, A W L E,
J MI G, J F B

J MD, T P, J LW, F K, W KS, G P, R HG

HO U M H K, IL IL, O MN N, E Q HL, R V E,
V P P, VF QJ L, R KS R, VF B B

O HG, F G, O F, Q M, VF YK, HG O II, C J,
O G, YR MM R, D G, O J

Figure C.18 Outcomes in Recognition

WD, WMMR, UK, HNI, ANR, PI, BQ,
WHG, SK, SMR, BMR, PII

TW, OB, BG, MOB, MCW, MGB, PU

JAV, AJV, JMAM, YRBP, AWLE,
JMIG, JFB

JMD, TP, JLW, FK, WKS, GP, RHG

HOUMK, ILIL, OMNN, EQHL, RVE,
VPP, VFQJL, RKSR, VFBB

OHG, FG, OF, QM, VFYK, HGOII, CJ,
OG, YRMMR, DG, OJ

Figure C.19 Outcomes in Restructuring and Postprocessing

Appendix D. Publications Related from This Research

The work described in this thesis has been published in a number of papers in international journals and referred to conference proceedings. References are provided together with the tag used in the document text. Full copies of the papers are also enclosed.

Journal papers:

[Ren99] Ren, Manling, Al-Dabass, David and Su, Daizhong, *Evaluation of different representation methods for off-line recognition of Chinese characters*, submitted to Knowledge And Information Systems: An International Journal (KAIS), 1999.

[Ren98a] Ren, Manling, Al-Dabass, David and Su, Daizhong, *A three-layer hierarchy algorithm for off-line recognition of Chinese characters*, accepted for The International Journal on Computer Processing of Oriental Languages (CPOL), 1998.

[Ren97b] Ren, Manling, Al-Dabass, David and Su, Daizhong, *A hierarchical approach to fuzzy possibilistic-reasoning for recognising Chinese characters*, International Journal of Computer Mathematics, 1997, Vol.64, No.1&2, pp.17-34, edited by David J. Evans, published by Gordon and Breach Science Publishers, ISSN: 0020-7160.

[Ren96a] Ren, Manling, Al-Dabass, David and Su, Daizhong, *Using a syntactic method to partition a Chinese character*, The Caledonian International Engineering Journal, 1996, 96(1), pp.51-60, edited by David J. Petty, published by Scottish Maintenance Centre Ltd., ISBN: 0 94 8255 83 8.

Conference papers and presentations:

[Ren98b] Ren, Manling, Al-Dabass, David and Su, Daizhong, *Using associative memory* *neural network with sub-nets structure for classification of radicals in Chinese characters*, accepted for ICNN&B'98: 1998 International Conference on Neural Network and Brain, Beijing, China, 27-30 October 1998, sponsored by China Neural Networks Council, co-sponsored by IEEE NNC, IEEE Beijing Section, IEEE NNC Beijing RIG, INNS-SIG, supported by National Natural Science Foundation of China.

[Ren97a] Ren, Manling, Al-Dabass, David and Su, Daizhong, *A fuzzy/possibilistic-reasoning AM-architecture for recognising radicals in Chinese characters*, ISFL'97: Proceedings of Second International ICSC (International Computer Science Conventions) Symposium on Fuzzy Logic and Applications, Zürich, Switzerland, 12-14 February 1997, pp.265-271, edited by N. Steele, published by ICSC Academic Press, Canada/Switzerland, ISBN: 3-906454-03-7.

[Ren96b] Ren, Manling, Al-Dabass, David and Su, Daizhong, *A three-layer hierarchy for representing Chinese characters*, Research and Development in Expert Systems XIII: Proceedings of Expert Systems 96, the Sixteenth Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, UK, 16-18 December 1996, pp.137-146, edited by J. L. Nealon and J. Hunt, sponsored by IEE, dti, EPSRC, ii, published by SGES Publications, ISBN: 1 899621 13 X.

[Ald95] Al-Dabass, David, Ren, Manling and Su, Daizhong, *An associative memory artificial neural networks system with a combined-radicals method for Chinese character recognition*, IEEE-ICONIP'95-Beijing: Proceedings of International Conference on Neural Network Information Processing, Beijing, China, 30 October –3 November 1995, Vol.2 of 2, pp.857-860, edited by Yixin Zhong, Yixian Yang and Minghui Wang, co-sponsored by Asia-Pacific Neural Networks Assembly (APNNA) IEEE Region 10, technical co-sponsored by IEEE Communication Society, International Neural Networks Society, European Neural Networks Society and Russian Neural Networks Society, published by Publishing House of Electronics Industry, ISBN: 7-5053-3355-0/TP.1288.

[Ren95a] Ren, Manling, Su, Daizhong and Al-Dabass, David, *An associative memory artificial neural network system*, ECAC'95-London: Proceedings of 1995 European Chinese Automation Conference, London, UK, 16-17 September 1995, pp.91-96, edited by Guoping Liu and Huaiqiang Zhang, sponsored by The Education Section of Chinese Embassy in London and National Natural Science Foundation of China, organised by The Chinese Automation Society in UK (CASUK).

Reports:

[Ren97c] Ren, Manling, *Algorithms for off-line recognition of Chinese characters*, August 1997, a report for the transfer from MPhil to PhD, The Nottingham Trent University, UK.

[Ren95b] Ren, Manling, *Application of ANN to Chinese character recognition*, September 1995, a report required by the committee of the MPhil/PhD degree registration in The Nottingham Trent University, UK, supervised by Dr David Al-Dabass, Dr Daizhong Su and Dr Lindsay L. Evett.

[Ren94] Ren, Manling, *An associative artificial neural network system with a combined-radical structure for Chinese character recognition*, May 1994, internal report, Department of Computing, The Nottingham Trent University, UK.

A Three-layer Hierarchy Algorithm for Off-line Recognition of Chinese Characters

Manling Ren*, David Al-Dabass* and Daizhong Su**

* Department of Computing
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel.: +44 115 941 8418 Ext. 2761
Fax: +44 115 948 6518
E-mail: mar@doc.ntu.ac.uk
dad@doc.ntu.ac.uk

** Department of Mechanical and
Manufacturing Engineering
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel.: +44 115 941 8418 Ext. 2306
Fax: +44 115 948 6506
E-mail: daizhong.su@ntu.ac.uk

ABSTRACT

This paper presents an algorithm of using the three-layer hierarchy method: character-radical-stroke, and its process: character-radical–chain code for off-line recognition of Chinese characters. Based on the algorithm, a system has applied fuzzy possibilistic rules to extract radicals from a character, an associative memory neural network to classify and recognise components: radicals, of the characters, and the chain code method to restructure a chain code for verification. Several special cases and application limitations of the algorithm are discussed. Test results show that the algorithm is effective and reasonable.

Key words: Associative-memory neural network, chain code, Chinese character recognition, fuzzy possibilistic rules, three-layer hierarchy, topological structure.

1. INTRODUCTION

Computer recognition of Chinese character is a challenging topic and important research area. It is relevant to documentation, publications, language translation, handwriting of Chinese and Japanese '*kanji*' in industry, business, diplomacy and daily life. Typical development of the recognition process focuses on printed, on-line and off-line handwriting characters using techniques such as a two-layer hierarchy, four-corner, radical and a whole character recognition [1]. Although their use has achieved some success, the lack of fundamental algorithms for representing the structure of Chinese characters has prevented the application of these methods to recognising characters with large vocabulary and complicated topological structure embedded within the 2-D pictorial format.

In order to remedy the lack of an effective recognition process of such characters, a system called Chinese Character Recognition System (CCRS) with an algorithm of applying the three-layer hierarchy method: character-radical-stroke [2] and the three-layer hierarchy process: character-radical-chain code is developed for off-line recognition of Chinese characters. The algorithm is created on investigation of methods of dealing with variable size, position, shape, vagueness and ambiguity of a character. The three-layer hierarchy method has significantly represented complex topological structure of Chinese characters, and the three-layer hierarchy process has been developed as an effective tool for classification and verification of Chinese characters.

The structure of the CCRS system is illustrated in Figure 1. In the structure, a character is input in its off-line format, i.e. a 2-D image format, and written in a formal printed or the '*kaishu*' style. In the Extraction processing, a character is deconstructed into radicals by fuzzy possibilistic rules in terms of their position, extracting order and shapes. The Recognition processing carries out classification and recognition of the radicals by means of an associative memory neural network. The radicals recognised are translated into codes instead and restructured them into a chain code for verification by using a code database in the Restructuring processing.

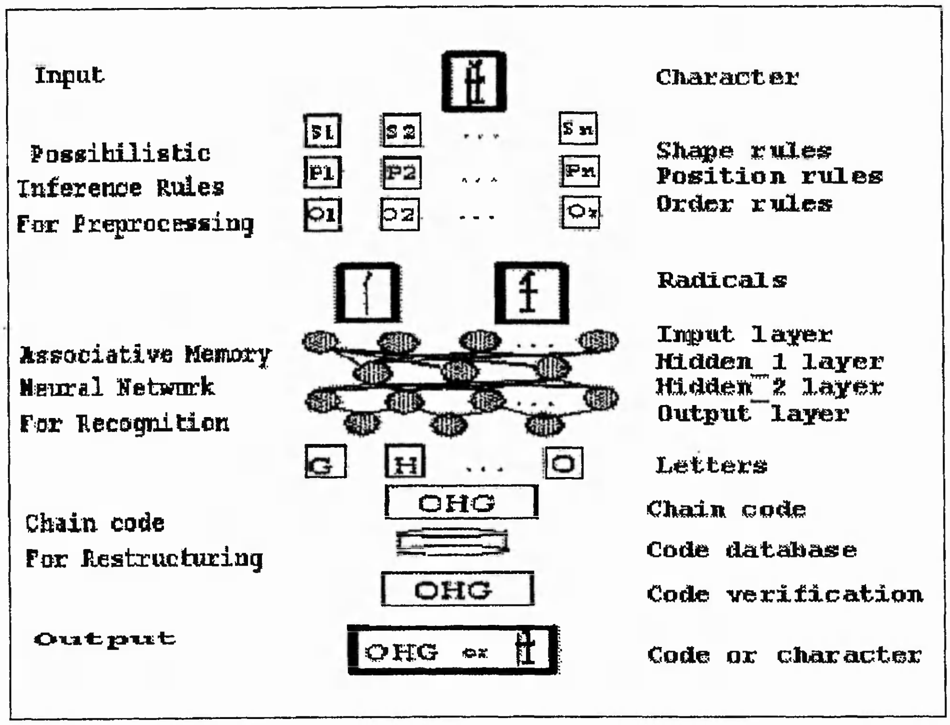


Figure 1. The structure of Chinese character Recognition System (CCRS)

2. FEATURES OF CHINESE CHARACTERS AND THREE-LAYER HIERARCHY METHOD

2.1 Features of Chinese Characters

Chinese characters possess three major features in their structures and quantities: a two-dimension (2-D) pictorial format, topological structure and large vocabulary.

In the 2-D pictorial format, basic components: strokes can be situated at any position of a character. Figure 2 (a) shows that the stroke '*horizontal line*' can be located at several places in a character. In Figure 2 (b), the stroke '*horizontal line*' may change its identity once its direction is altered. Figure 2 (c) displays that the stroke '*horizontal line*' has three different lengths in a character.

The topological structure of a character means that the character is combined with or deconstructed into several components as shown in Figure 3 (a). In reverse, a same component may appear in different characters as illustrated in Figure 3 (b). Components can be located at different positions in a character as shown in Figure 3 (c).

The vocabulary of Chinese characters is defined as 3,500 characters for daily use, 7,000 characters necessary in writing, and 60,000 characters in total. Based on the feature of a 2-D picture, each Chinese character may be seen as a pattern different from others. Therefore, an adequate representation of a character requires a matrix of pixels about 10 times the number needed for a Roman letter [3].

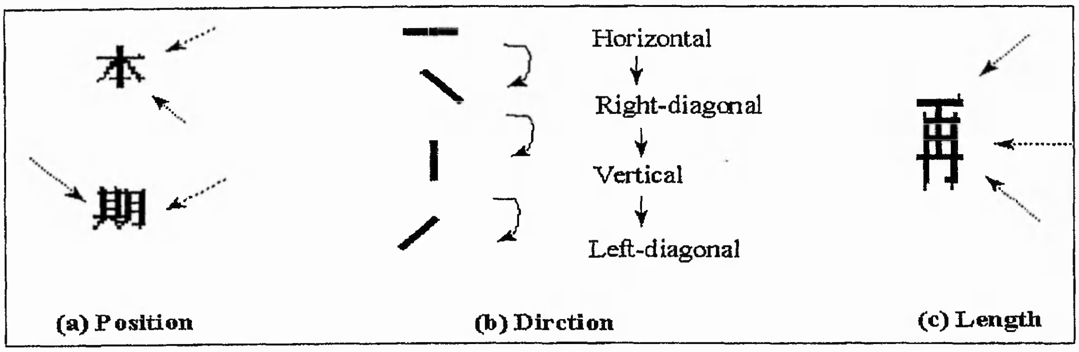


Figure 2. 2-D pictorial format of Chinese characters

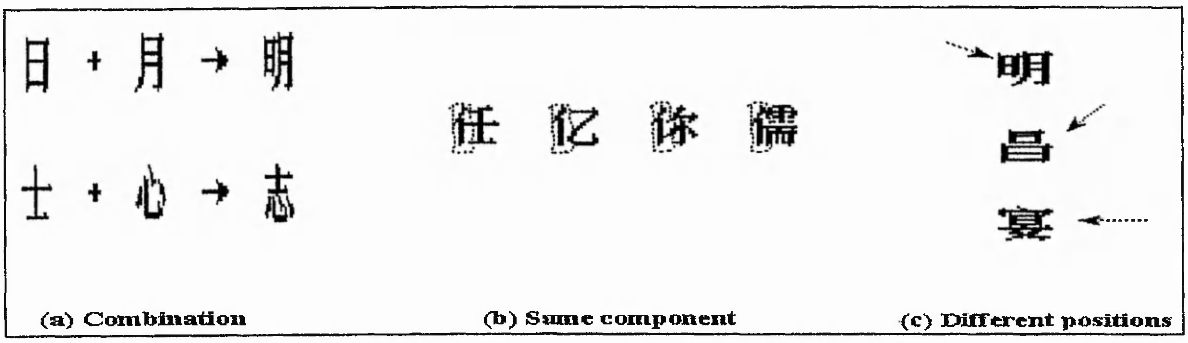


Figure 3. Topological structure of Chinese characters

2.2 Three-layer hierarchy

In the three-layer hierarchy method, the structure of Chinese characters is represented in three layers: character, radical and stroke [1]. Basic strokes in the bottom layer are treated as indexes to determine the shape of radicals in a character. Radicals in the second layer are used to deconstruct the internal topological structure of a character in order to reduce the amount of characters learnt by computer. Characters in the top layer are recognised by restructuring radicals into a chain code and verifying it by means of a code database. Based on this method, the processing of recognising a character is carried out in sequence of character, radical and chain code. Figure 4 gives an example for illustrating the method.

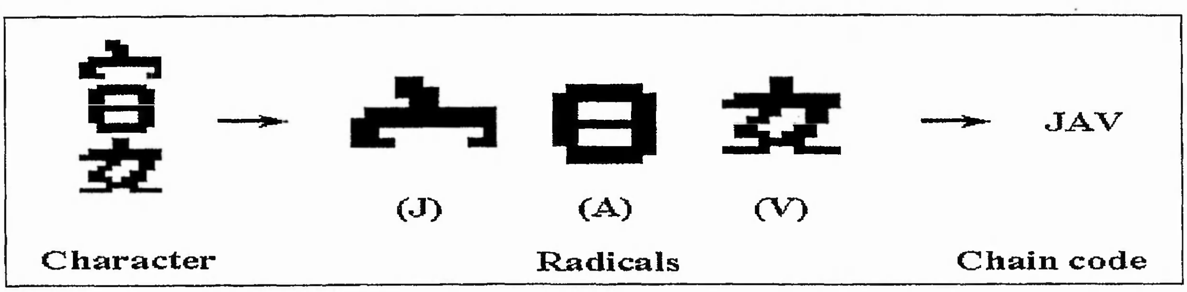


Figure 4. An example proceeded by the three-layer hierarchy [1]

The three-layer hierarchy offers the advantages of (a) processing Chinese characters with a similar structure; (b) describing the internal topological structure of a character in a more systematic representation; (c) reducing the vocabulary of the characters learnt by computer; and (d) using a chain code instead of a character to simplify its recognition process. However, relations between radicals and strokes in the method are still very complicated to be represented and simplified [1].

3. REPRESENTATION OF TOPOLOGICAL STRUCTURE FOR EXTRACTING A RADICAL

Based on the basic features of Chinese characters: 2-D and topological structure, difficulties on representing the structure of the characters centre on (a) how to deal with the vagueness problem of shape, position and order of radicals in a character, and (b) compressed relations between a character and radicals, and between a radical and strokes.

3.1 Fuzzy Possibilistic Reasoning

Fuzzy possibilistic reasoning in knowledge representation approaches is well suited to dealing with imperfect, uncertain and vague information [4]. Reducing the complexity of imperfect information is achieved by information-compressed representations based on if-then rules. These rules are interpreted as logical implications and are termed as possibilistic inference rules defined by the notation $\mathfrak{R}$.

Based on the approximate reasoning and probability theory, fuzzy possibilistic representation uses conjunctly combined rules to validate a possible resolution from various restrictions. In these if-then rules, antecedent in the *IF* clause and consequence in the *THEN* clause are constrained by their possibility distributions denoted by π. The possibility distributions are related with the interpretation of vague concepts as contour functions of random sets. Physical quantities of the distributions are defined by the possibility measures denoted by Poss_π.

Generally, a possibilistic inference rule $\mathfrak{R}_j$ can be expressed by

$$\mathfrak{R}_j: \textbf{ IF } \xi^{S}_{j} \text{ is } \mu_j \textbf{ THEN } \xi^{T}_{j} \text{ is } v_j, \qquad j = 1, \ldots, r,$$

or

$$\mathfrak{R}_j: \textbf{ IF } \xi^{S}_{j}{}^{(1)} \text{ is } \mu_j{}^{(1)} \textbf{ AND } \xi^{S}_{j}{}^{(2)} \text{ is } \mu_j{}^{(2)} \textbf{ THEN } \xi^{T}_{j} \text{ is } v_j, \qquad j = 1, \ldots, r,$$

where μ_j, $\mu_j^{(1)}$, $\mu_j^{(2)}$ and v_j are subsets of possibility distributions on the space sets S_j and T_j with regard to j respectively. ξ is a variable whose values can be arbitrary possibility distributions on S_j or T_j. The symbol is, appearing in possibilistic inference rules, serves as a linguistic description of the operator $\subseteq$ and is therefore to be seen as 'is at least as specific as'.

The relation $\mathfrak{R}$ of all rules is

$$\mathfrak{R} = \bigcap_{j=1}^{r} \mathfrak{R}_j$$

3.2 Situation Representation

The situation representation method uses inference rules defined by the above interpretation for determining radicals in a character. The representation focuses descriptions on (a) the position of a radical in a character, and (b) the order of extracting a radical from a character.

Let the notation P and O stand for two domains of the position of radicals in a character and the order of extracting a radical from a character respectively, their possibility distributions could be defined as $\pi(P)$ and $\pi(O)$ according to the definition of fuzzy possibilistic reasoning. The possibility measures are given by the notations $\mathrm{Poss}_\pi(P)$ for $\pi(P)$, and $\mathrm{Poss}_\pi(O)$ for $\pi(O)$. The notation $\mathfrak{R}^{(PO)}$ represents the possibilistic inference rules. If p and o denote variables with the domains P and O respectively, the $Poss_\pi$ (p) is the possibility measure of p on $\pi(P)$; similarly, $Poss_\pi$ (o) for o on $\pi(O)$.

Position variance: The investigation of position variance of radicals in a character is based on their features of a two-dimensional picture and a rectangular appearance, one of the major characteristics in the structure of Chinese characters. The domain of position variable is defined by

$$P = \{\textit{width, length}\}.$$

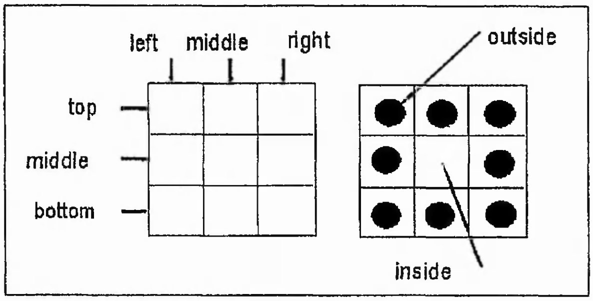
Because a radical may keep an independent position in a character, the possibility distribution of position variance of a radical on the domain P, shown in Figure 5, is defined by

$$\pi(P) = \{\textit{outside, inside, top, bottom, left, right, middle}\}.$$

$Poss_\pi$ (P) for $\pi(P)$ is defined by, for instance,

$Poss_\pi$ (*left*) = {*width* $\leq$ *2/3 width of P, length = length of P*}.

Extraction order: The extraction order indicates the sequence of radicals extracted from a character that might consist of two or more radicals. The domain of extraction order is expressed by



$$O = \{\textit{first, last}\}.$$

Figure 5. Possible position of a radical

The possibility distribution of extraction order on the domain O is represented by

$\pi(O)$ = {*outside* $\rightarrow$ *inside, inside* $\rightarrow$ *outside, top* $\rightarrow$ *(middle* $\rightarrow$ *bottom), top* $\rightarrow$ *bottom, left* $\rightarrow$ *(middle* $\rightarrow$ *right), left* $\rightarrow$ *right*}.

The notation '$\rightarrow$' stands for the sequence from the first to the latter. Distinction of some distribution representations, such as, 'outside $\rightarrow$ inside' and 'inside $\rightarrow$ outside', will depend on inference rules between order, position and shape mentioned in the next section.

With developing such basic possibility distribution of order $\pi(O)$ above, a complex distribution could be derived, for instance,

$\pi(O)^{(1)}$ (*top* $\rightarrow$ *bottom (left* $\rightarrow$ *right)*) = {*top* $\rightarrow$ *bottom left* $\rightarrow$ *bottom right*}.

$Poss_\pi$ (O) for $\pi(O)$ is defined by, for instance,

$Poss_\pi$ (*top* $\rightarrow$ *bottom*) = {*there are two rectangles*}.

Now, possibilistic inference rules $\mathfrak{R}^{(PO)}$ might be established for representing relations between the position and order of a radical. As examples, several rules are shown as follows.

$\mathfrak{R}^{(PO)}_{(1)}$: **IF** position is top **THEN** order is first,
$\mathfrak{R}^{(PO)}_{(2)}$: **IF** position is bottom **THEN** order is last.

3.3 Shape Representation

The shape representation method centres on the shape domain of radicals, its possibility distributions and measure technique. Inference rules are established for the representation of radicals' relationships between their shape, position and order.

Possibility distributions and measures: The domain of radicals is defined as a rectangle in different sizes in terms of features of combined strokes. The shape domain of radicals is expressed by

$$S = \{\textit{rectangle}\}.$$

Different valid combinations of basic strokes assign the possibility distributions on the domain S, where the validity of the combinations is checked. The modes of combinations are classified as connection and disconnection. The possibility distributions are represented by

$$\pi(S) = \{\textit{combination of basic strokes, basic strokes}\}.$$

In order to determine the shape of a radical, possibility measures are based on evaluation of a continuous line, direction of a line connecting with other lines, priority of such direction and disconnecting distance. For instance, one of the possibility measures $Poss_\pi$ (S) for $\pi(S)$ is defined as follows:

$$Poss_\pi(\textit{priority of up} \rightarrow \textit{down}) = \{\textit{up} \rightarrow \textit{down, up} \rightarrow \textit{down left, right} \rightarrow \textit{left, up} \rightarrow \textit{down right}\}.$$

Shape vagueness and possibilistic inference rules: To produce general concepts of forming a radical, the shape vagueness of radicals is investigated for expressing the relation of combining two strokes. The relations can be classified as angle, location, continuous, distance and discontinuous.

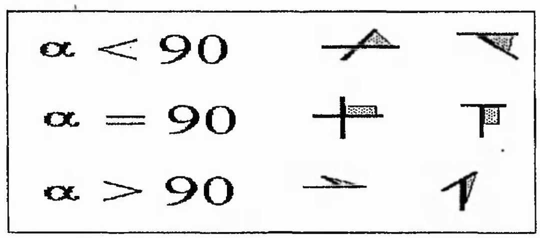


Figure 6. Angle of stroke connected

The angle relation indicates a contour expression of two connected strokes. For example, it is defined as a contour if two connected strokes form an angle. Figure 6 shows three different types of angles from two connected strokes.

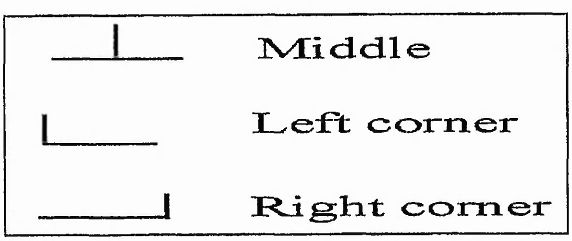


Figure 7. Location of strokes connected

The location relation stands for a place of two connected strokes. Figure 7 gives several examples to show the location relation.

The continuous relation expresses a possibility of a contour as part of a radical. A continuous contour is defined if a contour is formed with an angle.

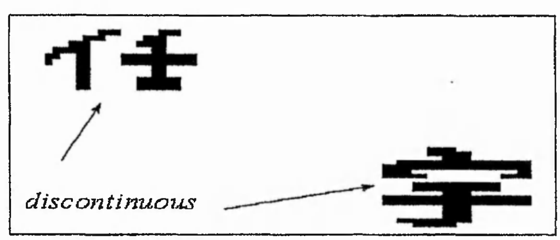


Figure 8. Discontinuous contour

The distance relation is to measure a scope of two disconnected strokes.

The discontinuous relation implies the possibility of a contour that may form two radicals in their shape. A discontinuous contour is decided by distance measure of two disconnected strokes. Figure 8 gives several examples for showing the discontinuous contour.

Possibilistic inference rules are established by representations of relations between shape vagueness denoted by $\Re^{(S)}$, between shape and position by $\Re^{(SP)}$, and between shape, position and order by $\Re^{(SPO)}$. For example, the inference rules shown below are defined to divide a character into two parts: c1 and c2 from the inside to outside.

$\Re^{(S)}{}_{(1)}$: **IF** contour of c1 **is** square **AND** c2 **is** continuous contour of c1
AND angle of c1 connecting with c2 **is** 90
AND location of c2 **is** on the top middle of c1
THEN shape **is** combination of c1 and c2 (c1+c2).
$\Re^{(SP)}{}_{(2)}$: **IF** shape **is** c1+c2 **THEN** c1 position **is** outside.
$\Re^{(SP)}{}_{(3)}$: **IF** shape **is** c1+c2 **THEN** c2 position **is** inside.
$\Re^{(SPO)}{}_{(4)}$: **IF** shape **is** c1+c2 **AND** position **is** outside **THEN** order **is** last.
$\Re^{(SPO)}{}_{(5)}$: **IF** shape **is** c1+c2 **AND** position **is** inside **THEN** order **is** first.

4. CLASSIFICATION AND RECOGNITION OF RADICALS BY USING AN ASSOCIATIVE MEMORY NEURAL NETWORK

The classification and recognition of radicals extracted from a character are carried out by a four-layer neural network, which is composed of multi sub-nets and based on architecture of associative memory function [1]. Sub-nets are developed to reduce intra connectivity of the network and to deal with radicals in a category. The associative memory function offers an important advantage of recalling a stored pattern from its partial or noisy input [5].

4.1 Classification of Radicals

The classification of radicals is developed through the application of the *Cang-Jie* method [6], which classifies radicals according to their shape rather than meanings; However, it is manual in operation and depends on human observation.

In the *Cang-Jie* method, 30,000 Chinese characters are classified into 4 groups that include 26 different categories (24 for radicals and 2 for special cases). In the 24 radical categories, each category includes a standard radical and some other radicals and token-radicals that are similar to or simplification of the standard radical, in terms of either their shape or meanings. The total number of radicals and token-radicals in these categories is 108. The 2 special case categories are designed for difficult characters and for making a new character. All categories are numbered by following the alphabetic letters, A, B, C, ... Z, where X and Z stand for difficult and new characters in the 2 special case categories, respectively.

Since radicals have been determined as major objects for recognition, the policy of classifying radicals has to be considered carefully. Three basic principles of determining categories are developed as: (a) a member in a category should have the physical properties of the category and major features from the group that the category belongs to; (b) each member in a category may be a radical or a token-radical or some combination of basic strokes; and (c) combinations of a token-radical and basic strokes are allowed to form a new integrated radical. The policy has some benefits in transforming knowledge of a radical identified abstractly by human analysis into its shape recognised by computer.

4.2 Algorithms of the Associative Memory Neural Network

Neural networks are developed on the basis of mathematical foundation, inherent parallelism knowledge store, fault tolerance, and adaptability [7]. Some mathematical equations affect inputs, memory, recall, determination of energy levels, convergence, and stability. The mathematical formula for the associative memory function is established on the construction of an energy equation [8]:

$$E=-\sum_i\sum_j X_j W_{ij} Y_j+\sum_i \theta_i X_i+\sum_j \varphi_j Y_j \qquad \ldots\ldots \qquad [4.1]$$

The algorithm contains two phases: learning and training. In the learning phase, the associative memory function is used to form the connectivity matrix W for training a set of input patterns $X_i(u)$ and output patterns $Y_i(u)$, where u =1, 2, ... M, i = 1, 2, ... N, the weight W(i, j) is determined by the Hebbian rules:

$$W(i,j)=\begin{cases}\sum_u X_i(u)Y_j(u) & if\quad i\neq j\\ 0 & if\quad i=j\end{cases} \qquad \ldots\ldots \qquad [4.2]$$

Two special assumptions are made:

$$\begin{aligned}&\theta_i=\varphi_j=0\\ &or\\ &\theta_i=\varphi_j=\frac{1}{2}\sum W_{ij}\end{aligned} \qquad \ldots\ldots \qquad [4.3]$$

In the training phase, the algorithm aids convergence because its value in equation [4.1] either is reduced or remains constant during the recall procedure [9], if the following conditions are satisfied:

$$\begin{aligned}Y_j^{n+1}&=\begin{cases}1 & \sum_i W_{ij}X_i^n-\varphi_j>0\\ Y_j^n & \sum_i W_{ij}X_i^n-\varphi_j=0\\ -1 & \sum_i W_{ij}X_i^n-\varphi_j<0\end{cases}\\ X_i^{n+1}&=\begin{cases}1 & \sum_j W_{ij}Y_j^n-\theta_i>0\\ X_i^n & \sum_j W_{ij}Y_j^n-\theta_i=0\\ -1 & \sum_j W_{ij}Y_j^n-\theta_i<0\end{cases}\end{aligned} \qquad \ldots\ldots \qquad [4.4]$$

When a portion of the original pattern is used as a retrieval cue, the algorithm is denoted as auto-associative memory. When the desired output is different from the input, the algorithm is called hetero-associative. The static method requires that the network is trained in advance with specific patterns, and retrieves them from portions or distorted inputs. When the learning and retrieval are embedded in the training process randomly, it is defined as a 'dynamic' method.

4.3 Architecture of the Associative Memory Neural Network

The associative memory neural network in the system is consisted of four layers: input, hidden-1, hidden-2 and output. Hidden-1 layer consists of multi sub-nets where each sub-net deals with radicals in a category. The number of neurones in each sub-net is decided by the learning patterns in the category. The connectivity from the input to the hidden-1 layer is static. Neurones in the hidde-2 layer are created by the results recognised from the hidden-1 layer. The connectivity between the two hidden layers is dynamic. The design of hidden-2 layer with a dynamic structure is used to further enhance convergence on global minimum of the associative algorithms. Figure 9 shows the architecture of the network.

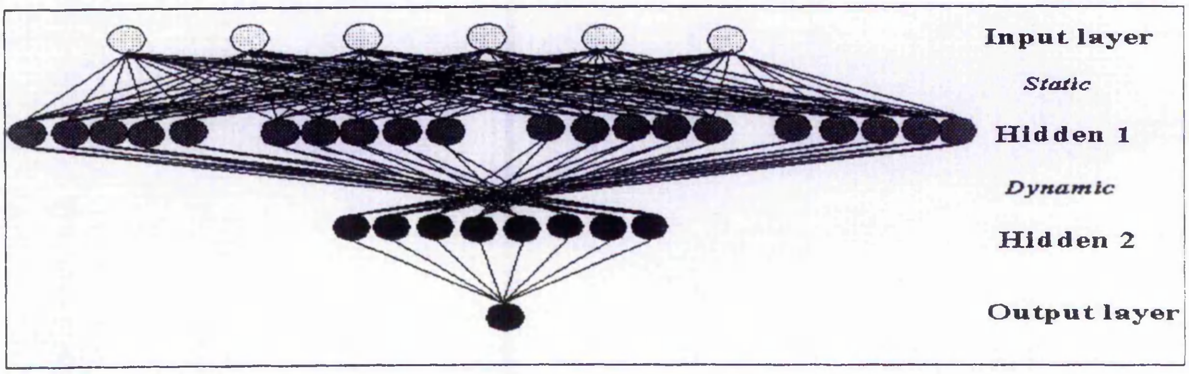


Figure 9. Architecture of the associative memory neural network

In the learning phase, radicals are classified into categories; each is forming a sub-network to reduce the connectivity of the whole network and to use shared weights. The 26 different sub-networks are composed of a whole neural network with associative memory function. The major task in the learning phase is to learn formal radicals and to form inter connectivity for training an input pattern. Although input patterns extracted from a character are different from the formal learning patterns, they could be assumed as approximate to these formal ones after normalisation, i.e. a processing between extraction of a character and recognition of radicals.

In the training phase, sub-nets in the hidden-1 layer are trained to converge to local minima. The hidden-2 layer is generalised by re-learning these patterns of local minima. Eventually, the global minimum will be converged to in the output layer.

When the network is connected as a whole, its inter connectivity is low but intra connectivity is high [10]. The intra connectivity could be reduced while the architecture of sub-nets is used.

5. USING THE CHAIN CODE METHOD TO RESTRUCTURE A CHARACTER FOR VERIFICATION

The chain code method is to translate recognised radicals into letters and to restructure them into a chain code instead of a character for verification at the character level. Advantages of using a chain code substitute are (a) restructuring several radicals to a character without considering their shape, size and position in a 2-D picture; and (b) confirming a character in a database using its chain code, i.e. a combination of several letters standing for a character, rather than a character in a bitmap or other different fonts.

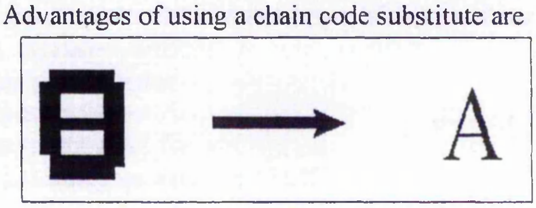


Figure 10. Format of radicals

While a radical is encoded from a pictorial format to an English letter, as shown in Figure 10, the letter only stands for a category label to which the radical belongs.

Restructuring cues are some specified rules for successful formation of a china code. Some of these basic rules are listed as follows.

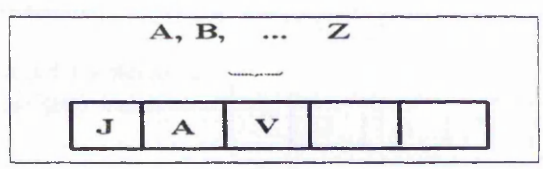


Figure 11. Format of a chain code

Format of a chain code: The format of a chain code consists of fewer than 5 English letters from A to Z in a one-dimension order. Figure 11 gives an example of the format.

Importance of the first code: The first code in a chain code is very important in encoding because it stands for the category and affects filling order of a chain code. Due to arrangement of chain codes following order of their first codes in the code database, the first code with a good reliability will greatly reduce the possibility of multi chain codes corresponding to one character. In addition, uncertainty of the first code will bring undefined priority into the filling order.

Omitted codes: A redundant code has to be omitted while the total number of the codes in a chain code is more than 5. In the other word, the structure of 5 codes has ensured the identity of a chain code. The priority of omitted codes is defined using a discontinuous selection in the last part of a chain code, as shown in Figure 12. The discontinuous selection is considered easier to be interpolated if the complete structure of a chain code is required.

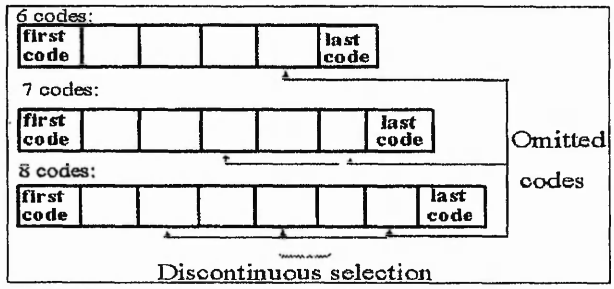


Figure 12. Selection of omitted codes

Necessity of the last code: The last code sometimes plays a necessary role while a chain code is formed by complex rules. In this case, the last code is a referring point to any omitting code; i.e. an omitted code will be located at one before the last code.

Possible code : A possible code is regarded as a replacement by guess while a replaced code has had a low reliability on recognition. In general, a possible code can get a better result if its position is located in the middle of a chain code. The quality of a whole chain code will be affected while a replacement is put in the first code.

Impossible position of a code: An impossible position of a code means that the structure of a chain code is contrary to one in Chinese characters. Figure 13 gives an example of impossible position. In this case, the code has to be regarded as failure.

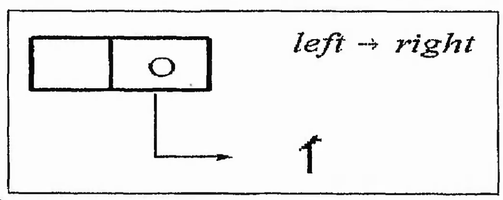


Figure 13. Impossible position

Quality of a chain code: The quality of a chain code depends on three factors: success of filling order, high reliability and result of matching to the code database. A

Quality	Code reliability	Matching to database	Filling order
* * *			
order	0 recognition	0 match	0 success
matching	1 mis-recognition	1 possible	1 failure
reliability	2 failure	2 failure	

Figure 14. Standard of quality of a chain code

combination of three factors will consist of the quality. Figure 14 lists relations of the quality and three factors.

6. EXPERIMENTAL RESULTS AND DISCUSSION

In order to show the effectiveness of the CCRS system based on fuzzy possibilistic rules for deconstructed a character, the associative memory neural network for classifying and recognising radicals and the chain code method for restructuring a character, some experiments have been concluded on a Sun workstation with X View function and C language to program algorithms of the system. Characters are input in a bitmap format. In the processing, the experimentation was focused on different extraction, classification, recognition, and restructuring of characters.

6.1 Results for Extraction

Test characters in printed and standard writing with different structures were used to examine the extraction rules. These characters were chosen with representation of radicals in different position, shape and order. The correct radicals have been extracted and achieved at a 96% success rate. In the implementation of the extraction processing, a dynamic scheme was employed for updating possibilistic inference rules while they were being evolved around a sample of special cases.

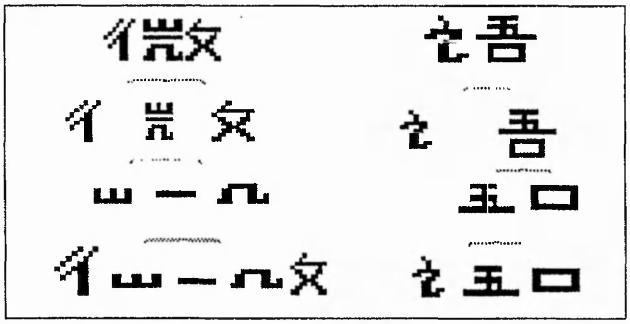


Figure 15. Examples for expansibility of rules

In addition to dividing a character as a picture into several radicals, i.e. sub-pictures, some sub-pictures might be further segmented into smaller sub-pictures, i.e. sub-subpictures, while the sub-pictures are combined of two or more radicals. The segmentation is termed expansibility of possibilistic rules as described as the complex distribution in Section 3.2. Examples in Figure 15 have shown implementing results of the expansibility.

6.2 Results for Classification

120 radicals covering 24 radical categories were used to examine classification and recognition of the system. The implementation of recognising radicals included two phases: learning and training. The learning phase was concentrated on effect of different weights and error tolerance of the network. The training phase was examined optimal architecture and convergence of the network.

Learning phase: Radicals classified into 24 categories were learnt by the learning phase of the network for forming connectivity of its sub-nets in the hidden-1 layer. Some learning patterns are shown in Figure 16.

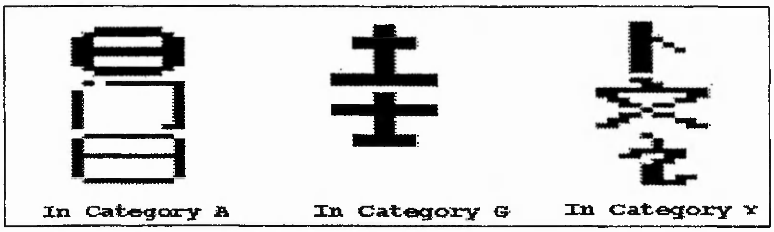


Figure 16. Some learning patterns

Using the better weight parameters, 74 radicals in 24 categories were learnt and formed the 24 sub-nets in the hidden-1 layer of the associative memory neural network. The partial weights of the sub-nets are shown in Figure 17.

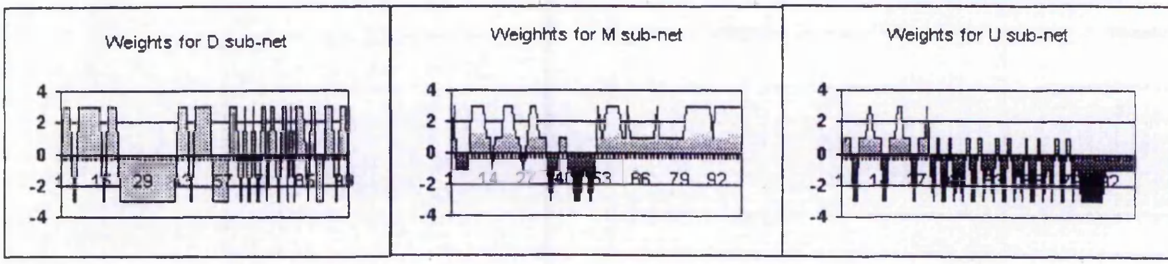


Figure 17. Weights for some sub-nets

Training phase: Training the network centred on the structure of neurones in the hidden-2 layer and improvement of the global convergence.

112 out of 120 radicals have been recognised by using the structure of the network, where 74 radicals were different from each other. Figure 18 gives the **recognition rate** of these **radicals** (both indicated by the y-axis) in different **categories** indicated by the x-axis). Figure 19 shows the recognition frequency of the radicals in the categories.

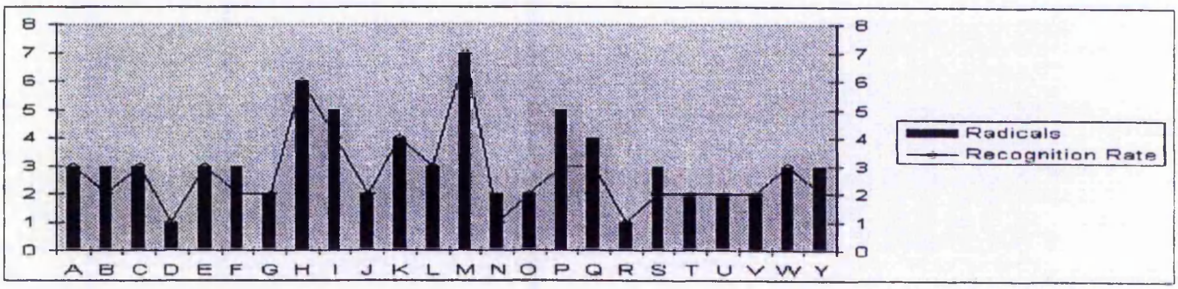


Figure 18. Recognition rate of radicals in different categories

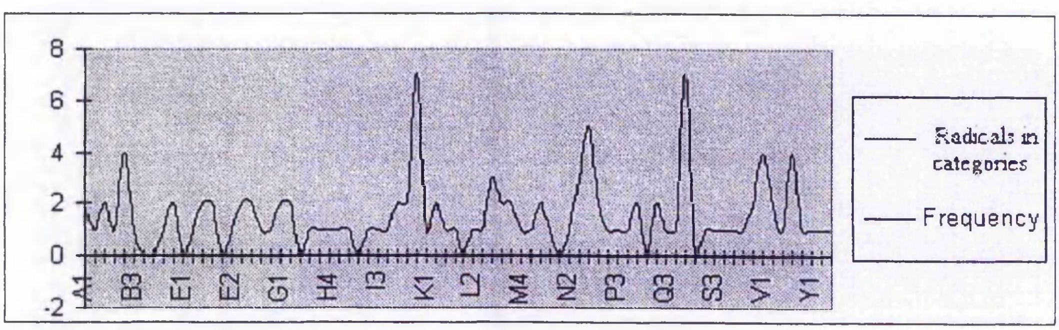


Figure 19. Recognition frequency of the radicals in the categories

6.4 Results for Restructuring

75 codes in different errors for combination of 10 chain codes were used to test the verification. 63 out of 75 test codes have been successfully verified. The verification of chain codes using the code database was easily

failed if a code in a chain code was not in a correct position, especially, the first code in a chain code was wrong. Figure 20 gives test results showing which position is most likely to cause a failure.

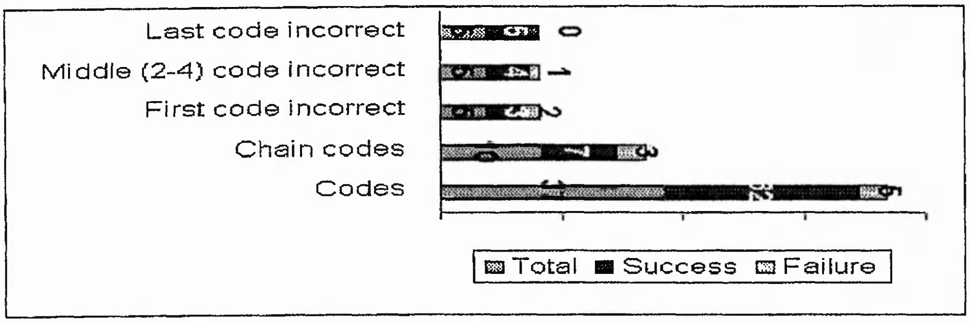


Figure 20. Test results on verifying chain codes

6.5 Discussion

Although some successful results have been obtained by the implementation of the system, some problems and special cases need to be discussed for further improvement.

Extraction: In analysing the extraction results, two types of incorrect results that appeared in the extraction process need to be carefully investigated. Incorrect results are involved in two special cases of (a) an individual radical with a discontinuous shape, and (b) a shape of two radicals connected together without a discontinuous part, shown in Figure 21, for example.

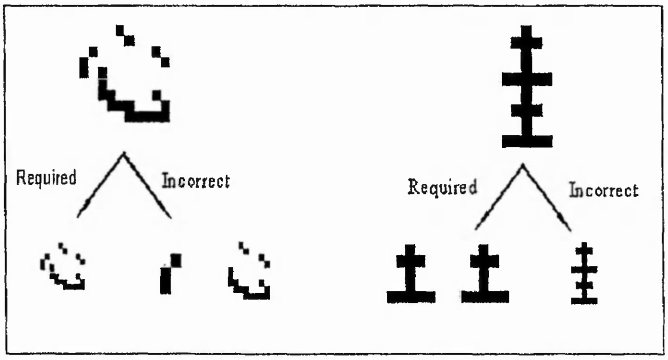


Figure 21. Two special cases

As for the first case, a complete radical is divided into two parts by current inference rules. There are two methods to deal with this problem according to human analysis. One method is to have special rules, probably against existing rules, to deal with these special radicals. The other is to attach new rules that could examine a rectangle area occupied by a radical. If the area is small enough only for a stroke rather than a radical, the extraction in this case will be invalid or treated it as a single radical without extracting. However, it needs a statistic value to decide a minimum area for tolerating a radical because only some of strokes can be treated as a radical.

In the second case, two radicals are connected, or overlaid together, and form a continuous shape. This is very difficult to deal with by only applying inference rules. Other methods should be further investigated for exploring these radicals. Currently, such radicals are treated as difficult ones. Obviously, they are not in the human observation.

Recognition: The associative memory algorithm has advantages of quicker convergence speed and the capability of recognising error patterns. According to the statistics of outcomes from the above trials, the number of iterations is less than 4 for reaching recognition. The capability of recognising the error patterns can be high up to as much as 60%.

The ambiguity of recognising patterns is basically caused by converging local minimum, especially, when two patterns had the same reliability rate. There is also another case of ambiguity as shown in Figure 22. The difference between patterns (a) and (b) might be treated as noise causing ambiguity of patterns to occur.

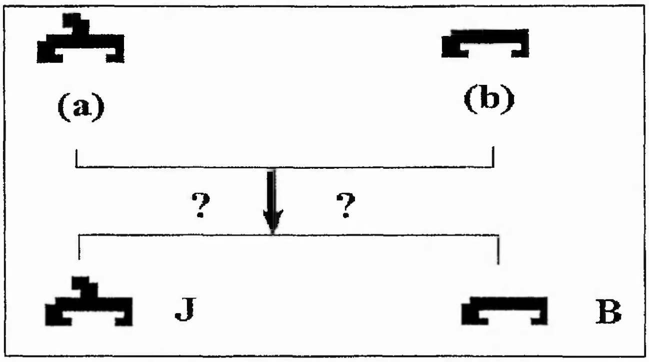


Figure 22. A case of ambiguity in Recognition

Restructuring: Apart from some results of mis-recognition caused by the Recognition process, the Restructuring process has the capability of forming a chain code for verifying a character. Although the quality of a chain code has provided a reference for searching for possible solutions as results of matching to the database, the ambiguity of solutions is often happened, especially, if the solutions were for the first code in a chain code. More human knowledge can be considered to enhance the restructuring cues.

7. CONCLUSIONS

In this paper, an algorithm of using the three-layer hierarchy method and the three-layer hierarchy process to recognise Chinese characters has been applied to the Chinese Character Recognition System (CCRS). The techniques of fuzzy possibilistic rules to extract radicals from characters, associative memory neural network to classify and recognise radicals, and the chain code method to restructure the radicals into a character for verification have been used to support the system. Several special cases and application limitation of the system are discussed. Implemented results show that the system is effective and reasonable. Further development will be focused on improvement of extracting rules, classification of radicals and dealing with mis-recognition cases.

8. REFERENCES

[1] M. Ren et al., "A hierarchical approach to fuzzy possibilistic-reasoning for recognising Chinese characters", *International Journal of Computer Mathematics*, Vol.64, No.1&2, 1997, pp.17-34.

[2] D. Al-Dabass et al., "An associative memory artificial neural networks system with a combined-radical method for Chinese character recognition", in *IEEE-ICONIP'95-Beijing: Proceedings of International Conference on Neural Network Information Processing*, Beijing, China, Vol.2 of 2, 1995, pp.857-860.

[3] E. Scurfield, *Chinese*, Hodder and Stoughton, UK, 1991.

[4] R. Kruse, *Foundations of Fuzzy Systems*, John Wiley & Sons Ltd, UK, 1994, pp.81-155.

[5] J. A. Freeman and D. M. Skapura, *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley Publishing Company, 1992, pp.373-393.

[6] Y-M. Liu, *Introduction of Cang-Jie Coding Method*, Ru Lin Ltd, Taiwan 1993. (In Chinese)

[7] J. E. Dayhoff, *Neural Network Architectures: An Introduction*, Van Nostrand Reinhold, USA, 1990, pp.37-57.

[8] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", in *Proc. Ntl. Acad. Sci. USA*, Vol.79, 1982, pp.2554-2558.

[9] T. Wang, "Improving recall in associative memories by dynamic threshold", *Neural Networks*, Vol.7, No.9, 1994, pp.1379-1385.

[10] Y. A. Bengio, "Connectionist approach to speech recognition", *Advances in Pattern Recognition Systems Using Neural Network Techniques*, World Scientific, USA, 1993, pp.3-24.

Intern. J. Computer Math., Vol. 64, pp. 17-34
Reprints available directly from the publisher
Photocopying permitted by license only

A HIERARCHICAL APPROACH TO FUZZY POSSIBILISTIC-REASONING FOR RECOGNISING CHINESE CHARACTERS

MANLING REN[a], DAVID AI-DABASS[a] and DAIZHONG SU[b]

[a]*Department of Computing, The Nottingham Trent University, Burton Street, Nottingham NG1 4BU UK;*
[b]*Department of Mechanical Engineering, The Nottingham Trent University, Burton Street, Nottingham NG1 4BU UK*

(Received 2 February 1996)

The paper presents a system based on a new paradigm of a three-layer hierarchy with fuzzy possibilistic reasoning rules and associative memory (AM) neural networks to recognise radicals in Chinese characters. The system extracts features of the complex topological structure of characters and constitutes a hierarchical representation based on their physical shape and classified according to their logical meanings and pattern structure. Using fuzzy possibilistic reasoning rules, the system applies this hierarchy to identify individual radicals embedded within a character. An associative memory neural network is employed to recognise identified radicals. Comparison of this system with other methods is carried out. Several special cases and application limitations of the system are discussed. Test results show that the system is effective and reasonable.

Keywords: Associative-memory neural network; chinese character recognition; fuzzy possibilistic reasoning; inference rules; extraction

C. R. Categories: F1.1, I2.3, I5.1, C1.3

1. INTRODUCTION

A fundamental aspect of computer recognition of Chinese characters centres on a successful knowledge representation structure (1). These characters are essentially pictographic and pose several difficulties including (i) the large volume of characters to be processed, (ii) the complicated internal structure of a character, as well the usual difficulty (iii) that text characters are two-dimensional information structures.

Generally, a Chinese character consists of basic 'stokes' which include: dot, horizontal, vertical, left-diagonal, right-diagonal, rising, hook and turn, as shown in Figure 1. Any two or more of these strokes may be combined to form a new structure as part of a character, where a character may constitute any number from one to sixteen strokes, with some having as many as thirty strokes or more.

The basic rules of strokes order in writing Chinese characters are: first horizontal, then vertical; first left-diagonal, then right-diagonal; from top to bottom, from left to right; first outside, then inside; finish inside, then close; first middle, then two sides. Figure 2 gives examples illustrating these rules.

The structure representation of Chinese characters may follow one of three paradigms: a two-layer hierarchy, a three-layer hierarchy and the four-corners method. The two-layer hierarchy is seen to relate strokes directly to characters. The number of strokes contained in a character forms the basis for a relationship between these strokes and the character. As a traditional

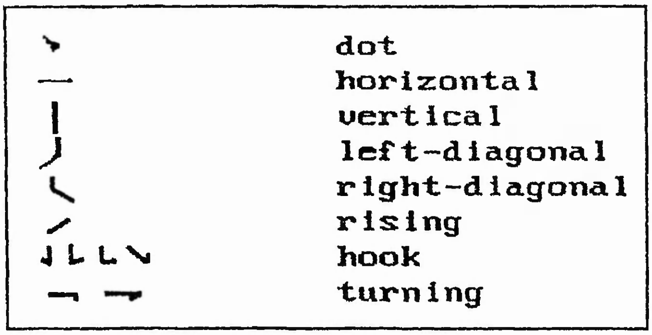


FIGURE 1 Basic Strokes.

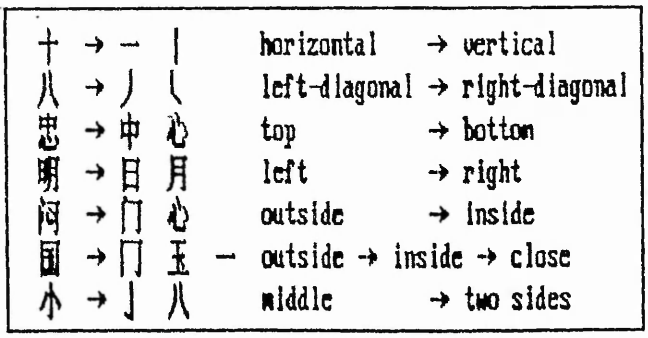


FIGURE 2 Basic Rules.

method of learning Chinese characters in the real world, this paradigm has been reported in recent and current research literature of Chinese character recognition (2). The two-layer hierarchy has the advantages of (i) only few strokes to be learnt and (ii) each stroke could be considered as a simple one-dimensional element. However, there are two major weaknesses: (i) the large number of strokes in a character, and (ii) the internal structural relationships between strokes and characters are rather difficult to describe by the two-layer paradigm. The second weakness leads to other difficulties, such as: combination format of strokes in a character, their positions and identifying the same cluster of strokes in different characters.

The three-layer hierarchy represents the structure of Chinese characters in three formats: characters, radicals and strokes. Radicals are introduced as common components of characters, where each radical is made up of a number of different strokes. The three-layer hierarchy reduces the complexity of a character's structure: a character would consist of less than five radicals (3) while these in turn would individually contain some two to eight strokes. Furthermore, because of position independence of radicals within a character, a character as a picture might be divided into several individual sub-pictures, which considerably eases further recognition by machine.

The four-corners method uses a number of codes to represent Chinese characters, where each code is a four-digit number and each digit (0 to 9) stands for a stroke or combination of strokes from one of the four corners of a character respectively. The four-corners method has a good structure of character classification with a grammar based on horizontal and vertical crossings. It is possible to classify 6763 characters into 1586 groups with 97.5% accuracy (4). The weakness of the method includes complex knowledge of its grammar (based on human observation) and the ambiguity of the code used to represent several different characters.

In this paper, the three-layer hierarchy is investigated in order to simplify the processing of computer recognition of these characters. Due to the characteristics that a Chinese character is a two-dimensional picture and varies its structure in different ways, difficulties of applying the hierarchy include (i) determining the shape, position and size of a radical in a character, which could be seen as the vagueness of a radical, and (ii) recognising the radical extracted from a character which focuses on the one-to-many relationships, i.e. the ambiguity of a radical, between radicals.

A fuzzy neural network system using possibilistic reasoning and associative memory (AM) algorithms has been developed for coping with the above difficulties. Based on the application of the three-layer hierarchy, the system consists of two parts. In the first part, fuzzy inference rules based on possibilistic reasoning are applied for dealing with the vague shape and position of a radical in a character. In the second part, a neural network algorithm based on the Hopfield and bidirectional associative memory function is used to eliminate noise embedded in a radical (extracted from a character), in order to recognise it (5). In this paper, techniques of normalising varying sizes of a radical are not presented. Therefore, the same size for testing radicals is assumed in the implementation of the system. Figure 3 illustrates the architecture of the two parts in the system.

The paper is organised as follows. The structure features and representation scheme of the hierarchy are described in the Section 2 and Section 3 respectively. The associative memory neural network is presented in Section 4. The experimental results, produced by an algorithm based on the application of the hierarchy, fuzzy possibilistic reasoning rules and AM neural network, together with discussions of several special cases are presented in Section 5. A preliminary comparison of the hierarchy with other methods and some suggestions for further development are given in Section 6.

2. A THREE-LAYER HIERARCHY

The structure representation in the three-layer hierarchy centres on the relationships of objects in these layers using two aspects: extraction and classification. Extraction takes into account the relationship of objects in their shape alteration between layers, which is also called vertical alteration. In the second aspect, classification generalises the relationship of objects in

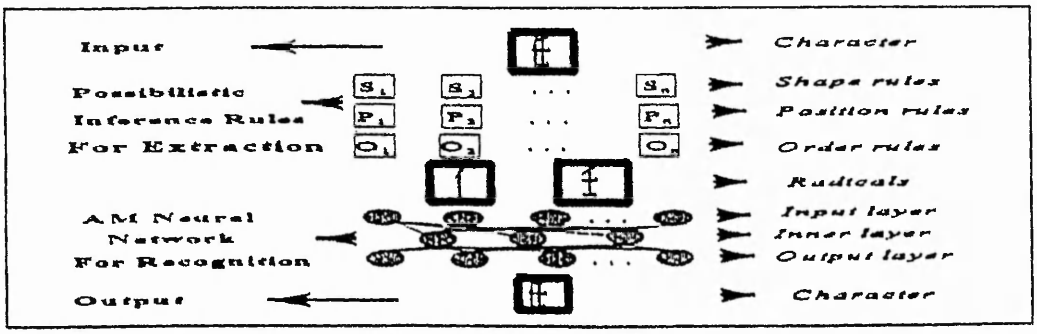


FIGURE 3 A Fuzzy Neural Network System with Possibilistic Reasoning and Associative Memory Architecture.

the same layer and forms different categories of these objects, which is termed horizontal classification.

2.1. Characters

The analysis and statistics of characters's structure in this layer are based on the historical development of Chinese characters. Chinese characters are derived from stages of symbols, pictographs to characters and ideographs. Because the symbols are drawn from actual objects in the real world, most of current Chinese characters still inherit the features of a two-dimensional picture derived from these early symbols. The development of pictographs began to represent some abstract concepts of daily life and eventually made original characters. Figure 4 gives examples to show the development.

Subsequently, two or more pictographs were combined to form new characters or ideographs for more complex ideas and concepts. Currently, 90% of characters in common use for writing (7000 characters) consists of two or more original characters. At the same time, some of the original characters still keep as individual characters and some of them degenerate as only components of a character. In current Chinese dictionaries, most of these original characters are termed radicals. Figure 5 shows several such characters.

In summary, the structure of characters has features of a two-dimensional picture, combination of components and different characters containing the same components.

2.2. Radicals

There are two parts included in the layer of radicals: radicals and token-radicals. According to the analysis of characters explained above, a radical

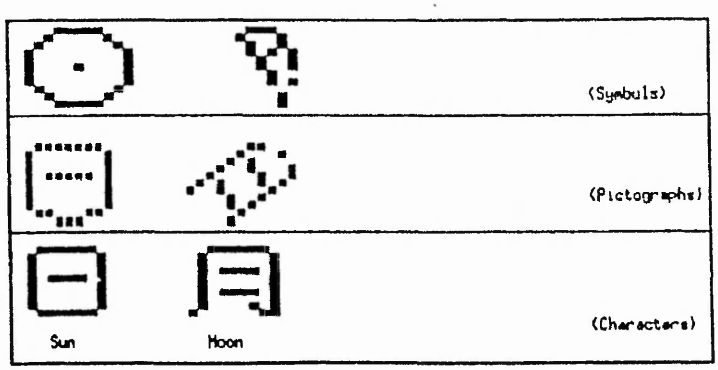


FIGURE 4 Symbols, Pictographs and Characters.

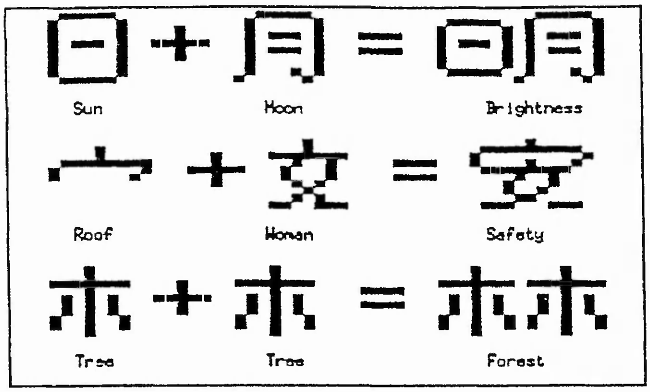


FIGURE 5 Combined Characters.

may inherit its individual meaning and occupy a completely independent position in a character. A token-radical is a simplification of such radicals which ultimately become simple combinations of basic strokes. Although some of these are overlaid or crossed in a character, token-radicals common to many characters might be classified into a certain category of radicals in terms of their abstract concepts and shape structure. In the three-layer hierarchy, radicals and token-radicals are classified into 26 different categories (24 radicals' categories and 2 special case categories). An example of radical categories in shown in Figure 6.

Whereas radicals are made up by basic strokes in the lower layer, the extraction and classification of characters in the hierarchy is well organised by the strength of radicals and token-radicals. For instance, classification of a character might depend on a combination of its radicals or token-radicals. Therefore, radicals and token-radicals form the basis of the structure representation in the three-layer hierarchy.

2.3. Strokes

In the three-layer hierarchy, a stroke is defined to be a continuous line shown in Figure 1. Although radicals form the foundation of extraction and classification of the hierarchy, the structure of their shapes has to be defined entirely by these primitive strokes. For instance, a combination of a horizontal and vertical strokes might from a token-radical, shown in Figure 7. Therefore, the structural representation in this layer will concentrate on these combinations as well as strokes themselves.

On the other hand, because combinations of strokes determine the structure of a character as the appearance of a rectangle or a square rather

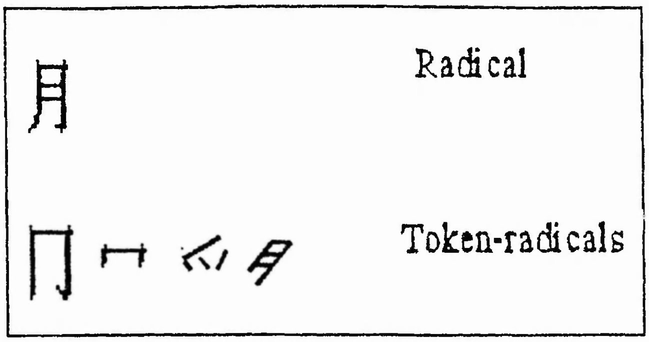


FIGURE 6 A Radical 'Moon' Category.

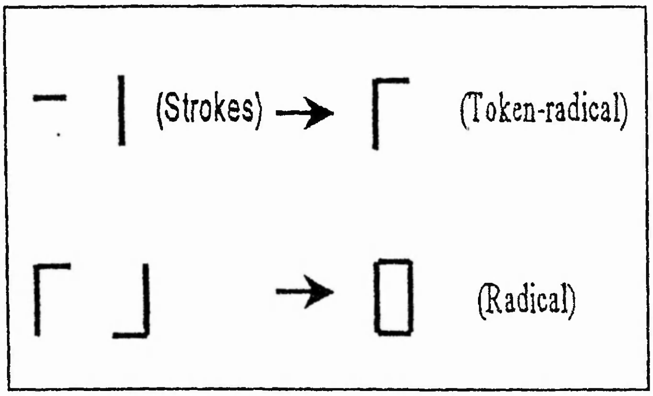


FIGURE 7 Strokes Combination.

than one of a curve in some other alphabetical characters, the structure description of a rectangle or a square will be taken to form the foundation of possibility distribution of a character's shape.

The basic rules of strokes order, shown in Figure 2, are developed as the essential rules for the structural representation of the hierarchy. In representing the order and position of a radical in a character, these rules will be applied as antecedent conditions.

3. REPRESENTATION SCHEME

The three-layer hierarchy has recently been successfully applied to the derivation of an algorithm to form the nucleus of a Chinese character recognition system (6) (7). The algorithm is based on the application of fuzzy possibilistic reasoning, the syntactic method and a combination of neural networks, the Cang-Jie method and a database for extraction, classification and recognition. In this section, the representation will focus on the extraction of characters into radicals using fuzzy possibilistic rules to incorporate the aspect of vagueness of character structure into the system. The system uses two representation methods: situation-rules to describe the position of a radical, and shape-rules to deal with shape domain of radicals, possibility distribution and measure techniques.

3.1. Fuzzy Possibilistic Reasoning

Fuzzy possibilistic reasoning for knowledge representation is well suited to dealing with imperfect, uncertain and vague information (8). Reducing the complexity of imperfect information is achieved by information-compressed representations based on if-then rules. These rules are interpreted as logical implications and are termed as possibilistic inference rules defined by the notation $\mathcal{R}$.

Based on fuzzy set theory, approximate reasoning and probability theory, fuzzy possibilistic representations use conjunctively combined rules to validate a possible resolution from various restrictions. In these if-then rules, antecedent in the *IF* clause and consequence in the *THEN* clause are constrained by their possibility distributions denoted by π. The possibility distributions are related to the interpretation of vague concepts as contour functions of random sets. Physical quantities of the distributions are defined by the possibility measures denoted by Poss_π.

Generally, a possibilistic inference rule $\Re_j$ can be expressed by

$$\Re_j : \text{IF } \xi_j^s \text{ is } \mu_j \text{ THEN } \xi_j^T \text{ is } \nu_j, \quad j = 1, \ldots, r,$$

or

$$\Re_j : \text{IF } \xi_j^{s(1)} \text{ is } \mu_j^{(1)} \text{AND } \xi_j^{s(2)} \text{ is } \mu_j^{(2)} \text{ THEN } \xi_j^T \text{ is } \nu_j, \quad j = 1, \ldots, r,$$

where $\mu_j, \mu_j^{(1)}, \mu_j^{(2)}$ and ν_j are subsets of possibility distributions on the space sets S_j and T_j with regard to j respectively. ξ is a variable whose values can be arbitrary possibility distributions on S_j or T_j. The symbol is, appearing in possibilistic inference rules, serves as a linguistic description of the operator and is therefore to be seen as 'is at least as specific as'.

The relation $\Re$ of all rules is

$$\Re = \bigcap_{l=l}^{r} \Re_j$$

3.2. Situation Representation

The situation representation method uses inference rules defined by the above interpretation for determining a radical's position. The representation focuses descriptions on (i) the position of a radical in a character, and (ii) the order of extracting a radical from a character.

Let the notation P and O stand for two domains of the position of radicals in a character and the order of extracting a radical from a character respectively, their possibility distributions could be defined as $\pi(P)$ and $\pi(O)$ according to the definition of fuzzy possibility reasoning. The possibility measures are given by the notations $Poss_\pi(P)$ for $\pi(P)$, and $Poss_\pi(O)$ for $\pi(O)$. The possibilistic inference rules are represented by the notation $\Re^{(PO)}$. If p and o denote variables with the domains P and O respectively, the $Poss_\pi(p)$ is the possibility measure of p on $\pi(P)$; similarly, $Poss_\pi(o)$ for o on $\pi(O)$.

3.2.1. Position Variance

The investigation of position variance of radicals in a character is based on their features of a two-dimensional picture and a rectangular appearance, one of the major characteristics in the structure of Chinese characters. The domain of position variable is defined by

$$P = \{\textit{width}, \ \textit{length}\}.$$

Because a radical may keep an independent position in a character, the possibility distribution of position variance of a radical on the domain P, shown in Figure 8, is defined by

$$\pi(P) = \{\textit{outside}, \ \textit{inside}, \ \textit{top bottom}, \ \textit{left}, \ \textit{right}, \ \textit{middle}\}$$

$Poss_\pi(P)$ for $\pi(P)$ is defined by, for instance,

$$Poss_\pi\ (\textit{left} = \{\textit{width} \leq 2/3 \ \textit{width of P}, \ \textit{length} = \textit{length of P}\}.$$

3.2.2. Extraction Order

The extraction order indicates the sequence of radicals extracted from a character which might consist of two or more radicals. The domain of

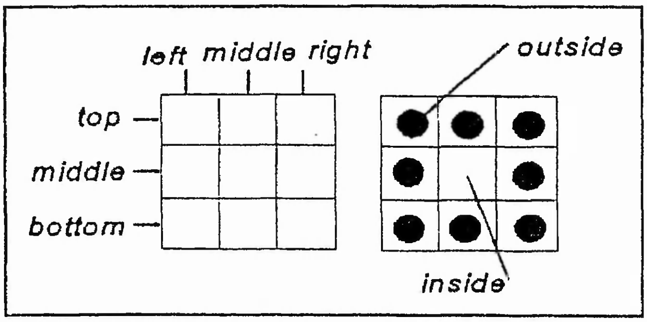


FIGURE 8 Possible Positions of a Radical.

extraction order is expressed by

$$O = \{\textit{first}, \ \textit{last}\}$$

The possibility distribution of extraction order on the domain O is represented by

$$\pi(O) = \{\textbf{\textit{outside}} \to \textbf{\textit{inside}}, \ \textbf{\textit{inside}} \to \textbf{\textit{outside}}, \ \textbf{\textit{top}} \to \ \textbf{\textit{middle}} \to \textbf{\textit{bottom}}, \ \to \textbf{\textit{bottom}}, \ \textbf{\textit{left}} \to \textbf{\textit{middle}} \to \textbf{\textit{right}}, \ \textbf{\textit{left}} \to \textbf{\textit{right}}\}.$$

The notation '$\to$' stands for the sequence from the first to the latter. Distinction of some distribution representations, such as, 'outside$\to$inside' and 'inside$\to$outside', will depend on inference rules between order, position and shape mentioned in the next section.

Developing such basic possibility distribution of order $\pi(O)$ above, a complex distribution could be derived, for instance,

$$\pi(O)^{(1)}(\textit{top} \to \textit{bottom} \ (\textit{left} \to \textit{right})) = \{\textit{top} \to \textbf{\textit{bottom left}} \to \textbf{\textit{bottom right}}\}$$

$\mathrm{Poss}_\pi(O)$ for $\pi(O)$ is defined by, for instance,

$$\mathrm{Poss}_\pi \ (\textit{top} \to \textit{bottom}) = \{\textbf{\textit{there are two rectangles}}\}.$$

Now, possibilistic inference rules $\Re^{(PO)}$ might be established for representing relations between the position and order of a radical. As examples, several rules are shown as follows.

$\Re^{(PO)}{}_{(1)}$: IF position is top THEN order is first,

$\Re^{(PO)}{}_{(2)}$: IF position is bottom THEN order is last

3.3. Shape Representation

The shape representation method centres on the shape domain of radicals, possibility distributions and measure technique. Inference rules are established for the representation of radicals relationships between their shape, position and order.

3.3.1. Possibility Distributions and Measures

The domain of radicals is defined as a rectangle in different sizes in terms of features of combined strokes. The shape domain of radicals is expressed by

$$\mathrm{S} = \{\mathit{rectangle}\}.$$

The possibility distributions on the domain S are assigned by different valid combinations of basic strokes, where the validity of the combinations is checked. The modes of combinations are classified as connection and disconnection. The possibility distributions are represented by

$$\pi(\mathrm{S}) = \{\textit{combination of basic strokes},\ \textit{basic strokes}\}.$$

In order to determine the shape of a radical, possibility measures are based on evaluation of a continuous line, direction of a line connecting with other lines, priority of such direction and disconnecting distance. For instance, one of the possibility measures $\mathrm{Poss}_{\pi}(\mathrm{S})$ for $\pi(\mathrm{S})$ is defined as follows:

$$\mathrm{Poss}_{\pi}\,(\textit{priority of up} \rightarrow \textit{down}) = \{\textit{up} \rightarrow \textit{down},\ \textit{up} \rightarrow \textit{down left},\ \textit{right} \rightarrow \textit{left}, \textit{up} \rightarrow \textit{down right}\}$$

3.3.2. Shape Vagueness and Possibilistic Inference Rules

To produce general concepts of forming a radical, the shape vagueness of radicals is investigated for expressing the relation of combining two strokes. The relations can be classified as angle, location, continuous, distance and discontinuous.

The angle relation indicates a contour expression of two connected strokes. For example, it is defined as a contour if an angle is formed by two connected strokes. Figure 9 shows three different types of angles from two connected strokes.

The location relation stands for a place of two connected strokes. Figure 10 gives several examples to show the location relation.

The continuous relation expresses a possibility of a contour as part of a radical. A continuous contour is defined if a contour is formed with an angle.

The distance relation is to measure a scope of two disconnected strokes. The discontinuous relation implies the possibility of a contour which may form two radicals from their shape. A discontinuous contour is decided by the distance measure of two disconnected strokes.

Possibilistic inference rules are established by representations of relations between shape vagueness denoted by $\Re^{(S)}$, between shape and position by $\Re^{(SP)}$, and between shape, position and order by $\Re^{(SPO)}$. For example, the inference rules shown below are defined to divide a character into two parts from the inside to outside.

$\Re^{(S)}_{(1)}$: IF contour $c1$ is square AND $c2$ is continuous contour of $c1$
AND angle of $c1$ connecting with $c2$ is 90
AND location of $c2$ is on the top middle of $c1$
THEN shape is combination of $c1$ and $c2$ ($c1 + c2$).

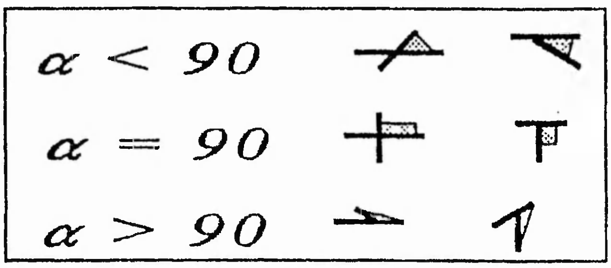


FIGURE 9 Angle of Strokes Connected.

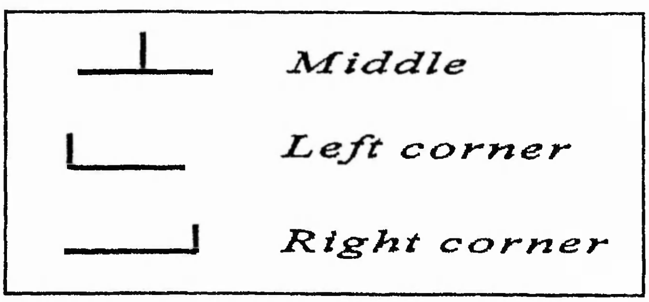


FIGURE 10 Location of Strokes Connected.

$\Re^{(S)}_{(2)}$: IF shape c1 is $c1 + c2$ THEN $c1$ position is outside.

$\Re^{(SP)}_{(3)}$: IF shape is $c1 + c2$ THEN $c2$ position is inside.

$\Re^{(SPO)}_{(4)}$: IF shape is $c1+c2$ AND position is outside THEN order is last

$\Re^{(SPO)}_{(5)}$: IF shape is $c1+c2$ AND position is inside THEN order is first

4. AN ASSOCIATIVE MEMORY (AM) NEURAL NETWORK FOR RECOGNISING RADICALS

The neural network in the system consists of three layers: input, inner and output. The number of neurons on the input and inner layers is decided by one of the learning patterns. Neurons on the inner and output layers are dynamically created as a result of combining input radicals.

The network possesses an associative memory function, which offers an important advantage of recalling a stored pattern from its partial or noisy input. Two different methods are applied in association with the function. A static method with the Hopfield auto-associative algorithm is used to the input and inner layers. A dynamic method with the bidirectional hetero-associative algorithm is employed in the inner and output layers. The mathematical formula for the associative memory function is established on the construction of an energy equation,

$$E = -\sum_i \sum_j X_i W_{ij} Y_j + \sum_i \theta_i X_i + \sum_j \varphi_j Y_j \tag{4.1}$$

whose value is reduced or remains constant during the recall procedure (9).

In the learning phase, the associative memory function is used to form the network's connection matrix W. For the training set of input patterns $X_i(u)$ and output patterns $Y_i(u)$, where $u = 1, 2, \dots M$; $i = 1, 2, \dots N$, the weight $W(i,j)$ is determined by the Hebbian rules,

$$W(i,j) = \begin{cases} \sum_u X_i(u)\, Y_j(u) & \text{if } i \neq j \\ 0 & \text{if } i = j = j \end{cases} \tag{4.2}$$

When a portion of the original pattern is used as a retrieval cue, the algorithm is denoted as auto-associative. When the desired output is different from the input, the algorithm is called hetero-associative. The static method requires that the network is trained in advance with specific patterns, and retrieves them from portions or distorted inputs. When the learning and retrieval are embedded in the training process randomly, the method is defined as dynamic. The dynamic bidirectional method used in the system aims to recognize an input pattern that has a relationship with the others.

The network learns standard radicals in the learning phase and identifies them in the training phase. In the recognising process, the network also provides its recognising results with a parameter, which is called the reliability rate, for a further reference. There are three possibilities of recognising a result: recognition, mis-recognition and failure. The reliability rate produced depends on the matching quality and recalling quantity of a pattern.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to test the effectiveness of extraction rules and the associative memory neural network, some experiments have been conducted on a Sun workstation with X View function and the C language to program the algorithm.

In the extraction processing, the experimentation was focused on different structures of radicals in a character and the technique of extracting them. In the neural network interface, the system was examined by the effect of different weight factors and error tolerance. Moreover, optimising the neurons' structure and analysing results of convergence were investigated.

5.1. Experimental Results for Extraction Rules

In the first part, 50 printed characters with different structures were used to examine the extraction rules. The correct radicals have been extracted from 48 out of 50 test characters, i.e. a 96% success rate. Some of these are shown in Figure 11. The two types of incorrect results will be discussed in Section 5.3.

In the implementation of the first interface, a dynamic scheme is employed for updating possibilistic inference rules while they are being evolved around a sample of special cases.

5.2. Test Results for AM Neural Network

The test data for the AM neural network is divided into three parts: (i) different weight factors on recognition; (ii) improving the global convergence; and (iii) reducing the mis-recognition rate.

Table I gives results of different weight factors, and their effects on recognition for test data, shown in Figure 12.

Three groups of test data with different error rates were used for examining the error tolerance of the system. The error rates were 10%, 20% and 60% respectively.Table II gives the recognition results for this trial.

The results in Table I and II show a better rate of recognition while the weight M is 3. Figure 12 shows some of learning, testing radicals and recognition results.

Table III shows results for the convergence of local minima onto global minimum for an input radical with different parameters which refer to the reliability rate, while parameters θ_i and φ_i in the equation 3.1 are defined as 0.

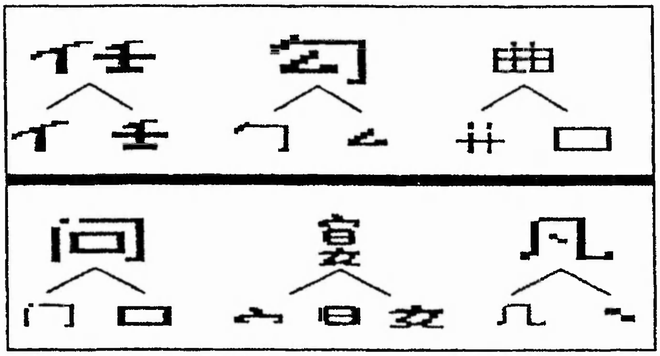


FIGURE 11 Extracted Results of Characters with Different Radicals.

TABLE I Results of Different Weights

Weight M	2	3	4
Input Number	4	4	4
Error Rate	10%	10%	10%
Recognition	4	4	4
Mis-Recognition	426	0	1
Failure	0	0	0

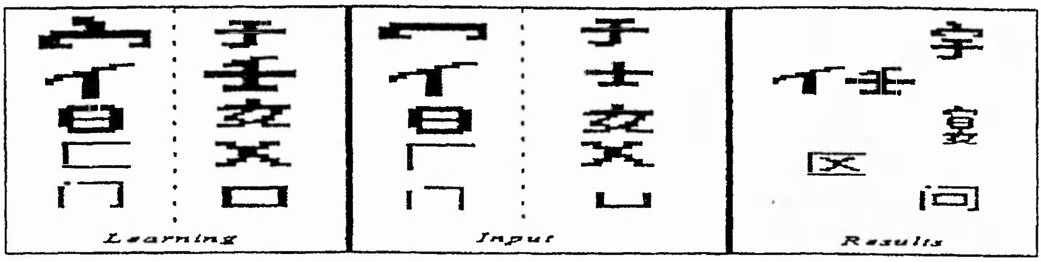


FIGURE 12 Recognition of Different Radicals.

TABLE II Results of Test Data with different Error Rates

Weight M	3	3	3
Input Number	4	4	4
Error Rate	10%	20%	60%
Recognition	4	4	3
Mis-Recognition	0	1	2
Failure	0	0	0

TABLE III Test Results of Modifying Associative Memory Function

Input Number	5	5	5
Error Rate	0%	10%	20%
Weight M	3	3	3
Number of Iterations	1	3	2
Local Minimum	5	14	7
Global Minimum	5	7	5
Reference to Learning Patterns	5	8	7

After modifying parameters, θ_i and φ_j to $1/2\Sigma W_{ij}$, the mis-recognition rate of these test data is reduced. Table IV gives results of the enhancement.

5.3. Discussions

In analysing the above results, two types of incorrect results that appeared in the extraction processing need to be carefully investigated. Incorrect results are involved in two special cases: (i) an individual radical with a discontinuous shape, (ii) a shape of two radicals connected together without a discontinuous part shown in Figure 13 for reference.

In the first case, the radical might be divided into two parts by current inference rules. Therefore, it is necessary to have some special rules, perhaps against existing rules, to deal with these special radicals.

TABLE IV Testing Results with Modifying Hebbian Results

Input Number	5	5	5
Error Rate	0%	10%	20%
Weight M	3	3	3
Number of Iterations	1	2	2
Local Minimum	5	7	5
Global Minimum	5	6	5
Reference to Learning Patterns	5	7	5

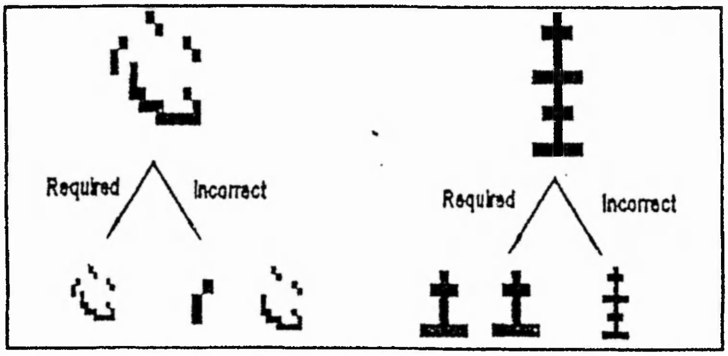


FIGURE 13 Two Special Cases.

In the second case, two radicals are connected, or overlaid together, and form a continuous shape. This is very difficult to deal with by only applying inference rules. Other methods should be investigated for exploring these radicals. Currently, there are two special categories for classifying such radicals.

In the second part, the main problem is mis-recognition. Modification of the associative memory function was enhanced to give better results as shown in Table III and Table IV. Consequently, further rejection for mis-recognising results is required for improving the associative memory function.

6. CONCLUSIONS

In this paper, a three-layer hierarchy of representing Chinese characters was introduced and a fuzzy neural network system using possibilistic reasoning and AM neural network architecture was developed. In the system implementation, the possibilistic reference rules and the neural network have been applied for different characters with different radicals. Test results obtained show the efficiency of the system. The modification of the associative memory function in several stages has enhanced the recognition rate of the system.

Compared with the two-layer hierarchy paradigm reported in recent and current and research literature of Chinese character recognition, the three-layer hierarchy offers the advantages of (i) having the capability of processing Chinese characters with a similar structure, (ii) a more systematic internal topological structure of a character and (iii) reducing the vocabulary of learning the characters by machine.

Moreover, although both of the three-layer hierarchy and the four-corners method have a good structure representation in the classification of characters, the ambiguity of classified characters is much higher in the four-corners method than the three-layer hierarchy.

Further research is needed to extend possibilistic rules and improve algorithms for reducing the mis-recognition.

References

[1] Tseng, L. Y. and Chuang, C. T. (1992). An Efficient Knowledge-Based Stroke Extraction Method for Multi-Font Chinese Characters, *Pattern Recognition*, 25(12), pp. 1445–1458.

[2] Hildebrandt, T. H. and Liu, W. (1993). Optical Recognition of Handwritten Chinese Characters: Advances Since 1980, *Pattern Recognition*, 26(2), pp. 205–225.

[3] Liu, W. (1993). *Introduction of Cang-Jie Coding Method*, Run Lin Ltd. (Chinese)

[4] Nagy, G. (1988). Chinese Character Recognition: A Twenty-Five-Year Retrospective, 9th International Conference on Pattern Recognition, **1**, Rome, Italy, pp. 163–167, *Sponsored by Int. Assoc. Pattern Recognition, IEEE Computer Society Press*, ISBN: 0818608781.

[5] Ren, M., Su, D. and Al-Dabass, D. (1995). An Associative Memory Artificial Neural Networks System, ECAC'95-London: Proceedings of 1995 European Chinese Automation Conference, London UK, 1995, pp. 91–96.

[6] Al-Dabass, D., Ren, M. and Su, D. (1995). An Associative Memory Artificial Neural Network System with a Combined-Radicals Method for Chinese Character Recognition, ICONIP'95-Beijing: Proceedings of International Conference on Neural Network Information Processing, **2**, of 2, Beijing, China, 1995, pp. 857–860, Co-Sponsored by Asia-Pacific Neural Networks Assembly (APNNA) IEEE Region 10, Technical Co-Sponsored by IEEE Communication Society, Publishing House of Electronics Industry, ISBN:7-5053-3355-0/TP.1288.

[7] Ren, M., Al-Dabass, D. and Su, D. (1996). Using a Syntactic Method to Partition a Chinese Character, *The Caledonian International Engineering Journal*, 96(1), pp. 51–60.

[8] Kruse, R., Gebhardt, J. and Klawonn, F. (1994). *Foundations of Fuzzy Systems*, John Wiley and Sons Ltd, pp. 81–155.

[9] Wang, T. (1994). Improving Recall in Associative Memories by Dynamic Threshold, *Neural Networks*, 7, **9**, pp. 1379–1385.

USING A SYNTACTIC METHOD TO PARTITION A CHINESE CHARACTER

Manling Ren*, David Al-Dabass* and Daizhong Su**,

* Department of Computing
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel: +44 (0)115 941 8418 Ext. 2761
Fax: +44 (0)115 948 6518
E-mail: mar@doc.ntu.ac.uk
dad@doc.ntu.ac.uk

** Department of Mechanical Engineering
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel: +44 (0)115 941 8418 Ext. 2306
Fax: +44 (0)115 948 6506
E-mail: mec3sud@ntu.ac.uk

ABSTRACT

This paper presents a system that applies a syntactic method for partitioning radicals from a Chinese character. The method applies static and dynamic rules to determine the shape, position and order of a radical. The background to the method, its definition and implementation are introduced and discussed. Results obtained show the efficiency and robustness of the system developed.

1. INTRODUCTION

In order to build a knowledge-based recognition system, many researchers apply expert system techniques to pattern recognition [Bunke and Sanfeliu, 1986]. The primary task of such a knowledge-based recognition system is to be able to sort pattern data into classes. Approaches used in the systems may be divided into two general groups: decision-theoretic (or statistical), and syntactic (or structure) [Hamilton, Pringle and Grant, 1991]. Decision-theoretic methods extract a set of characteristic measurement numbers, called features, from the pattern data. Syntactic methods employ a set of rules to describe the structure of a pattern class in terms of primitive features of the patterns.

Considering the differences between Chinese characters and other alphabetical characters, it is quite difficult to extract the feature of or to describe the structure of every Chinese characters. The difficulties include: (*i*) the large volume of Chinese characters to be processed; (*ii*) the complicated internal structure for each character; and (*iii*) that a Chinese character is a two-dimension picture [Liu and Tai, 1988].

To deal with these problems, a system called Radical Partitioning has been developed. It relies on the syntactic, Cang-Jie and combined-radicals methods for describing the complex topological structure of Chinese characters. A Chinese character is segmented into its radicals and typefaces for further processing. Application of these methods intends to solve problems of shape, position and order for a radical and a typeface, and to determine the category to which they belong.

2. ANALYSIS OF THE COMBINED-RADICALS METHOD

The method is based on analysing the topological structure of Chinese characters [Ren, 1994] and the coding rules used in the Cang-Jie method [Liu, 1993]. It contains two major phases: one is for partitioning a character; and the other is to combine radicals into a complete character. This paper focuses only on the first part of the method.

2.1 History of Chinese characters development

Chinese characters have a long historical development (3,500 years on recorded history) and have their own special characteristics. Generally, a Chinese character is composed of some basic strokes, which are presented as dot, horizontal, vertical, left-diagonal, right-diagonal, rising, hook and turning, as shown in Figure 1. These basic strokes determine the structure of a character as the outline of a rectangle or a square rather than one of a curve in some other alphabetical characters. The basic rules of stroke order in writing Chinese characters are first horizontal, then vertical; first left-diagonal, then right-diagonal; from top to bottom, from left to right; first outside, then inside; finish inside, then close; first middle, then two sides. Figure 2 gives examples for illustrating these rules.

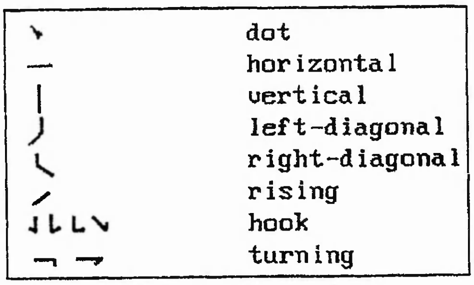


Figure 1. Basic Strokes

十 → 一 丨	horizontal → vertical
八 → 丿 ㇏	left-diagonal → right-diagonal
忠 → 中 心	top → bottom
明 → 日 月	left → right
闷 → 门 心	outside → inside
国 → 冂 玉 一	outside → inside → close
小 → 亅 八	middle → two sides

Figure 2. Basic Rules

Early Chinese characters were mainly symbols and *pictographs* that could also represent some abstract concepts of daily life; later versions expressed more complex ideas and concepts. Symbols and *pictographs* were extended or combined to form *ideographs*. Extensions of multiple meanings were sometimes inherent in these characters themselves and required no further addition. These *ideographs* formed some 80% of the total 3,500 Chinese characters used in daily life. This historical development is simply shown in Figure 3.

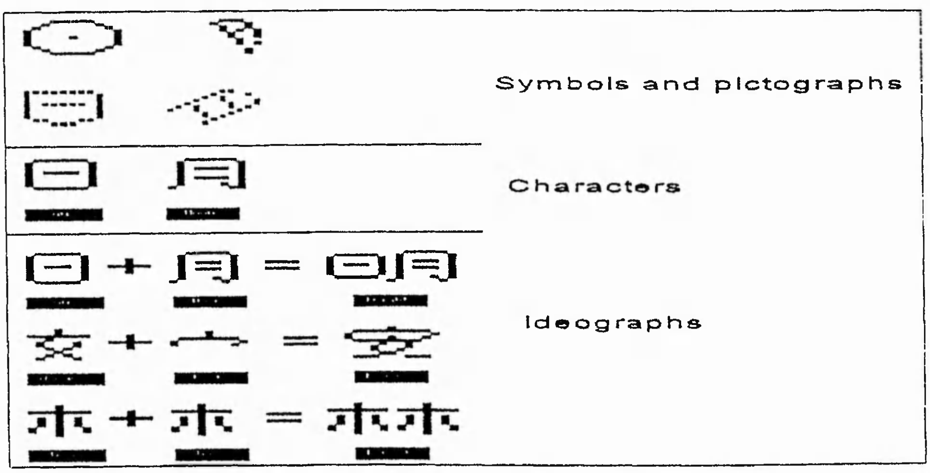


Figure 3. Development of Chinese Characters

Most *ideographs* are made up of two components: (i) a radical symbol (the Chinese call these '*signifiers*' or '*common heads*') which indicates the classification of the character; and (ii) a 'phonetic' symbol to partially aid in the pronunciation of a character [Scurfield, 1991]. Old Chinese dictionaries list 214 radicals. Modern ones have reduced this to 189. If the forms in combination and/or full characters are listed separately, this can increase the number to around 250. Many of these radicals are originally *pictographs*, but some of them have been simplified so much that the original picture has almost been lost.

2.2 The Cang-Jie method

The Cang-Jie method introduced by Bang-Fu Zu in the 1980's [Liu, 1993] is a method that illustrates partitioning a Chinese character into its radicals and typefaces. In the method, primitive components of Chinese characters are defined as radicals and typefaces rather than strokes, although a radical or a typeface is made up of strokes. The method also classifies radicals and typefaces into 26 different categories (24 radical's categories and 2 special case categories) in terms of their logical meanings and shape structures. In this method, a typeface is specified as a simple combination of one or two, even more basic strokes. A radical inherits its basic concepts from the historical development of Chinese characters, i.e. a radical may keep its individual character meaning and occupy a completely independent position in a character. Figure 4 shows an example of a radical and a typeface in this method.

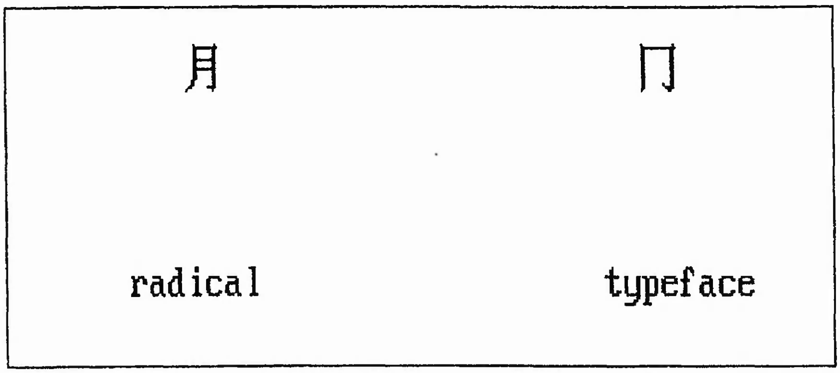


Figure 4. An Example of A Radical and A Typeface

Even though it includes some technical rules for partitioning a character, the whole Cang-Jie method is based on human operations, i.e. identifying a radical or a typeface and applying partitioning rules depending on human manual analysis and selection. In fact, the method is used for training a typist to input a Chinese character from a keyboard at its first usage.

2.3 The combined-radicals method

The main feature of this method is to apply the partitioning rules of the Cang-Jie method to computer aided Chinese character recognition [Al-Dabass, Ren and Su, 1995], so that the processing of recognising a character can be simplified to one of recognising a radical or a typeface [Ren, Su and Al-Dabass, 1995]. Figure 5 shows a five levels structure: stroke, typeface, radical, character and phrase. Initially, the application of the method is limited to extracting a radical or a combination of a radical and a typeface from a character.

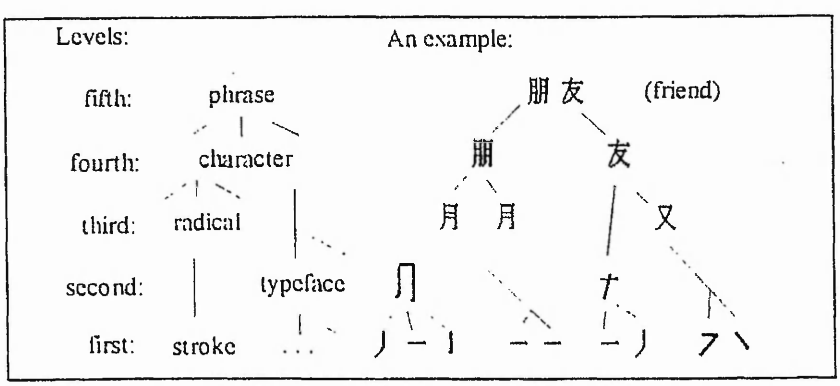


Figure 5. Five Levels Structure of a Chinese Character

3. APPLICATION OF THE SYNTACTIC METHOD

The syntactic method used in the system defines a number of specifications that describe how to partition a Chinese character into its radicals and typefaces. These specifications or rules will determine the shape, position and order of a radical in a character. The rules are designated as static and dynamic. The static rules mainly deal with general partitioning specifications for defining the shape of a radical or a typeface. The dynamic rules take the position and the order of a partitioned radical into account, while the partitioning process is being carried out.

According to the definition of the five levels structure in stroke, typeface, radical, character and phase, shown in Figure 5, a radical is a two-dimensional picture based on a structural relationship between strokes and typefaces. The description of a radical can be grammatically defined as a fuzzy mathematical set with five attributes, which is expressed as follows.

$$R = \{T, S, RS, P, N\}$$

Where:

R stands for a fuzzy set of a radical attributes.

T represents a fuzzy attribute's subset of typefaces that may consist of the radical.

S is a fuzzy attribute's subset of strokes that contain parts, which cannot be presented by any typeface, of the radical.

RS expresses a structural relationship between these typefaces or between a typeface and a stroke.

P stands for a fuzzy subset of the typefaces and strokes position in a radical.

N is an attribute's subset of categories to which the radical, typefaces and strokes belong.

An example of a radical description is shown in Figure 6.

Following the structure in Figure 5, a character is built up by a structural relationship between radicals and typefaces. Its description can be defined as such a fuzzy mathematical set:

$$C = \{R, T, RS, P, O, D\}$$

Where:

C is a fuzzy set of a character attribute.

R is a fuzzy attribute's subset of radicals which consist of the character.

T represents a fuzzy attribute's subset of typefaces that are parts of the character, but cannot be presented by any radical.

RS expresses a structural relationship between these radicals and typefaces.

P stands for a fuzzy subset of the radicals and typefaces position in a character.

O is a fuzzy subset which describes an omitted part in a character.

D is a fuzzy attribute's subset of parts which are difficult to be described in the character.

Figure 7 gives an example for illustrating a fuzzy set of a character.

```
  月 (moon)

R = {T, S, RS, P, N}
T = {t1}
    t1 = {冂}
S = {s1, s2}
    s1 = {—}; s2 = {—}
RS= {rs1, rs2, rs3, rs4, rs5}
    rs1 = {t1 connects s1},  rs2 = {t1 connects s2};
    rs3 = {t1 is around s1}, rs4 = {t1 is around s2};
    rs5 = {s1 and s2 are independent}
P = {p1, p2}
    p1 = {s1 and s2 are in the inside of t1};
    p2 = {s1 is on the top of s2}
N = {n1, n2, n3, n4}
    n1 = {R ∈ 2AA ('moon category')};
    n2 = {t1 ∈ 21B ('the first typeface group in 'moon' category)};
    n3 = {s1 ∈ 002 (the second one in the strokes set)};
    n4 = {s2 ∈ 002 (the second one in the strokes set)}
```

Figure 6. A Radical Fuzzy Set

```
  朋 (friend)

C = {R, RS, P}
R = {r1, r2}
    r1 = {月}; r2 = {月}
RS = {rs1}
    rs1 = {r1 and r2 are independent}
P = {p1, p2}
    p1 = {r1 is on the left of r2};
    p2 = {r1 and r2 are side by side}
```

Figure 7. A Character Fuzzy Set

3.1 Definition of static rules

The definition of static rules is denoted by the following assumptions for an input character and a number of basic specifications for determining the shape of a radical.

Assumption:
1) Input characters are in the form of a formal printing character.
2) The strokes have nearly constant width;
3) The input image is either black (binary 1) or white (binary -1).

Definition:

The whole input image P, which denotes the set of the image, and p(x, y), which is a pixel in an X*Y resolution image, are satisfied by the following expression:
$P = \{p(x, y)|0<x\leq X, 0<y\leq Y\}$ and P_b and P_w denote sets of black and white pixels respectively, where: $P_b \cap P_w = \varnothing$ and $P_b \cup P_w = P$.

Neighbour definition: $p(x_1, y_1)$ and $p(x_2, y_2)$ are neighbours if the pixel $p(x_2, y_2)$ position is next to any one of eight directions from the pixel $p(x_1,y_1)$, shown in Figure 8, and $p(x_1, y_1)$ and $p(x_2, y_2) \in P_b$.

Joint definition: P_{b1} and P_{b2} are joint sets if a pixel p(x, y) is a common pixel of P_{b1} and P_{b2}, where: P_{b1} and $P_{b2} \in P_b$. An illustration for the definition is shown in Figure 9.

Touch definition: P_{b1} and P_{b2} are touch sets if a pixel $p(x_1, y_1) \in P_{b1}$ is a neighbour of a pixel $p(x_2, y_2) \in P_{b2}$, P_{b1} and $P_{b2} \in P_b$ and $P_{b1} \cap P_{b2} = \varnothing$. An explanation for the definition is shown in Figure 10.

Continuous definition: P_{b1} and P_{b2} are continuous if P_{b1} and P_{b2} are joint or touch set, where: P_{b1} and $P_{b2} \in P_b$.

Basic strokes set definition: eight basic strokes: dot, horizontal, vertical, left-diagonal, right-diagonal, rising, hook and turning, shown in Figure 1, are defined as a special set which can be accepted by the system.

Rectangle or Square definition: a combination of P_{b1} and P_{b2} is assumed a radical if the width and the height of a shape in the combination are in the block of a rectangle or square. Here, P_{b1} and P_{b2} are continuous or $\in$ basic strokes set.

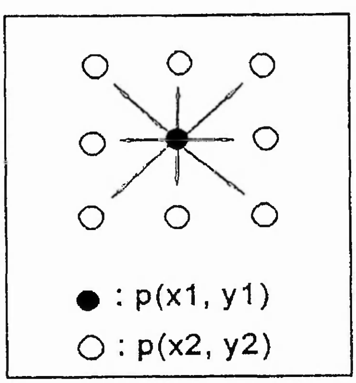


Figure 8. Neighbour Definition

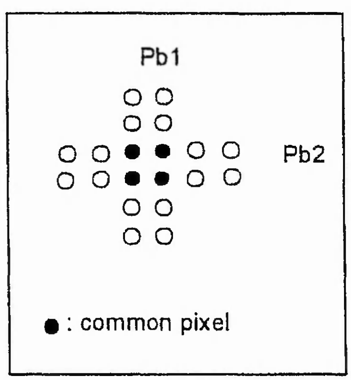


Figure 9. Joint Definition

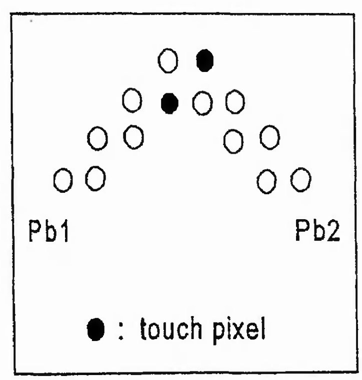


Figure 10. Touch Definition

Rules:

Line partitioning: a character should be partitioned on a line between a radical and a typeface, shown in Figure 11, not in a corner or a hook position.

No common part partitioning: partitioned radicals and typefaces are required without sharing any common part of a character.

Partitioning number: the total number of radicals and typefaces partitioned from a character is lower than five, for an indivisible character, and lower than six, for others.

Special radicals partitioning: several radicals made up of independent parts, listed in Figure 12, are dealt with by special processing of the partitioning rules.

Pseudo partitioning: partitioning, which satisfies the definition of a radical's shape explained above, but without the feedback check, is called pseudo partitioning.

Real partitioning: partitioning after the feedback check is called real partitioning. Results after real partitioning will be passed to further processing.

Check:

Strokes Check: P_{b1} has to belong to the strokes set if P_{b1} is P_{b2} continuous set, where: P_{b1} and $P_{b2} \in P_b$.

Rectangle check: the width and height of a continuous set P_{b1}, which satisfies the rectangle definition and $P_{b1} \in P_b$, should be in a block of a rectangle or square. Figure 13 illustrates the check.

Complete radical's shape check: the shape of a complete radical should be kept on a continuous line except the special radicals case, shown in Figure 13.

Feedback check: real partitioning will be required if the shape of a pseudo partitioned radical can be matched to a formal radical sample stored in a database of the system. Otherwise, pseudo partitioning has failed and needs to be done in a different way.

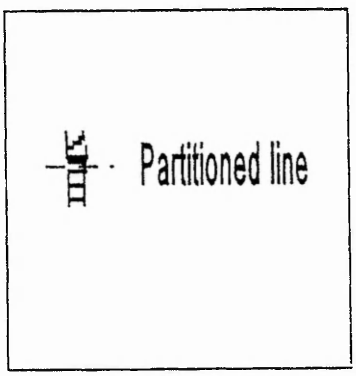


Figure 11. Line Partitioning

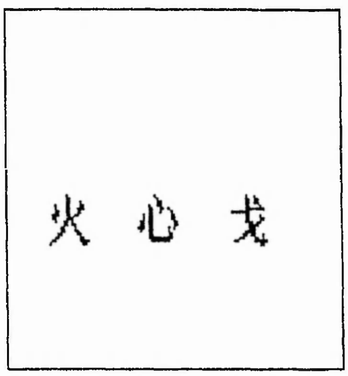


Figure 12. Radicals with Several Independent Parts

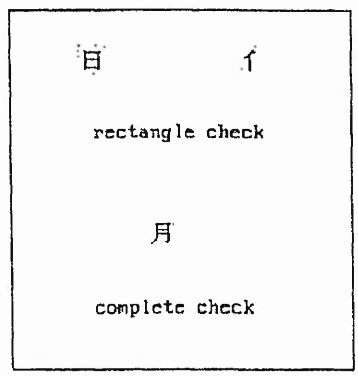


Figure 13. Rectangle and Complete Check

3.2 Definition of dynamic rules

The definition of dynamic rules takes account of the position and order of a partitioned radical. The rules are organised by the basic rules of stroke order in writing Chinese characters, shown in Figure 2, and concepts used in the Cang-Jie method: from outside to inside; from top to bottom; and from left to right. Further rules will be developed on these dynamic rules, such as a combination of these rules.

Basic dynamic rules are classified as the following groups. Each group includes two important parts: the position and order of a radical. The group name indicates the position of a radical in a character. The name order denotes the order of a radical taken from a character. For instant, the group name: outside and inside signifies two radicals which will be partitioned from a character; one on the outside of another radical will be partitioned first, the other will be taken out secondly. A partitioned character is processed following the order of groups until it is fully partitioned.

Groups:
(1) Outside and inside
Two parts are assumed to be partitioned from a character. One on the outside of another will be taken out first. An example is shown in Figure 14.
(2) Inside and outside
Two parts will be segmented. The inside one is taken out first. An example is illustrated in Figure 14.
(3) Top, middle and bottom
Three parts are being partitioned. One in the top of a character has to be segmented first; the second one is taken out from the middle; the third one is on the bottom.
(4) Top and bottom
Two parts are required to be partitioned from a character. The first one is on the top, and the second one on the bottom.
(5) Left, middle and right
Three parts, on the left, middle and right of a character separably, need to be partitioned.
(6) Left and right
Two parts, first left then right, are partitioned.
(7) No partitioning
A part, as an indivisible character, is confirmed to be a radical or a typeface without doing any partitioning.
(8) Omitted part
If the partitioning number is > 5 (partitioning limitation), the whole part will be omitted. An example is shown in Figure 14.
(9) Difficult part
A part may be too difficult to partition from a character. An example is illustrated in Figure 14.

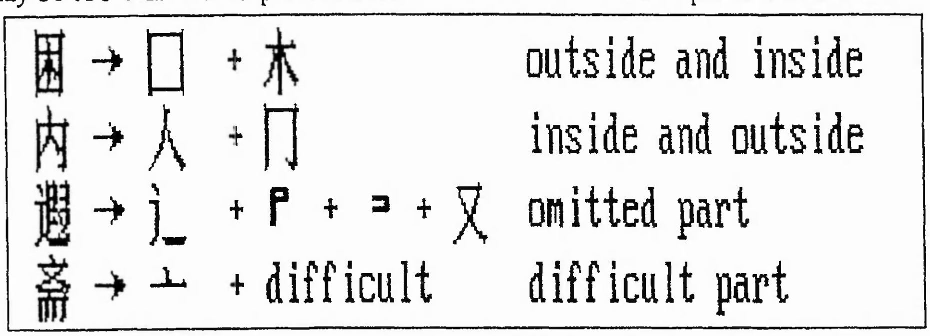


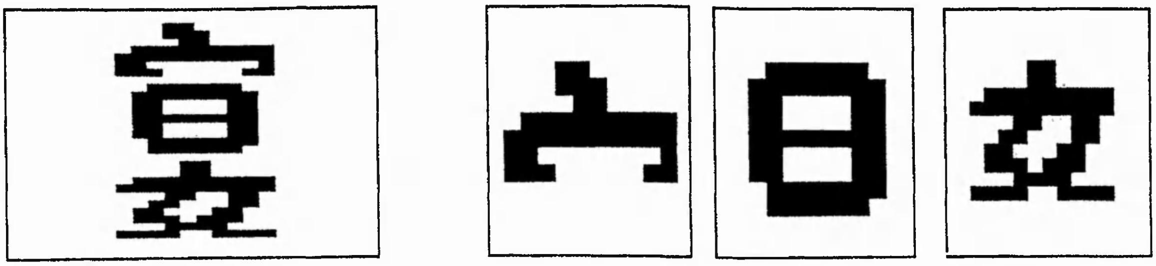
Figure 14. Several Examples for Dynamic Rules

4. EXPERIENTIAL RESULTS AND DISCUSSION

To investigate the partitioning capability of the system described above, some experiments have been conducted. The system is implemented in the C language on a SUN workstation with X View functions. The simulated characters were input in a 30*30 bitmap format. A character is represented by pixels, 1 and -1, of a 30*30 binary matrix. Considering the difficulty of extracting a typeface from a character, the system at present is only applied to partition a radical or a combination of a radical and a typeface. Partial static rules and five groups of dynamic rules have been applied to the system and the partial test results are shown in the following figures.

4.1 Partial test examples

(1) Top, middle and bottom
The character '*banquet*' consists of three radicals and is partitioned into three parts as shown in Figure 15.

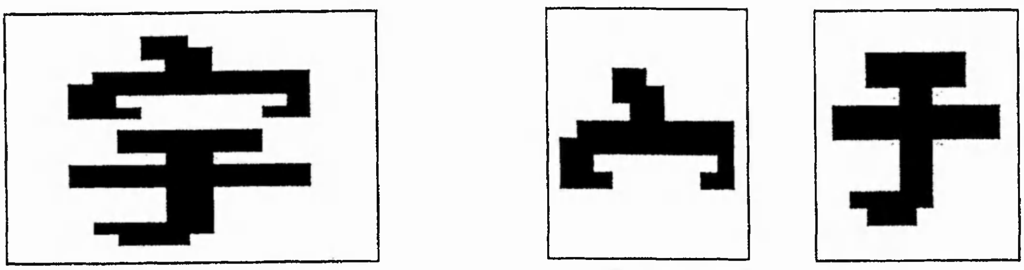


(a) An Input Character (b) Partitioned Results

Figure 15. A Partitioning Example of Top, Middle and Bottom

(2) Top and bottom

The input character '*universe*' is partitioned into two parts as shown in Figure 16.

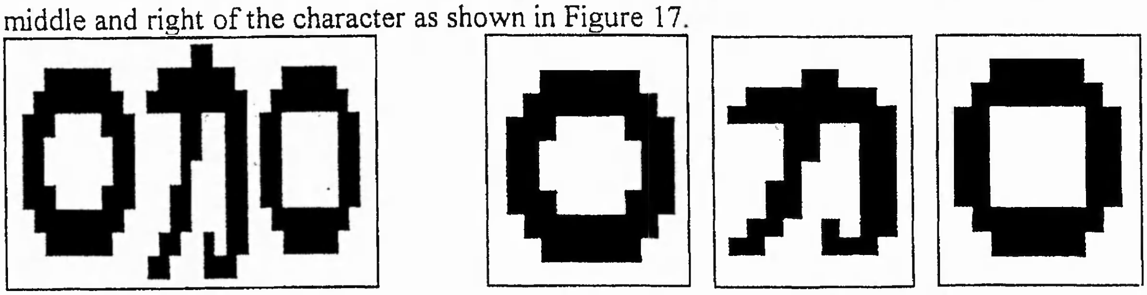


(a) An Input Character (b) Partitioned Results

Figure 16. A Partitioning Example of Top and Bottom

(3) Left, middle and right

Three radicals make up the character '*coffee*' and are separated into three parts from the left, middle and right of the character as shown in Figure 17.

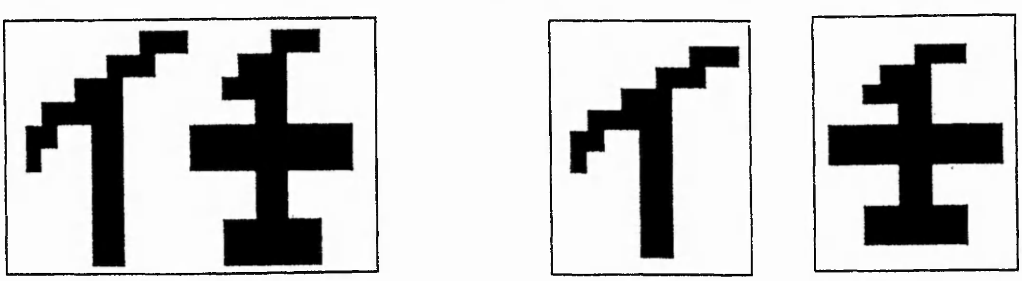


(a) An Input Character (b) Partitioned Results

Figure 17. An Example of Left, Middle and Right Partitioning

(4) Left and right

The character '*ren*' is divided into the left and right parts, illustrated in Figure 18.

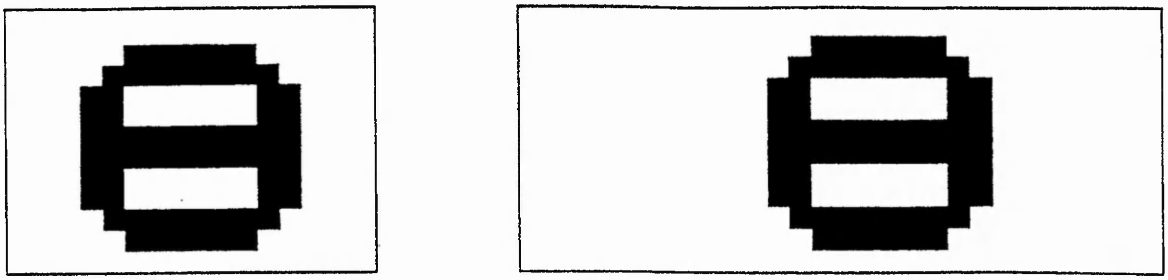


(a) An Input Character (b) Partitioned Results

Figure 18. A Partitioning Example of Left and Right

(5) No partitioning

The character '*sun*' is a radical and cannot be segmented at all.



(a) An Input Character (b) Partitioned Results

Figure 19. An Example of No Partitioning

4.2 Discussion

Form the above test results, it is clear that the difficulty of partitioning still centres on determining the shape of a radical. It will be more difficult when the special radicals are being carried out. The problem of variable size of the radical's shape has not been dealt with by the system so far. Partitioning a typeface from a character or from a combination of a radical and a typeface will be another tough target to be achieved.

5. CONCLUSIONS

For simplifying Chinese character recognition, a Radical Partitioning system using the syntactic, Chang-Ji and combined-radicals methods has the advantages of breaking down the structure of a Chinese character into its radicals and typefaces, to extract radicals from a character. A number of static and dynamic rules were defined and partially tested. Further research is needed to enhance the static and dynamic rules, apply these rules to a typeface and extend the application of rules to more sophisticated Chinese characters.

REFERENCES

Al-Dabass, D.; Ren, M. and Su, D. (1995), *An Associative Memory Artificial Neural Network System with a Combined-Radicals Method for Chinese Character Recognition*, ICONIP'95-Beijing: Proceedings of International Conference on Neural Network Information Processing, Beijing, China, Vol.2 of 2, pp.857-860.

Bunke, H. and Sanfeliu, A. (1986), *Introduction to the Special Issue: Advances in Syntactic Pattern Recognition*, Pattern Recognition, Vol.19, No.4, pp.249-254.

Hamilton, R. J.; Pringle, R. D. and Grant, P. M. (1991), *Syntactic Pattern Recognition of Sampled Data Signals*, Electronics Letters, Vol.27, Iss.24, pp.2213-2214.

Liu, Y. M. (1993), Introduction of Cang-Jie Coding Method, Run Lin Ltd. (Chinese)

Liu, Y. J. and Tai, J. W. (1988), *The Theory and Practice of On-line Chinese Character Recognition*, Journal of Chinese Information Processing, Vol.2, No.4, pp.1-13. (Chinese)

Ren, M.; Su, D. and Al-Dabass, D. (1995), *An Associative Memory Artificial Neural Network System*, ECAC'95: Proceedings of European Chinese Automation Conference, London, UK, pp.91-96.

Ren, M. (1994), *An Associative Artificial Neural Network System with a Combined-Radical Structure for Chinese Character Recognition*, internal report, Department of Computing, The Nottingham Trent University, UK.

Scurfield, E. (1991), Chinese, Hodder and Stoughton, UK.

Using Associative Memory Neural Network with Sub-Nets Structure for Classification of Radicals in Chinese Characters

Manling Ren*, David Al-Dabass* and Daizhong Su**

* Department of Computing
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel.: +44 115 941 8418 Ext. 2761
Fax: +44 115 948 6518
E-mail: mar@doc.ntu.ac.uk
dad@doc.ntu.ac.uk

** Department of Mechanical and Manufacturing Engineering
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel.: +44 115 941 8418 Ext. 2306
Fax: +44 115 948 6506
E-mail: daizhong.su@ntu.ac.uk

ABSTRACT

This paper presents architecture of an associative memory neural network with sub-nets structure for classifying radicals in Chinese characters. The architecture is based on representing the topological structure of these characters by a three-layer hierarchy. Input radicals are recognised and translated from a 2-D pattern to single letters in the processing. Breaking the network into sub-nets reduces its intra connectivity and forms another novel technique for classification, which together with recognition and translation of radicals is implemented by the Recognition system. Test results show that the system is effective and reasonable.

Key words: Associative memory neural sub-networks, classification of radicals, recognition of 2-D patterns.

1. INTRODUCTION

As part of the research in computer recognition of Chinese characters, a new representation scheme proposed by the authors in a previous paper (1) is being developed by using the three-layer hierarchy: character-radical-stroke, for deconstructing the topological structure of a character into radicals, and combining strokes into a radical. Although the representation method has advantages in reducing vocabulary of Chinese characters offered for recognition, radicals extracted from a character are still difficult to recognise. The difficulty mainly lies in that (a) a radical is a 2-D pattern; (b) it is not easy to classify radicals; (c) restructuring the radicals back to a character as the final result will be a problem for recognition of another new 2-D pattern.

Currently, there are some methods for a human to search for or classify Chinese characters using computer. For instance, using pronunciation, called *Pinyin*, of a character can get the character as shown in Figure 1 (a). Observing shape in the four corners of a character and translating them into codes, termed as Four-Corner, is another method for determining a character, shown in Figure 1 (b). Breaking down a character into several parts according to some rules, identifying these parts by their classification, and combining category codes of classification into a chain code for obtaining a character are called as *Cang-Jie* method shown in Figure 1 (c).

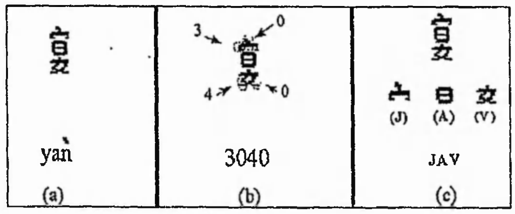


Figure 1. Three methods for classification of Chinese characters

Although all of these methods depend on human analysis and intelligence, classification of characters could be arranged in a certain order. In the *Pinyin* method, characters are classified into 26 categories by following vowels and consonants in pronunciation: 23 are in common use and another 3 rarely used. In the Four-Corner method, characters are classified by 10 figures from 0 to 9. The *Cang-Jie* method classifies 108 different radicals into 26 categories by using 26 English letters.

A common feature of the three methods is to use a chain code (consisting of vowels and consonants in the *Pinyin* method, 4 numbers in the Four-Corner method and less than five letters in the *Cang-Jie* method) in a certain format to stand for a 2-D character. Comparative analysis of the 3 methods shows: (i) the *Pinyin* method: no relation connects the shape of a character to classification of its vowels or consonants; (ii) the Four-Corner method: a fuzzy definition, depends on near similarity of shape and classification; and (iii) the *Cang-Jie* method has more merits in that the classification is formed by shape and identification of a character.

In this paper, a 'Recognition' system with an architecture using associative memory neural network with sub-nets structure is applied to carry out classification of radicals deconstructed from a character to output a letter as the recognised result of a radical. The technique of sub-networks is introduced to reduce intra connectivity of the whole network, speed up processing time, and to use shared weights where possible. As part of a Chinese Character Recognition System (CCRS) which carries out the whole recognition process of a Chinese character, the Recognition system deals only with (a) classification of radicals; (b) recognition of a radical; and (c) translation of a radicals from a 2-D pattern into a single letter. The processing diagram of the system is shown in Figure 2.

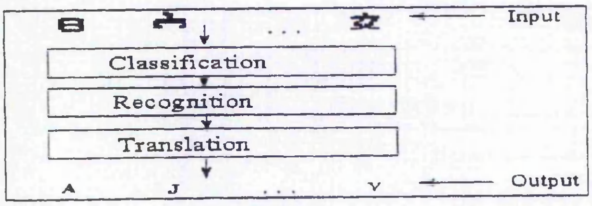


Figure 2. Processing of Recognition system

The paper is organised as follows. The foundation of radical classification is described in Section 2. The recognition of radicals with the architecture of associative memory neural network is presented in Section 3. Translation of radicals from a 2-D pattern to a single letter is given in Section 4. Experimental results produced by the Recognition system with classification, recognition and translation of radicals together with discussions of several special cases are presented in Section 5. Conclusions and suggestions for further development are given in Section 6.

2. CLASSIFICATION OF RADICALS

The classification of radicals in the system is developed through the application of the *Cang-Jie* method. Compared to other methods, the *Cang-Jie* method classifies radicals according to their shape rather than meaning, which gives an advantage when computer deals with the shape structure of radicals. Disadvantages of the method include manual operation and complete dependence on human observation. Based on its good classification of radicals, the system has employed the knowledge of human expert analysis to computer processing.

2.1 Cang-Jie Method

The *Cang-Jie* method, contributed by Bang-Fu Zu (2), is a method of classifying radicals into related categories using human observation and operation. Usage of the method is mainly for input of Chinese characters from a normal computer keyboard. Based on analysing the structure of 30,000 Chinese characters, the method has classified the characters into 26 different categories (24 for radicals and 2 for special cases).

2.2 Standard of Classification

In the 24 radical categories, each category includes a standard radical and some other radicals and token-radicals that are similar to or simplification of the standard radical, in terms of either their shape structure or logical meanings. The total number of radicals and token-radicals in these categories is 108. The 2 special case categories are designed for difficult characters and for making a new character. All categories are numbered by following the alphabetic letters, A, B, C, ... Z, where X and Z stand for difficult and new characters in the 2 special case categories, respectively.

The 24 radical categories are arranged in four groups: philosophy, stroke combination, physical symbol, and shape similarity. The **philosophy** group includes 7 categories, termed *sun, moon, metal, wood, water, fire* and *soil*. Radicals that belong to this group are derived from abstract concepts of basic requirements of human lives in ancient China. The **stroke combination** group has 7 categories, which come from the combination of basic strokes: *left-diagonal, dot, cross, x connection, vertical, horizontal* and *hook* or *turning*. The **physical symbol** group consists of 4 categories, which stand for an abstract expression of such symbols from human body: *person, heart, hand* and *mouth*. The **shape similarity** group has 6 categories, called *flanking open, abreast balance, u shape, twisting shape, square* and *y shape*.

2.3 Development of Classification of Radicals

Although some radicals from characters are quite difficult to match into a category even by trained experts, these groups and categories have special advantages for illustrating topological structure and traditional culture customs. Assigning the 26 English letters A to Z to these categories allows us to translate a radical from its pictorial to a letter format, i.e. from a 2-D structure to a single letter. Therefore, both the method of classifying radicals and available labels of these categories are applied to the Recognition system.

Three basic principles of determining categories in the system are developed: (a) a member in a category should have the physical properties of the category and major features from the group that the category belongs to; (b) each member in a category may be a radical or token-radical or some combination of basic strokes; and (c) combinations are allowed between a token-radical and basic strokes to form a new integrated radical. The principles have some benefits in transforming knowledge of a radical identified abstractly by human analysis into a problem of recognising different shapes by computer.

3. A RECOGNITION SYSTEM USING NEURAL NETWOR ARCHITECTURE

Basically, the Recognition system is a four-layer neural network, composed of multi sub-nets and based on architecture of associative memory function (3). Sub-nets are developed to reduce intra connectivity of the network and deal with radicals in a category. The associative memory function offers an important advantage of recalling a stored pattern from its partial or noisy input (4). Figure 3 shows the architecture of the Recognition system using associative memory neural network with sub-nets structure.

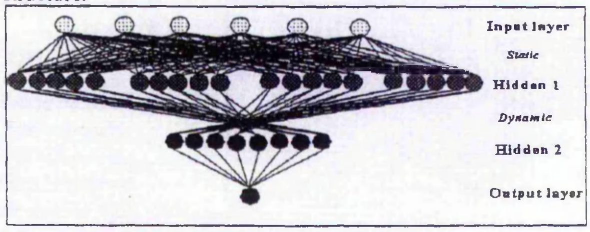


Figure 3. Architecture of the Recognition system using associative memory neural network with sub-nets structure

3.1 Architecture of Neural Network

There are four layers in the Recognition system: input, hidden-1, hidden-2 and output. Hidden-1 layer consists of multi sub-nets where each sub-net deals with radicals in a category. The number of neurones in each sub-net is decided by the learning patterns in a category. The connectivity from input to hidden-1 layer is static. Neurones in hidden-2 layer are created by the recognised results from hidden-1 layer. The connectivity between the two hidden layers is dynamic. The design of hidden-2 layer with a dynamic structure is used to further enhance convergence on global minimum of the associative algorithms.

The network has two phases for its implementation: learning and training. In the learning phase, radicals are classified into categories. Each category is formed as a sub-network. The 26 different sub-networks are composed of a whole neural network with associative memory function, i.e. the Recognition system. The major task in the learning phase is to learn formal radicals and to form inter connectivity for training an input pattern.

In the training phase, sub-nets in the hidden-1 layer are trained to converge to local minima. The hidden-2 layer is generalised by re-learning these patterns of local minima. Eventually, the global minimum will be converged to in the output layer.

The purpose of breaking the network into sub-nets is to reduce the connectivity of the whole network and to use shared weights. When the network is connected as a whole, its inter connectivity is low but intra connectivity is high (5). The intra connectivity could be reduced while the architecture of sub-nets has been used.

Weights in the network can be shared in both of feature space and time, i.e. weights that are associated to different (e.g. translated) input feature may be shared, and weights that are associated to different times may be shared. In the current Recognition system, shared weights in feature space are considered, but not in time.

3.2 Algorithms of Associative Memory Neural Network

Neural networks are developed on the basis of mathematical foundation, inherent parallelism, knowledge store, fault tolerance, and adaptability (6). Some mathematical equations affect inputs, memory, recall, determination of energy levels, convergence, and stability. The mathematical formula for the associative memory function is established on the construction of an energy equation (7):

$$E = -\sum_i \sum_j X_i W_{ij} Y_j + \sum_i \theta_i X_i + \sum_j \varphi_j Y_j \qquad \cdots\cdots \quad (3.1)$$

The algorithm contains two phases: learning and training. In the learning phase, the associative memory function is used to form the connectivity matrix W for training a set of input patterns Xi (u) and output patterns Yi (u), where u=1,2, ...M; i = 1,2, ... N, the weight W (i,j) is determined by the Hebbian rules:

$$W(i,j) = \begin{cases} \sum_u X_i(u) Y_j(u) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \qquad \cdots\cdots \quad (3.2)$$

In the training phase, the algorithm aids convergence because its value in equation (3.1) either is reduced or remains constant during the recall procedure (8), if the following conditions are satisfied:

$$Y_j^{n+1} = \begin{cases} 1 & \sum_i W_{ij} X_i^n - \varphi_j > 0 \\ Y_j^n & \sum_i W_{ij} X_i^n - \varphi_j = 0 \\ -1 & \sum_i W_{ij} X_i^n - \varphi_j < 0 \end{cases}$$
$$X_i^{n+1} = \begin{cases} 1 & \sum_j W_{ij} Y_j^n - \theta_i > 0 \\ X_i^n & \sum_j W_{ij} Y_j^n - \theta_i = 0 \\ -1 & \sum_j W_{ij} Y_j^n - \theta_i < 0 \end{cases} \qquad \cdots\cdots \quad (3.3)$$

When a portion of the original pattern is used as a retrieval cue, the algorithm is denoted as auto-associative memory. When the desired output is different from the input, the algorithm is called hetero-associative. The static method requires that the network is trained in advance with specific patterns, and retrieves them from portions or distorted inputs. When the learning and retrieval are embedded in the training process randomly, it is defined as a 'dynamic' method.

4. TRANSLATION OF RADICALS FROM 2-D TO SINGLE LETTER

In the Recognition system, a radical as a 2-D pattern is dealt with in input, hidden-1 layer and hidden-2 layer. A radical is translated from its 2-D pattern to a single letter while the radical is recognised as the final result in the output layer. In its 2-D format, a radical is kept in a pictorial format, as shown in Figure 4 (a). After translation, an English letter, as shown in Figure 4 (b) replaces the radical and stands for a category label to which the radical belongs.

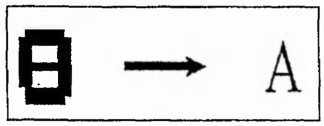


Figure 4. Formats of radicals

Analysing the results of the translation, it is observed that shape, size, and real meanings of radicals have been lost in its single letter. Advantages of using a single letter substitute are (a) restructuring several radicals to a character without considering its shape, size and position; and (b) confirming a character in a database using its chain code, i.e. a combination of several letters standing for a character, rather than a character in bitmap or other different fonts.

Its disadvantage is the fact that a letter is an unrelated symbol of a category set rather than its member. If a mis-recognised case happens, the only solution for recovering the symbol is to refer to its neighbours in its chain code for correction, or by a guessing process.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to show the effectiveness of the Recognition system based on associative memory neural network for classification of radicals, some experiments have been conducted on a Sun workstation with X View function and the C language to program algorithms of the network. In the processing, the experimentation was focused on different classification, recognition and modification of network weights.

5.1 Classification of Radicals

According to standards of classification, 120 test radicals are divided into the 24 categories as shown in Figure 5.

Within these categories, a radical with a tag represents a combination of radicals or token-radicals, which are independent in different categories in the *Cang-Jie* method. The tag used refers to a database of Chinese characters instead of building up a new one.

Formal radicals, as shown in Figure 6, in each category have been learnt and formed connectivity of sub-networks, as shown in Figure 7, of the network.

5.2 Recognition of Radicals

112 out of 120 radicals have been recognised by using the structure of the network, where 74 radicals were different from each other. Figure 8 gives the recognition rate of these radicals (both indicated by the y-axis) in different categories indicated by the x-axis). Figure 9 shows the recognition frequency of the radicals in the categories.

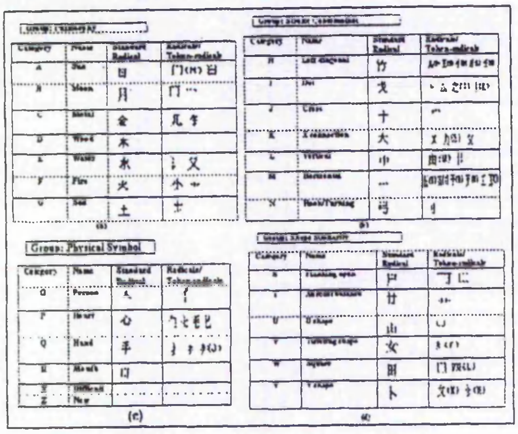


Figure 5. Classification of test radicals

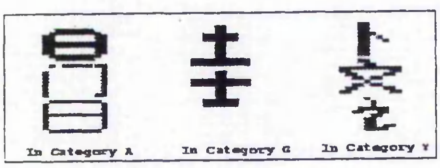


Figure 6. Some learning radicals

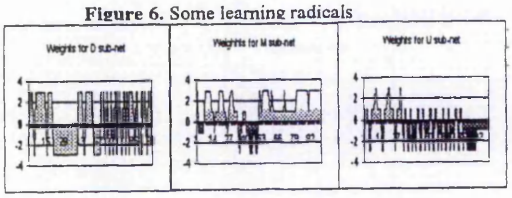


Figure 7. Weights for some sub-nets

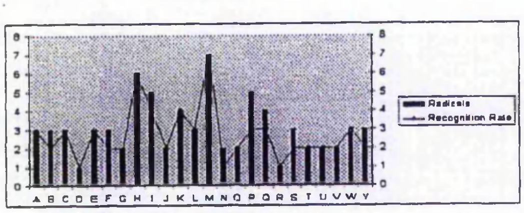


Figure 8. Recognition rate of radicals in different categories

5.3 Discussion

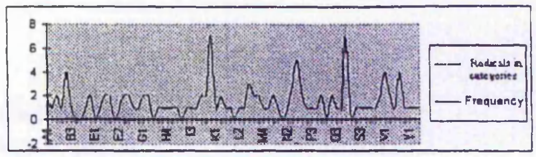


Figure 9. Recognition frequency of the radicals in the categories

The associative memory algorithm has the advantage of quicker convergence speed and the capability of recognising the error patterns. In the Recognition system, dividing the whole network into sub-nets has solved the limitation of associative memory

The ambiguity of recognising patterns is basically caused by converging local minimum, especially when two patterns had the same reliability rate.

6. CONCLUSIONS

In this paper, the architecture of associative memory neural network with sub-nets structure for classification of radicals was introduced. Based on the *Cang-Jie* method, the classification of radicals has been specially divided into 26 categories. The recognition and translation of radicals have been carried out by the Recognition system. The modification of the associative memory function in several stages has enhanced the recognition rate of the system. The ambiguity of recognising patterns, i.e. mis-recognition, still can happen in some cases, but this is unrelated to the advantage of faster convergence speed. Further development will focus on improvement of classification of radicals, dealing with mis-recognition cases and reducing the processing time.

7. REFERENCES

1. Ren, M., Al-Dabass, D. and Su, D. A Three-Layer Hierarchy for Representing Chinese Characters, Research and Development in Expert Systems XIII: Proceedings of Expert Systems 96, the Sixteenth Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems, 1996, pp.137-146.

2. Liu, Y.M. Introduction of Cang-Jie Coding Method, 1993, Run Lin Ltd. (Chinese)

3. Al-Dabass, D., Ren, M. and Su, D. An Associative Memory Artificial Neural Network System with a Combined-Radicals Method for Chinese Character Recognition, ICONIP'95-Beijing: Proceedings of International Conference on Neural Network Information Processing, Vol.2 of 2, Beijing, China, 1995, pp.857-860.

4. Freeman, J. and Skapura, D. Neural Networks: Algorithms, Applications, and Programming Techniques, 1992, Addison-Wesley Publishing Company, pp.127-168.

5. Bengio, Y. A Connectionist Approach to Speech Recognition, Advances in Pattern Recognition Systems Using Neural Network Technologies, 1993, edited by I. Guyon and P. S. P. Wang, World Scientific, pp.3-24.

6. Dayhoff, J. Neural Network Architectures: An Introduction, 1990, Van Nostrand Reinhold, pp.37-57.

7. Hopfield, J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proc. Ntl. Acad. Sci., 1982, Vol.79, pp.2554-25558.

8. Wang, T. Improving Recall in Associative Memories by Dynamic Threshold, Neural Networks, 1994, Vol.7, No.9, pp.1379-1385.

A Fuzzy/Possibilistic-Reasoning AM-Architecture for Recognising Radicals in Chinese Characters

Manling Ren*, David Al-Dabass* and Daizhong Su**

* Department of Computing
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel: +44 (0)115 941 8418 Ext. 2761
Fax: +44 (0)115 948 6518
E-mail: mar@doc.ntu.ac.uk
dad@doc.ntu.ac.uk

** Department of Mechanical Engineering
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel: +44 (0)115 941 8418 Ext. 2306
Fax: +44 (0)115 948 6506
E-mail: mec3sud@ntu.ac.uk

ABSTRACT

This paper presents a fuzzy neural network system with possibilistic reasoning rules and the associative memory (AM) neural network architecture for Chinese character recognition. Using fuzzy possibilistic reasoning rules, the system extracts features of the topological structure of characters and constitutes a subdivision representation based on their physical shape and classified according to their logical meanings and pattern structure. The associative memory neural network is applied for recognising radicals subdivided from a character after implementing fuzzy possibilistic rules. Several special cases and application limitations of the system are introduced and discussed. Test results show that the system is effective and reasonable.

Key words: Associative-memory neural network, Chinese character recognition, fuzzy possibilistic reasoning, inference rules, subdivision.

1. INTRODUCTION

As part of the research in computer recognition of pictorial language text (e.g. Chinese characters), a new method proposed by the authors in a previous paper (1) is being developed by using radicals as part of the recognition process. Radicals are common elements (or components) which appear in many characters. Compared to other methods of Chinese character recognition, for instance, strokes or words recognition, this method has significant advantages of (i) emulating the human learning process of these characters, (ii) systematising the internal topological structure of a character, and (iii) reducing the vocabulary of characters to be learnt by computer.

Due to the characteristics that a Chinese character is a two-dimensional picture and varies its structure in different ways, difficulties of applying the method include (i) determining the shape, position and size of a radical in a character, which could be seen as the vagueness of a radical, and (ii) recognising the radical subdivided from a character, which focuses the one-to-many, i.e. the ambiguity of a radical, relationship between the radical and others.

A fuzzy neural network system using possibilistic reasoning and associative memory (AM) architecture has been developed for coping with the above difficulties. The system includes two parts. In the first part, fuzzy inference rules based on possibilistic reasoning are applied for dealing with the vague shape and position of a radical in a character. In the second part, a neural network with algorithms based on the Hopfield and bidirectional associative memory function is used to eliminate noise embedded in a radical, subdivided from a character, in order to recognise it (2). In this paper, techniques of normalising varying sizes of a radical are not presented. Therefore, the same size for testing radicals is assumed in the implementation of the system. Figure 1 illustrates the architecture of the two parts in the system.

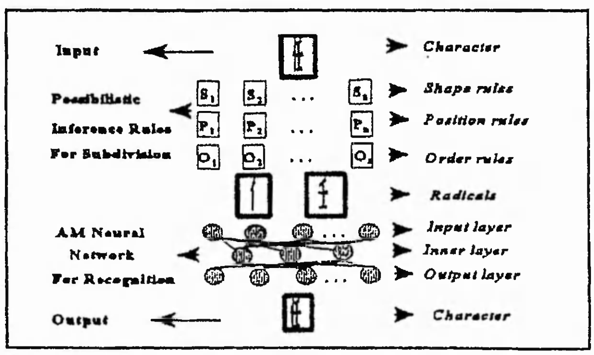


Figure 1 A Fuzzy Neural Network System with Possibilistic Reasoning and Associative Memory Architecture

2. FUZZY POSSIBILISTIC REASONING RULES FOR REPRESENTING STRUCTURE OF RADICALS IN A CHARACTER

In order to subdivide a radical from a Chinese character for further recognition, the shape and position of the radical need to be well described by a representation scheme. Considering features of the vagueness and two-dimensional picture of a radical in a character, fuzzy inference rules based on possibilistic reasoning theory are applied for this task. The system uses two representation methods: situation-rules to describe the position of a radical, and shape-rules to deal with the shape domain of radicals, possibility distribution and measure techniques.

2.1 Fuzzy Possibilistic Reasoning

Fuzzy possibilistic reasoning for knowledge representation is well suited to dealing with imperfect, uncertain and vague information (3). Reducing the complexity of imperfect information is achieved by information-compressed representations based on if-then rules. These rules are interpreted as logical implications and are termed as possibilistic inference rules defined by the notation $\Re$.

Based on fuzzy set theory, approximate reasoning and probability theory, fuzzy possibilistic representations use conjunctively combined rules to validate a possible resolution from various restrictions. In these if-then rules, antecedent in the *IF* clause and consequence in the *THEN* clause are constrained by their possibility distributions denoted by π. The possibility distributions are related to the interpretation of vague concepts as contour functions of random sets. Physical quantities of the distributions are defined by the possibility measures denoted by Poss_π.

Generally, a possibilistic inference rule $\Re_j$ can be expressed by

$$\Re_j: \textbf{IF } \xi^{S}_j \textbf{ is } \mu_j \textbf{ THEN } \xi^{T}_j \textbf{ is } \nu_j, \qquad j = 1, \ldots, r,$$

or

$$\Re_j: \textbf{IF } \xi^{S}_j{}^{(1)} \textbf{ is } \mu_j^{(1)} \textbf{ AND } \xi^{S}_j{}^{(2)} \textbf{ is } \mu_j^{(2)} \textbf{ THEN } \xi^{T}_j \textbf{ is } \nu_j, \qquad j = 1, \ldots, r,$$

where μ_j, $\mu_j^{(1)}$, $\mu_j^{(2)}$ and ν_j are subsets of possibility distributions on the space sets S_j and T_j with regard to j respectively. ξ is a variable whose values can be arbitrary possibility distributions on S_j or T_j. The symbol **is**, appearing in possibilistic inference rules, serves as a linguistic description of the operator $\sqsubset$ and is therefore to be seen as 'is at least as specific as'.

The relation $\Re$ of all rules is

$$\Re = \bigcap_{j=1}^{r} \Re j$$

2.2 Situation Representation

The situation representation method uses inference rules defined by the above interpretation for determining a radical's position. The representation focuses descriptions on (i) the position of a radical in a character, and (ii) the order of extracting a radical from a character.

Let the notation P and O stand for two domains of the position of radicals in a character and the order of extracting a radical from a character respectively, their possibility distributions could be defined as $\pi(P)$ and $\pi(O)$ according to the definition of fuzzy possibilistic reasoning. The possibility measures are given by the notations Poss_π (P) for $\pi(P)$, and Poss_π (O) for $\pi(O)$. The possibilistic inference rules are represented by the notation $\Re^{(PO)}$. If p and o denote variables with the domains P and O respectively, the Poss_π (p) is the possibility measure of p on $\pi(P)$; similarly, Poss_π (o) for o on $\pi(O)$.

2.2.1. Position Variance

The investigation of position variance of radicals in a character is based on their features of a two-dimensional picture and a rectangle appearance, one of the major characteristics in the structure of Chinese characters. The domain of position variable is defined by

P = {*width, length*}.

Because a radical may keep an independent position in a character, the possibility distribution of position variance of a radical on the domain P, shown in Figure 2, is defined by

$\pi(P)$ = {***outside, inside, top, bottom, left, right, middle***}.

Poss_π (P) for $\pi(P)$ is defined by, for instance,

Poss_π (*left*) = {***width*** $\le$ ***⅔ width of P, length = length of P***}.

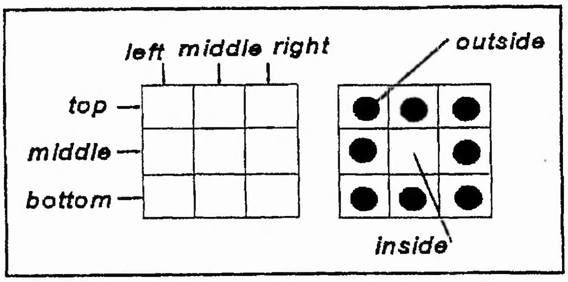


Figure 2 Possible Positions of a Radical

2.2.2 Extraction Order

The extraction order indicates the sequence of radicals extracted from a character which might consist of two or more radicals. The domain of extraction order is expressed by

$$O = \{first, last\}.$$

The possibility distribution of extraction order on the domain O is represented by

$\pi(O) = \{$***outside → inside, inside → outside, top → middle → bottom, top → bottom, left → middle → right, left → right***$\}$.

The notation '→' stands for the sequence from the first to the latter. Distinction of some distribution representations, such as, 'outside → inside' and 'inside → outside', will depend on inference rules between order, position and shape mentioned in the next section.

Developing such basic possibility distribution of order $\pi(O)$ above, a complex distribution could be derived, for instance,

$\pi(O)^{(1)}$ (*top → bottom (left → right)*) =
$\{$***top → bottom left → bottom right***$\}$.

$Poss_\pi$ (O) for $\pi(O)$ is defined by, for instance,

$Poss_\pi$ (*top → bottom*) = $\{$***there are two rectangles***$\}$.

Now, possibilistic inference rules $\Re^{(PO)}$ might be established for representing relations between the position and order of a radical. As examples, several rules are shown as follows.

$\Re^{(PO)}_{(1)}$: **IF** position **is** top **THEN** order **is** first,
$\Re^{(PO)}_{(2)}$: **IF** position **is** bottom **THEN** order **is** last.

2.3 Shape Representation

The shape representation method centres on the shape domain of radicals, possibility distributions and measure technique. Inference rules are established for the representation of radicals relationships between their shape, position and order.

2.3.1. Possibility Distributions and Measures

The domain of radicals is defined as a rectangle in different sizes in terms of features of combined strokes. Basic strokes in the system are defined as dot, horizontal, vertical, left-diagonal, right-diagonal, rising, hook and turn, shown in Figure 3. Any two or more of these strokes may be combined to form a new structure as part of a radical.

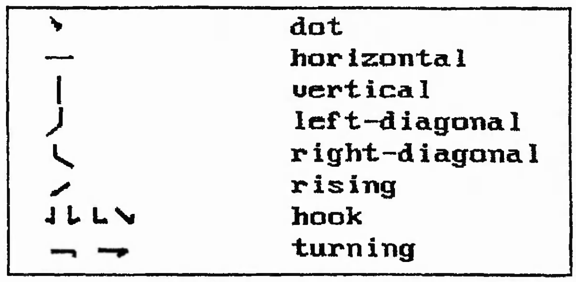


Figure 3 Basic Strokes

The shape domain of radicals is expressed by

$$S = \{rectangle\}.$$

The possibility distributions on the domain S are assigned by different valid combinations of basic strokes, where the validity of the combinations is checked. The modes of combinations are classified as connection and disconnection. The possibility distributions are represented by

$\pi(S) = \{$***combination of basic strokes, basic strokes***$\}$.

In order to determine the shape of a radical, possibility measures are based on evaluation of a continuous line, direction of a line connecting with other lines, priority of such direction and disconnecting distance. For instance, one of the possibility measures $Poss_\pi$ (S) for $\pi(S)$ is defined as follows:

$Poss_\pi$ (*priority of up → down*) = $\{$***up → down, up → down left, right → left, up → down right***$\}$.

2.3.2. Shape Vagueness and Possibilistic Inference Rules

To produce general concepts of forming a radical, the shape vagueness of radicals is investigated for expressing the relation of combining two strokes. The relations can be classified as angle, location, continuous, distance and discontinuous.

The angle relation indicates a contour expression of two connected strokes. For example, it is defined as a contour if an angle is formed by two connected strokes.

The location relation stands for a place of two connected strokes.

The continuous relation expresses a possibility of a contour as part of a radical. A continuous contour is defined if a contour is formed with an angle.

The distance relation is to measure a scope of two disconnected strokes.

The discontinuous relation implies the possibility of a contour which may form two radicals from their shape. A discontinuous contour is decided by the distance measure of two disconnected strokes.

Possibilistic inference rules are established by representations of relations between shape vagueness denoted by $\Re^{(S)}$, between shape and position by $\Re^{(SP)}$, and between shape, position and order by $\Re^{(SPO)}$. For example, the inference rules shown below are defined to divide a character into two parts from the inside to outside.

$\Re^{(S)}{}_{(1)}$: **IF** contour c1 **is** square
AND c2 **is** continuous contour of c1
AND angle of c1 connecting with c2 **is** 90
AND location of c2 **is** on the top middle of c1
THEN shape **is** combination of c1 and c2 (c1+c2).

$\Re^{(SP)}{}_{(2)}$: **IF** shape **is** c1+c2
THEN c1 position **is** outside.

$\Re^{(SP)}{}_{(3)}$: **IF** shape **is** c1+c2 **THEN** c2 position **is** inside.

$\Re^{(SPO)}{}_{(4)}$: **IF** shape **is** c1+c2 **AND** position **is** outside
THEN order **is** last.

$\Re^{(SPO)}{}_{(5)}$: **IF** shape **is** c1+c2 **AND** position **is** inside
THEN order **is** first.

3. AN ASSOCIATIVE MEMORY (AM) NEURAL NETWORK FOR RECOGNISING RADICALS

The neural network in the system consists of three layers: input, inner and output. The number of neurons on the input and inner layers is decided by one of the learning patterns. Neurons on the inner and output layers are dynamically created by a result of combining input radicals.

The network possesses an associative memory function, which offers an important advantage of recalling a stored pattern from its partial or noisy input. Two different methods are applied in association with the function. A static method with the Hopfield auto-associative algorithm is used to the input and inner layers. A dynamic method with the bidirectional hetero-associative algorithm is employed in the inner and output layers. The mathematical formula for the associative memory function is established on the construction of an energy equation,

$$E = -\sum_i \sum_j X_i W_{ij} Y_j + \sum_i \theta_i X_i + \sum_j \varphi_j Y_j \quad \ldots 3.1$$

whose value is reduced or remains constant during the recall procedure (4).

In the learning phase, the associative memory function is used to form the network's connection matrix W. For the training set of input patterns $X_i(u)$ and output patterns $Y_i(u)$, where u = 1, 2 ... M; i = 1, 2 ... N, the weight W(i, j) is determined by the Hebbian rules,

$$W(i, j) = \begin{cases} \sum_u X_i(u)\, Y_j(u) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \quad \ldots 3.2$$

When a portion of the original pattern is used as a retrieval cue, the algorithm is denoted as auto-associative. When the desired output is different from the input, the algorithm is called hetero-associative. The static method requires that the network is trained in advance with specific patterns, and retrieves them from portions or distorted inputs. When the learning and retrieval are embedded in the training process randomly, the method is defined as dynamic. The dynamic bidirectional method used in the system aims to recognize an input pattern that has a relationship with the others.

The network learns standard radicals in the learning phase and identifies them in the training phase. In the recognising process, the network also provides its recognising results with a parameter, which is called the reliability rate, for a further reference. There are three possibilities of recognising a result: recognition, mis-recognition and failure. The reliability rate produced depends on the matching quality and recalling quantity of a pattern.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to test the effect of subdivision inference rules and the associative memory neural network, some experiments have been conducted on a Sun workstation with X View function and the C language to program the algorithm.

In the subdivision processing, the experimentation was focused on different structures of radicals in a character and the technique of subdividing them. In the neural network interface, the system was examined by the effect of different weight factors and error tolerance. Meanwhile, optimising the neurons' structure and analysing results of convergence were investigated.

4.1 Experimental Results for Subdivision Rules

In the first part, 50 printed characters with different structures were used to examine the subdivision rules. The correct radicals have been extracted from 48 out of 50 test characters, i.e. a 96% success rate. Some of them are shown in Figure 4. The two types of incorrect results will be discussed in Section 4.3.

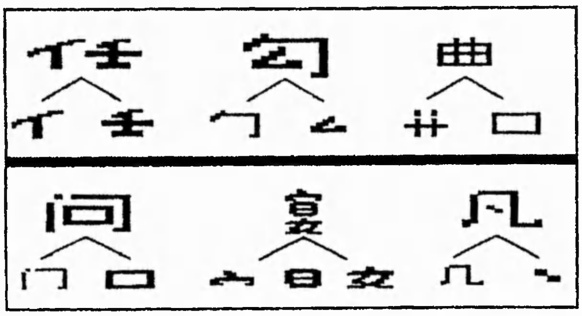


Figure 4. Subdivided Results of Characters with Different Radicals

In the implementation of the first interface, a dynamic scheme is employed for updating possibilistic inference rules while they are being evolved around a sample of special cases.

4.2 Testing Results for AM Neural Network

The testing data for the AM neural network was divided into three parts: (i) different weight factors on recognition; (ii) improving the global convergence; and (iii) reducing the mis-recognition rate.

Table 1 gives results of different weight factors and their effects on recognition for testing data shown in Figure 5.

Weight M	2	3	4
Input Number	4	4	4
Error Rate	10%	10%	10%
Recognition	4	4	4
Mis-Recognition	426	0	1
Failure	0	0	0

Table 1. Results of Different Weights

Three groups of test data with different error rates were used for examining the error tolerance of the system. The error rates were 10%, 20% and 60% respectively. Table 2 gives the recognition results for this trial.

Weight M	3	3	3
Input Number	4	4	4
Error Rate	10%	20%	60%
Recognition	4	4	3
Mis-Recognition	0	1	2
Failure	0	0	0

Table 2. Results of Test Data with different Error Rates

The results in tables 1 and 2 show a better rate of recognition while the weight M is 3.

Figure 5 shows some of learning, testing radicals and recognition results.

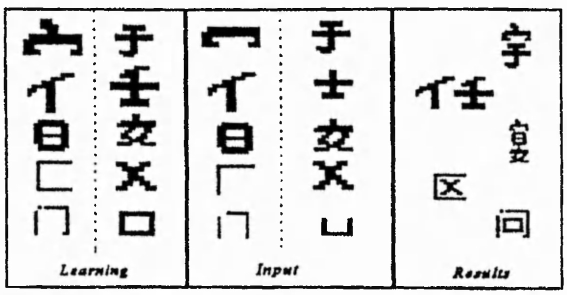


Figure 5. Recognition of Different Radicals

Table 3 shows results for the convergence of local minima onto global minimum for an input radical with different parameters, which refer to the reliability rate, while parameters θ_i and φ_j in the energy equation 3.1 are defined as 0.

Input Number	5	5	5
Error Rate	0%	10%	20%
Weight M	3	3	3
Number of Iterations	1	3	2
Local Minimum	5	14	7
Global Minimum	5	7	5
Reference to Learning Patterns	5	8	7

Table 3. Test Results of Modifying Associative Memory Function

After modifying parameters, θ_i and φ_j to $\frac{1}{2} \Sigma W_{ij}$, the mis-recognition rate of these test data is reduced. Table 4 gives results of the enhancement.

Input Number	5	5	5
Error Rate	0%	10%	20%
Weight M	3	3	3
Number of Iterations	1	2	2
Local Minimum	5	7	5
Global Minimum	5	6	5
Reference to Learning Patterns	5	7	5

Table 4. Testing Results with Modifying Hebbian Rules

4.3 Discussions

In analysing the above results, two types of incorrect results that appeared in the subdivision processing need to be carefully investigated. Incorrect results are involved in two special cases: (i) an individual radical with a discontinuous shape, (ii) a shape of two radicals connected together without a discontinuous part shown in Figure 6 for reference.

In the first case, the radical might be divided into two parts by current inference rules. Therefore, it is necessary to have some special rules, perhaps against existing rules, to deal with these special radicals.

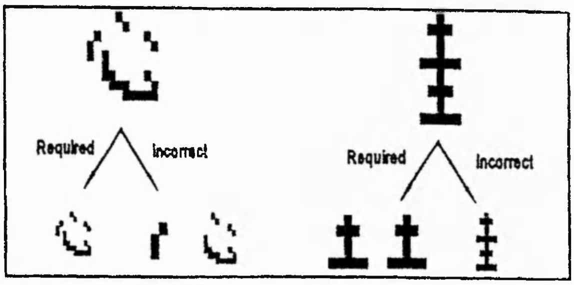


Figure 6. Two Special Cases

In the second case, two radicals are connected, or overlaid together, and form a continuous shape. This is very difficult to deal with by only applying inference rules. Other methods should be investigated for exploring these radicals. Currently, there are two special categories for classifying such radicals.

In the second part, the main problem is mis-recognition. Modification of the associative memory function was enhanced to give better results as shown in Table 3 and Table 4. Consequently, further rejection for mis-recognising results is required for improving the associative memory function.

5. CONCLUSIONS

In this paper, a fuzzy neural network system using possibilistic reasoning and AM neural network architecture was developed. In the system implementation, the possibilistic reference rules and the neural network have been applied for different characters with different radicals. Test results obtained show the efficiency of the system. The modification of the associative memory function in several stages has enhanced the recognition rate of the system. Further research is needed to extend possibilistic rules and improve algorithms for reducing the mis-recognition.

6. REFERENCES

1. Al-Dabass, D., Ren, M. and Su, D. An Associative Memory Artificial Neural Network System with a Combined-Radicals Method for Chinese Character Recognition, ICONIP'95-Beijing: Proceedings of International Conference on Neural Network Information Processing, Beijing, China, Vol.2 of 2, 1995, pp.857-860.

2. Ren, M., Su, D. and Al-Dabass, D. An Associative Memory Artificial Neural Networks System, ECAC'95-London: Proceedings of 1995 European Chinese

Automation Conference, London, UK, 1995, pp.91-96.

3. Kruse, R., Gebhardt, J. and Klawonn, F. Foundations of Fuzzy Systems, 1994, John Wiley & Sons Ltd, pp.81-155.

4. Wang, T. Improving Recall in Associative Memories by Dynamic Threshold, Neural Networks, Vol.7, No.9, 1994, pp.1379-1385.

A Three-Layer Hierarchy for Representing Chinese Characters

M Ren, D Al-Dabass, D Su
Nottingham Trent University, Burton Street, Nottingham NG1 4BU

ABSTRACT

This paper presents a new paradigm of a three-layer hierarchy to represent Chinese characters. The three-layer hierarchy extracts features of the complex topological structure of characters and constitutes a hierarchical representation based on their physical shape and classified according to their logical meanings and pattern structure. Using fuzzy possibilistic reasoning rules, an algorithm based on applying this hierarchy is derived to represent subdivisions within each character in a Chinese text recognition system. Comparisons of the paradigm with other methods are introduced and discussed. Test results obtained show the efficiency of the three-layer hierarchy.

1. INTRODUCTION

A fundamental aspect of computer recognition of Chinese characters centres on a successful knowledge representation structure (1). These characters are essentially pictographic and pose several difficulties including (i) the large volume of characters to be processed, (ii) the complicated internal structure of a character, as well the usual difficulty (iii) that text characters are two-dimensional information structures.

Generally, a Chinese character consists of basic 'strokes' which include: dot, horizontal, vertical, left-diagonal, right-diagonal, rising, hook and turn, as shown in Figure 1. Any two or more of these strokes may be combined to form a new structure as part of a character, where a character may constitute any number from four to thirteen strokes, with some having as many as twenty-six strokes or more.

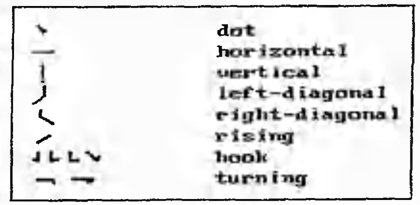


Figure 1 Basic Strokes

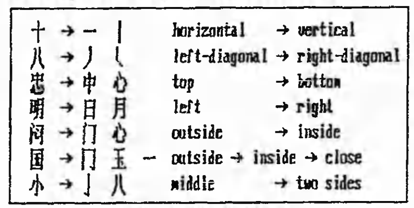


Figure 2 Basic Rules

The basic rules of strokes order in writing Chinese characters are first horizontal, then vertical; first left-diagonal, then right-diagonal; from top to bottom, from left to right; first outside, then inside; finish inside, then close; first middle, then two sides. Figure 2 gives examples illustrating these rules.

The structure representation of Chinese characters may follow one of three paradigms: a two-layer hierarchy, a three-layer hierarchy and the four-corners method. The two-layer hierarchy is seen to relate strokes directly to characters. The number of strokes contained in a character is a major relationship between these strokes and the character. As a traditional method of learning Chinese characters in the real world, this paradigm has been used in recent and current research literature of Chinese character recognition (2). The two-layer hierarchy has the advantages of (i) only several strokes to be learnt and (ii) each stroke could be considered as a simple one-dimensional element. There are also two disadvantages: (i) the large number of strokes in a character, and (ii) the internal structural relationships between strokes and characters (such as a combination format of strokes in a character, their positions and distinguishing the same number of strokes in different characters) are rather difficult to describe by the two-layer paradigm.

The three-layer hierarchy represents the structure of Chinese characters in three formats: characters, radicals and strokes. Radicals are introduced as common components of characters, where each radical is made up of a number of different strokes. The three-layer hierarchy reduces the complexity of a character's structure: a character would consist of less than five radicals (3) while these in turn would individually contain some two to eight strokes. Furthermore, because of position independence of radicals within a character, a character as a picture might be divided into several individual sub-pictures, which considerably eases further recognition by machine.

The four-corners method uses a number of codes to represent Chinese characters, where each code is a four-digit number and each digit (0 to 9) stands for a stroke or combination of strokes from one of the four corners of a character respectively. The four-corners method has a good structure of character classification with a grammar based on horizontal and vertical crossings. It is possible to classify 6763 characters into 1586 groups with 97.5% accuracy (4). The weakness of the method includes complex knowledge of its grammar based on human observation and the ambiguity of the code used to represent several different characters.

In this paper, the three-layer hierarchy is investigated in order to simplify the processing of computer recognition of these characters. The structure features and representation scheme of the hierarchy are described in the Section 2 and 3 respectively. The experimental results, produced by an algorithm based on application of the hierarchy and fuzzy possibilistic reasoning rules, together with discussions of several special cases are presented in Section 4. In Section 5, conclusions and a preliminary comparison of the hierarchy with other methods are given.

2. A THREE-LAYER HIERARCHY

The structure representation in the three-layer hierarchy centres on the relationships of objects in these layers using two aspects: subdivision and classification. Subdivision takes into account the relationship of objects in their shape alteration between layers, which is also called vertical alteration. In the second aspect, classification generalises the relationship of objects in the same layer and forms different categories of these objects, which is termed horizontal classification.

2.1. Characters

The analysis and statistics of characters' structure in this layer are based on the historical development of Chinese characters. Chinese characters are derived from stages of symbols,

pictographs to characters and ideographs. Because the symbols are drawn from actual objects in the real world, most of current Chinese characters still inherit the features of a two-dimensional picture derived from these early symbols. The development of pictographs began to represent some abstract concepts of daily life and eventually made original characters. Figure 3 gives examples to show the development.

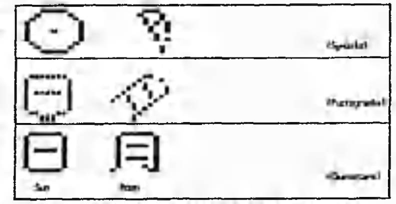


Figure 3 Symbols, Pictographs and Characters

Subsequently, two or more pictographs were combined to form new characters or ideographs for more complex ideas and concepts. Currently, 90% of characters in common use (7000 characters) consist of two or more original characters. At the same time, some of the original characters still keep as individual characters and some of them degenerate as only components of a character. In current Chinese dictionaries, most of these original characters are termed radicals. Figure 4 shows several such characters.

Figure 4 Combined Characters

In summary, the structure of characters has features of a two-dimensional picture, combination of components and different characters containing the same components.

2.2. Radicals

There are two parts included in the layer of radicals: radicals and token-radicals. According to the analysis of characters explained above, a radical may inherit its individual meaning and occupy a completely independent position in a character. A token-radical is developed from such radicals which are further simplified to ultimately become simple combinations of basic strokes. Although some of these are overlaid or crossed in a character, token-radicals common to many characters might be classified into a certain category of radicals in terms of their abstract concepts and shape structure. In the three-layer hierarchy, radicals and token-radicals are classified into 26 different categories (24 radicals' categories and 2 special case categories). An example of radicals' categories is shown in Figure 5.

Whereas radicals are made up by basic strokes in the lower layer, the subdivision and classification of characters in the hierarchy is well organised by the strength of radicals and token-radicals. For instance, classification of a character might depend on a combination of its radicals or token-radicals. Therefore, radicals and token-radicals form the basis of the structure representation in the three-layer hierarchy.

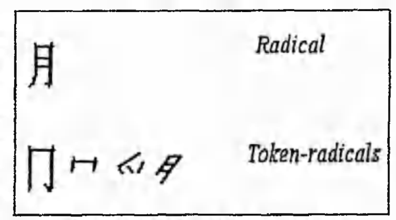


Figure 5 A Radical 'Moon' Category

2.3. Strokes

In the three-layer hierarchy, a stroke is defined to be a continuous line shown in Figure 1. Although radicals form the foundation of subdivision and classification of the hierarchy, the structure of their shapes has to be defined entirely by these primitive strokes. For instance, a combination of a horizontal and vertical strokes might form a token-radical, or a radical, shown in Figure 6. Therefore, the structural representation in this layer will concentrate on these combinations as well as strokes themselves.

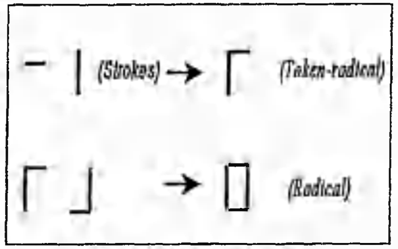


Figure 6 Strokes Combination

On the other hand, because combinations of strokes determine the structure of a character as the outline of a rectangle or a square rather than one of a curve in some other alphabetical characters, the structure description of a rectangle or a square will be taken to form the foundation of possibility distribution of a character's shape.

The basic rules of strokes order, shown in Figure 2, are developed as the essential rules for the structural representation of the hierarchy. In representing the order and position of a radical in a character, these rules will be applied as antecedent conditions.

3. REPRESENTATION SCHEME

The three-layer hierarchy has recently been successfully applied to the derivation of an algorithm to form the nucleus of a Chinese character recognition system (5) (6). The algorithm is based on the application of fuzzy possibilistic reasoning and the syntactic method using a combination of neural networks, the Chang-Jie method and a database for subdivision, classification and recognition. In this paper, the representation will focus on the subdivision of characters into radicals using fuzzy possibilistic rules to incorporate the aspect of vagueness of characters' structure into a knowledge-based system.

The scheme of subdivision representation includes two levels: situation and shape. In the situation level, the representation focuses on (i) the position of a radical in a character, and (ii) the order of extracting a radical from a character. The position variance of radicals in a character will be discussed and classified in Section 3.2.1. The order of extracting a radical will be presented in Section 3.2.2.

Shape vagueness of radicals are investigated in the next level. The interpretation of a radical's basic structure is carried out by possibility measures to constrain the possibility distributions of its shape to an available area, given in Section 3.3.1. General concepts of forming a radical, derived from the manual process of script writing, are translated into operations using fuzzy and possibilistic rules, which will be explained in Section 3.3.2.

3.1. Fuzzy Possibilistic Reasoning

Fuzzy possibilistic reasoning for knowledge representations is well suited to deal with imperfect, uncertain and vague information (7). Reducing the complexity of imperfect information is achieved by information-compressed representations based on if-then rules. These rules are

interpreted as logical implications and are termed possibilistic inference rules defined by the notation $\Re$. Based on the development of fuzzy set theory, approximate reasoning and probability theory, fuzzy possibilistic representations use conjunctively combined rules to validate a possible resolution from various restrictions. In these if-then rules, antecedent in the *IF* clause and consequence in the *THEN* clause are constrained, respectively, by their possibility distributions denoted by π. The possibility distributions are related to the interpretation of vague concepts as contour functions of random sets. Physical quantities of the distributions are defined by the possibility measures denoted by $Poss_\pi$.

Generally, a possibilistic inference rule $\Re_j$ can be expressed by

$$\Re_j: \text{ IF } \xi^S_j \text{ is } \mu_j \text{ THEN } \xi^T_j \text{ is } v_j, \qquad j = 1, \ldots, r,$$

or

$$\Re_j: \text{ IF } \xi^S_j{}^{(1)} \text{ is } \mu_j{}^{(1)} \text{ AND } \xi^S_j{}^{(2)} \text{ is } \mu_j{}^{(2)} \text{ THEN } \xi^T_j \text{ is } v_j, \quad j = 1, \ldots, r,$$

where μ_j, $\mu_j{}^{(1)}$, $\mu_j{}^{(2)}$ and v_j are subsets of possibility distributions on the space sets S_j and T_j with regard to j, respectively. ξ is a variable whose values can be arbitrary possibility distributions on S_j or T_j. The symbol is, appearing in possibilistic inference rules, serves as a linguistic description of the operator $\subset$ and is therefore to be seen as 'is at least as specific as'.

The relation $\Re$ of all rules is

$$\Re = \bigcap_{j=1}^{r} \Re j$$

3.2. Situation Representation

The situation representation focuses descriptions on (i) the position of a radical in a character, and (ii) the order of extracting a radical from a character. If the notation P and O stand for two domains of the position of radicals in a character and the order of extracting a radical from a character respectively, their possibility distributions could be defined as $\pi(P)$ and $\pi(O)$. The possibility measures are given by the notations $Poss_\pi$ (P) for $\pi(P)$, and $Poss_\pi$ (O) for $\pi(O)$. The possibilistic inference rules are represented by the notation $\Re^{(P/O)}$. If p and o denote variables with the domains P and O respectively, the $Poss_\pi$ (p) is the possibility measure of p on $\pi(P)$, similarly, $Poss_\pi$ (o) for o on $\pi(O)$.

3.2.1. Position Variance

The investigation of position variance of radicals in a character is based on their features of a two-dimensional picture and outline of a rectangle. The domain of position variable is defined by

$$P = \{\textit{width, length}\}.$$

Because a radical may keep an independent position in a character, the

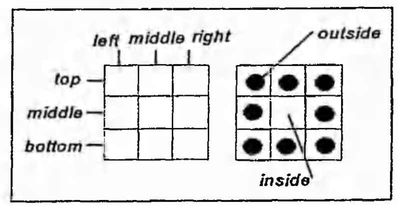


Figure 7 Possible Positions of a Radical

possibility distribution of position variance of a radical on the domain P, shown in Figure 7, is defined by

$\pi(P) = \{$*outside, inside, top, bottom, left, right, middle*$\}$.

$Poss_\pi$ (P) for π(P) is defined by, for instance,

$Poss_\pi$ *(left)* = {*width* $\leq$ ⅔ *width of P, length = length of P*}.

3.2.2 Extraction Order

The extraction order indicates the sequence of radicals extracted from a character consisting of two or more radicals. The domain of extraction order is expressed by

O = {*first, last*}.

Based on some basic rules of radicals order developed from basic strokes rules in writing characters shown in Figure 2, the possibility distribution of extraction order on the domain O is represented by

$\pi(O) = \{$*outside – inside, inside – outside, top – middle – bottom, top – bottom, left – middle – right, left – right*$\}$.

The notation – stands for the sequence from the first to the latter. Note that inside – outside is acceptable as a computer recognition construct. Distinction of some distribution representations, such as, outside – inside and inside – outside, will depend on inference rules between order, position and shape mentioned in the next section.

Developing such basic possibility distribution of extraction order π(O) above, a complex distribution could be derived, for instance,

$\pi(O)^{(1)}$ *(top – bottom (left – right))* = {*top – bottom left – bottom right*}.

$Poss_\pi$ (O) for π(O) is defined by, for instance,

$Poss_\pi$ *(top – bottom)* = {*there are two rectangles*}.

Now, possibilistic inference rules $\Re^{(PO)}$ might be established for representing relations between the position and order of a radical. As examples, several rules are shown as follows:

$\Re^{(PO)}_{(1)}$: **IF** position is top **THEN** order is first,
$\Re^{(PO)}_{(2)}$: **IF** position is bottom **THEN** order is last.

3.3. Shape Representation

The shape representation centres on the investigation of the shape domain of radicals, possibility distributions and measure technique. Inference rules are established for the representation of radicals relationships between their shape, position and order.

3.3.1. Possibility Distributions and Measures

The domain of radicals is defined as a rectangle in different sizes in terms of features of combined strokes. The domain of shape is expressed by

S = {*rectangle*}.

The possibility distributions on the domain S are assigned by different valid combinations of basic strokes shown in Figure 1, where the validity of the combinations is checked. For instance, a dot on top of a left-diagonal is invalid. The modes of combinations are classified as connection and disconnection. The possibility distributions are represented by

$\pi(S) = \{$***combination of basic strokes, basic strokes***$\}$.

In order to determine the shape of a radical, possibility measures are based on evaluation of a continuous line, direction of a line connecting with other lines, priority of such direction and disconnecting distance. For instance, one of the possibility measures $Poss_\pi$ (S) for π(S) is defined as follows,

$Poss_\pi$ *(priority of up – down)* = {***up – down, up – down left, right – left, up – down right***}.

3.3.2. Shape Vagueness and Possibilistic Inference Rules

To produce general concepts of forming a radical, shape vagueness of radicals are investigated for combining two strokes. The relations can be classified as angle, location, continuous, distance and discontinuous.

The angle relation indicates a contour expression of two connected strokes. For example, it is a contour if an angle is formed by two connected strokes. Figure 8 shows three different types of angles from two connected strokes.

The location relation stands for a place of two connected strokes. Figure 9 gives several examples to show the location relation.

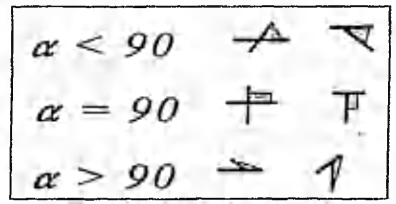


Figure 8 Angle of Strokes Connected

The continuous relation expresses a possibility of a contour as part of a radical. A continuous contour is defined if a contour is formed with an angle.

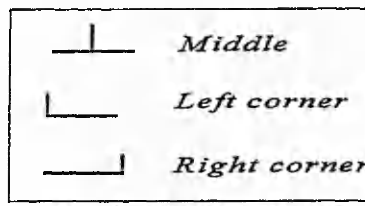


Figure 9 Location of Strokes Connected

The distance relation is to measure a scope of two disconnected strokes.

The discontinuous relation implies the possibility of a contour which may form two radicals from their shape. A discontinuous contour is decided by the distance measure of two disconnected strokes.

Possibilistic inference rules are established by representations of relations between shape vagueness denoted by $\Re^{(S)}$, between shape and position by $\Re^{(SP)}$, and between shape, position and order by $\Re^{(SPO)}$. For example, several inference rules are shown below while a character is divided into two parts from the inside to outside.

$\Re^{(S)}_{(1)}$: **IF** contour c1 is square **AND** c2 is continuous contour of c1
AND angle of c1 connecting with c2 is 90
AND location of c2 is on the top middle of c1
THEN shape is combination of c1 and c2 (c1+c2).
$\Re^{(SP)}_{(2)}$: **IF** shape is c1+c2 **THEN** c1 position is outside.
$\Re^{(SP)}_{(3)}$: **IF** shape is c1+c2 **THEN** c2 position is inside.
$\Re^{(SPO)}_{(4)}$: **IF** shape is c1+c2 **AND** position is outside **THEN** order is last.
$\Re^{(SPO)}_{(5)}$: **IF** shape is c1+c2 **AND** position is inside **THEN** order is first.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to show the effectiveness of subdivision representation based on fuzzy possibilistic reasoning rules, some experiments have been conducted on a Sun workstation with X View function and the C language to program an algorithm based on these rules. The simulated characters were input in bitmap format. A character was digitised by numbers: 1 and -1. In the implementation of the algorithm, a dynamic scheme is employed for updating possibilistic inference rules while they are being evolved around a sample of special cases.

4.1 Experimental Results

In the experiment, 50 printed characters with different structures were used to examine the subdivision results. The correct radicals have been extracted from 48 out of 50 test characters, i.e. a 96% success rate. Some examples of the final results are shown in Figures 10 and 11.

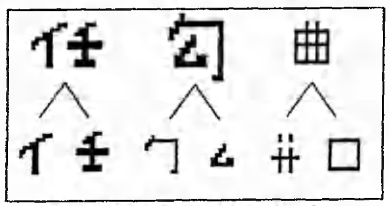


Figure 10 Results Of Different Situations

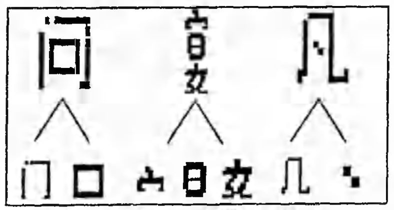


Figure 11 Results of Different Shapes

Other characters which have similar structure to the 50 characters in situation or shape are unlikely to give different experimental results. The two types of incorrect results will be discussed in Section 4.3.

4.2 Dynamic Scheme

The dynamic scheme is a method for dealing with a sample which has a special feature that conflicts with the possibilistic inference rules. For instance, two strokes form a special angle so that the priority of shape measures cannot be carried out. In the implementation of the algorithm, the dynamic scheme was incorporated as an additional tool to ensure the completion of these rules.

4.3 Discussions

In analysing the above results, the incorrect results need careful consideration. Incorrect results are involved in two special cases: (i) an individual radical with a discontinuous shape, (ii) a shape of two radicals connected together without a discontinuous part shown in Figure 12 for reference.

In the first case, the radical might be divided into two parts by current inference rules. Therefore, it is necessary to have some special rules, perhaps against existent rules, for dealing with these special radicals.

In the second case, two radicals are connected, or overlaid together, and form a continuous shape. This is very difficult to deal with by only applying inference rules. Other methods should be investigated for exploring these radicals. Currently, there are two special categories for classifying such radicals.

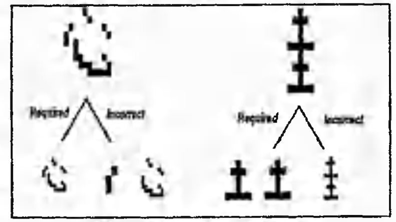


Figure 12 Two Special Cases

5. CONCLUSIONS

In this paper, a three-layer hierarchy of representing Chinese characters was introduced. Using fuzzy possibilistic reasoning rules, a number of inference rules for the subdivision of characters into radicals were produced and implemented. Test results show the efficiency and robustness of the hierarchy.

Compared with the two-layer hierarchy paradigm reported in recent and current and research literature of Chinese character recognition, the three-layer hierarchy offers the advantages of (i) having the capability of processing Chinese characters with a similar structure, (ii) a more systematic internal topological structure of a character and (iii) reducing the vocabulary of learning the characters by machine.

Similarly, both of the three-layer hierarchy and the four-corners method have a good structure representation in the classification of characters. The ambiguity of classified characters is much higher in the four-corners method than the three-layer hierarchy.

Comparative efficiency figures between this and other methods will be the subject of future papers.

6. REFERENCES

1. Tseng, L.Y. and Chuang, C.T. An Efficient Knowledge-Based Stroke Extraction Method for Multi-Font Chinese Characters, Pattern Recognition, 1992, Vol.25, No.12, pp.1445-1458.

2. Hildebrandt, T.H. and Liu, W. Optical Recognition of Handwritten Chinese Characters:

Advances Since 1980, Pattern Recognition, 1993, Vol.26, No.2, pp.205-225.

3. Liu, Y.M. Introduction of Chang-Jie Coding Method, 1993, Run Lin Ltd. (Chinese)

4. Nagy, G. Chinese Character Recognition: A Twenty-Five-Year Retrospective, 9th International Conference on Pattern Recognition, Vol.1, Rome, Italy, 1988, pp.163-167, Sponsored by Int. Assoc. Pattern Recognition, IEEE Computer Society Press, ISBN:0818608781.

5. Al-Dabass, D., Ren, M. and Su, D. An Associative Memory Artificial Neural Network System with a Combined-Radicals Method for Chinese Character Recognition, ICONIP'95-Beijing: Proceedings of International Conference on Neural Network Information Processing, Vol.2 of 2, Beijing, China, 1995, pp.857-860, Co-Sponsored by Asia-Pacific Neural Networks Assembly (APNNA) IEEE Region 10, Technical Co-Sponsored by IEEE Communication Society, Publishing House of Electronics Industry, ISBN:7-5053-3355-0/TP.1288.

6. Ren, M., Al-Dabass, D. and Su, D. Using a Syntactic Method to Partition a Chinese Character, The Caledonian International Engineering Journal, 1996, 96(1), pp.51-60.

7. Kruse, R., Gebhardt, J. and Klawonn, F. Foundations of Fuzzy Systems, 1994, John Wiley & Sons Ltd, pp.81-155.

Foundations for a Proposed Method of Applying Software Metrics in Knowledge Engineering

E Tovar

Universidad Politécnica de Madrid, Campus de Montegancedo, s/n, 28660 Madrid

Spain

ABSTRACT

This paper stresses the importance of evaluation during the process of Knowledge-based System (KBS) development in order to get higher-quality final systems. Much of this work related with research into software metrics can be exploited in KBS development. In this paper, a series of criteria are put forward for software metric classification. These help to differentiate metric application levels. This preliminary work has enabled us to propose a method that facilitates an analysis of the applicability of the metrics used in conventional software to product attributes that are generally generated throughout KBS development. This method has been applied to 36 software metrics, to produce 31 new knowledge engineering metrics.

1. NEED TO EVALUATE KNOWLEDGE-BASED SYSTEMS

While until very recently most Knowledge-based Systems (KBS) were in the exploratory research phases, many have now moved into working environments. Accordingly, many of the requirements specified in the preliminary development phases of such systems correspond to aspects of quality supplied either by the client or by the system's end user. However, as in any other engineering activity, it is only possible to get high-quality final systems, if there is a guarantee that the construction process also is of the required quality.

Metrics provide for the assessment of software products and processes in this field and maximum use can be made of these metrics by establishing relations between different aspects evaluated: the possibility of predicting, early on in the process, whether the best means for achieving the quality objectives foreseen for the final system are being used. However, where an accurate prediction is out of the question, metrics also have other uses, such as the identification of anomalous components. Despite the number of papers completed in this area, success has been limited, for several reasons: no agreement on what are the metrics that are to be applied in each case, non-existence of standardized techniques and methodologies, difficulty in comparing data obtained by different researchers or taken from different systems and, above

AN ASSOCIATIVE MEMORY ARTIFICIAL NEURAL NETWORK SYSTEM WITH A COMBINED-RADICALS METHOD FOR CHINESE CHARACTER RECOGNITION

David Al-Dabass*, Manling Ren* and Daizhong Su**

* Department of Computing
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel: +44 (0)115 941 8418 Ext. 2761
Fax: +44 (0)115 948 6518
E-mail: mar@doc.ntu.ac.uk
dad@doc.ntu.ac.uk

** Department of Mechanical Engineering
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel: +44 (0)115 941 8418 Ext. 2306
Fax: +44 (0)115 948 6506
E-mail: mec3sud@ntu.ac.uk

Abstract

This paper presents an artificial neural network approach for Chinese character recognition. It consists of the following aspects: (i) an associative memory neural network using Hopfield and Kosko algorithms, (ii) syntactic rules in the combined-radicals method for extracting a part of a Chinese character, and (iii) a combination method with rejecting neuron strategy. The system developed by this approach has been tested using a set of data and the results obtained has proved its efficiency and robustness.

1. Introduction

At present, there is an increasing tendency to use artificial neural networks (ANNs) for Chinese character recognition. The main advantage of using ANNs is that they can provide attractive traits such as association ability, parallel processing capability and high error tolerance (Deng & Yu, 1992). Furthermore, many new ANNs' algorithms based on mathematical and biological models have been applied to solve problems that used to be very difficult for other methods.

Computing recognition of Chinese character is a challenging research topic. It not only inherits basic problems from pattern and character recognition, i.e. how to recognize the variable size, position and orientation of an object (Cowell, 1990), but also has difficulties to deal with (i) a large number of characters to be recognized, (ii) the sophisticated topological structure of Chinese characters, and (iii) the variety of character fonts and styles (Xu & Ding, 1992).

In this research, an artificial neural network system has been developed to achieve the following objectives:

* To provide an effective method for recognizing a general Chinese character.
* To break down the topological structure of a Chinese character into its radicals and typefaces that are much more easy to be recognized.
* To solve the variable size and position of a radical.

The structure of the system is shown in figure 1. It consists of six models: a user interface for input, an inference engine with syntactic rules, databases of radicals and typefaces, a neural network, a mechanism of constructing a character by combining radicals and typefaces, and post processor for output.

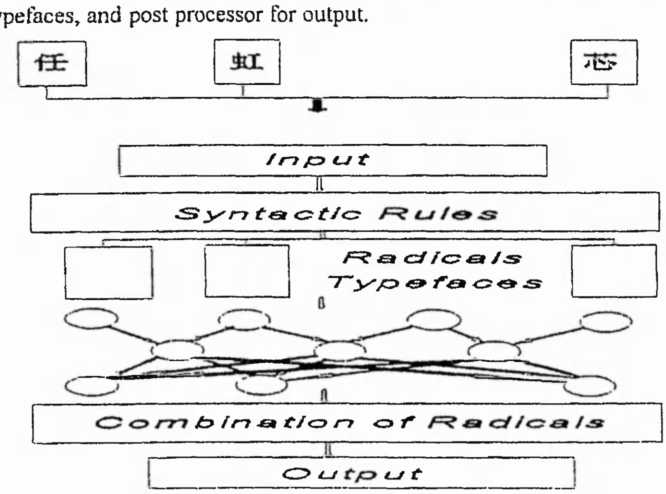


Figure 1 An Associative Memory Artificial Neural Network System with a Combined-Radicals Method

The system has two important features: an associative memory neural network structure and a combined-radicals method. The network is developed using the Hopfield and Kosko algorithms. The combined-radicals method is incorporated with syntactic rules and a combination method with rejecting neuron strategy.

2. The Neural Network Architecture

The neural network in the system consists of three layers: input, inner and output. The number of neurons on the input and inner layers is decided by one of the learning patterns. Neurons on the inner and output layers are dynamically created by a result of partitioning an input character.

The network possesses an associative memory function. Two different methods are involved in the function. A static method with the Hopfield auto-associative algorithm is applied to the input and inner layers. A dynamic method with the Kosko bidirectional hetero-associative algorithm is used in the inner and output layers. The associative memory function offers an important advantage of recalling a stored pattern from its partial or noisy input. The mathematical formula for the associative memory function is established on the construction of an energy equation,

$$E = - \sum_i \sum_j X_i W_{ij} Y_j + \sum_i \theta_i X_i + \sum_j \varphi_j Y_j$$

whose value is always reduced or remains constant during the recall procedure (Wang, 1994).

In the learning phase, the associative memory function is used to form the network's connection matrix W. For a training set of input patterns $X_i(u)$ and output patterns $Y_i(u)$, where u=1,2 ... M; i=1, 2 ... N, the weight W(i,j) is determined by the Hebbian rule,

$$W(i, j) = \begin{cases} \sum_u X_i(u)\, Y_j(u) & if\ i \neq j \\ 0 & if\ i = j \end{cases}$$

When a portion of the original pattern is used as a retrieval cue, the algorithm is denoted as auto-associative. When the desired output is different from the input, the algorithm is called hetero-associative. The static method requires that the network is trained in advance with specific patterns, and retrieves them from portion or distorted inputs. When the learning and retrieving are embedded in the training process randomly, the method is defined as dynamic. The dynamic bidirectional method used in the system aims to recognize a pattern that has a relationship with the other one in a sense.

The network learns formal radicals and typefaces in the learning phase and identifies them, which are extracted from a character, in the training phase. In the recognizing process, the network also provides its recognising result with a parameter, which is called the reliability rate, for a further reference. There are three possibilities of recognising a result: recognition, mis-recognition and failure. The reliability rate produced depends on the matching quality and recalling quantity of a pattern.

3. The Combined-Radicals Method

The combined-radicals method is a method for separating and combining a Chinese or Kanji character. The method is based on the topological structure of Chinese characters (Ren, 1994) and the coding rules used in Cang-Jie Chinese Coding Dictionary (Liu, 1993). In the method, primitive components of Chinese characters are defined as radicals and typefaces rather than strokes, although a radical or a typeface is structured by strokes. According to the Cang-Jie coding rules, a radical is specified as an individual or a regular part in a character. A typeface is considered as a more irregular radical. The combined-radicals method contains two main phases. One is to implement syntactic rules for partitioning a character; and the other is a combination method with rejecting neuron strategy.

3.1 Syntactic Rules For Partitioning

The syntactic rules used in the combined-radicals method are a number of specifications that define how to partition a Chinese character into its radicals and typefaces. There are three approaches involved in these rules: shape, position and order. The shape approach is concerned with determining the shape of a radical or a typeface from a character. The position approach is used to predict the position of a radical in a character. The order approach determines the order of a number of radicals partitioned from a character. An example of three approaches applied to the character '*bright*' is shown in Figure 2. The rules employed by the system are designated as static and dynamic. The static rules deal with general partitioning specifications for defining the shape of a radical or a typeface. The dynamic rules take account of the position and the order of a partitioned radical, while the partitioning process is carried out.

According to the Cang-Jie coding method, the radicals in the system are classified into 24 different groups. Each group contains a number of typefaces. The definitions of static rules are followed by rules of Cang-Jie coding method (Ren, 1994). The dynamic rules, which can be accepted by the system currently, are as follows.

. Left and right.
. Left, middle and right.
. Top and bottom.
. Top, middle and bottom.
. Inside and outside.
. Outside and inside.

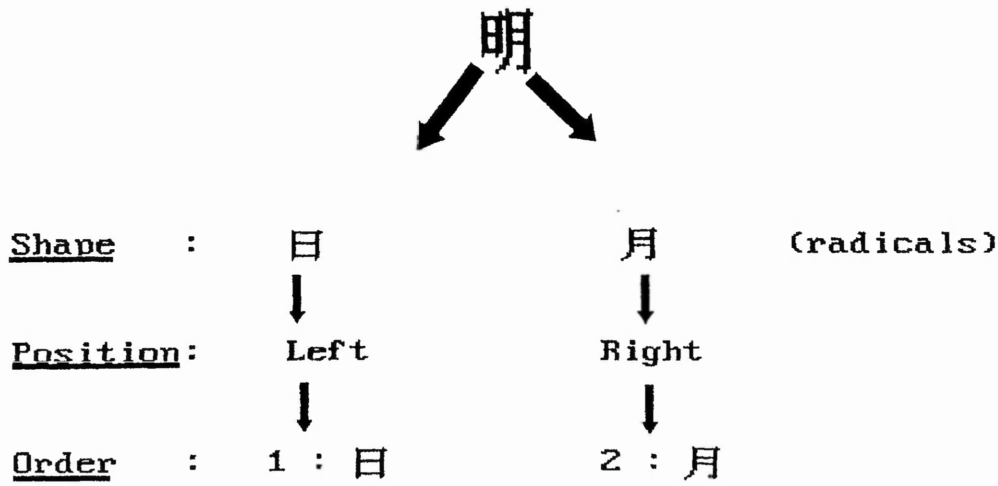


Figure 2 An Example of Syntactic Rules for Partitioning a Chinese Character

The syntactic rules in the system need to be expanded and further development is currently being undertaken.

3.2 The Combination Method with Rejection

The combination method with rejecting neuron strategy is to guarantee correct radicals and typefaces for combination into a complete character. The method is embedded between the network and the Cang-Jie code database. The task of examining an effective radical is appropriated by the rejection strategy. Two rejection schemes are employed in the proposed method: one is by the reliability rate and the other by matching Cang-Jie code. If two radicals with different reliability rates are produced as results of the same input, the one with the lower reliability rate will be rejected. The scheme of matching Cang-Jie code depends on two factors: the category to which the radical belongs, and the order of the partitioned radical. Any mis-matching result will be rejected.

4. Experimental Results and Discussion

To investigate the feasibility of the recognition system described above, some experiments have been conducted. Two different group data are implemented. One is a group where characters consist of similar radicals and typefaces shown in Fig.3 (a). The other one is from different characters displayed in Fig.3 (b).

(a) Similar Characters Group (b) Different Characters Group

Figure 3. Simulating Characters

The network recognizes characters with error rates of three different groups: less than 10%, between 10% and 40% and higher than 40%. For testing the capability of its associative memory, the weight parameter M with two different selections, M=2 and M=3 shown in the table 1 and table 2 separately, is applied to the network.

The system is implemented in the C language on a SUN workstation with X View functions. The simulated characters were input in the bitmap format.

The two group data were tested by different weight parameters of the network, input characters with different error rates, and output results with different reliability rates.

Results of recognition, mis-recognition, failure and rejection are taken into account. The partial testing results are shown in the following tables.

Weight M	2	3
Input Number	4	4
Error Rate	10%	10%
Recognition	4	4
Mis-Recognition	8	2
Failure	0	0

Table 1. Results of Similar Characters Group

Weight M	2	3
Input Number	5	5
Error Rate	20%	20%
Recognition	5	5
Mis-Recognition	51	0
Failure	0	0

Table 2. Results of Different Characters Group

The results in the above tables show the better weight parameter to be 3. Input data organized by different radicals can be achieved at a good recognition rate. For the weight parameter 2, the mis-recognition represents a big problem. It shows that the algorithms used in the system need to be optimized.

5. Conclusion

An associative memory artificial neural network with a combined-radicals method was presented. The system uses the neural network as its main architecture. The combined-radicals method, syntactic rules and combination method with rejecting neuron strategy were employed as techniques to deal with the sophisticated topological structure of a Chinese character. The recognition in several cases was implemented and presented. Further research on improving syntactic rules and optimizing network algorithms is required.

References

Cowell, J. R. (1990), *Character Recognition in Unconstrained Environment*, PhD Thesis, The Nottingham Trent University, UK.

Deng, D. and Yu, Y. (1992), *A Fast-Converging Hamming Net Used in an Offline Chinese Character Recognition System*, IJCNN International Joint Conference on Neural Networks, Vol. 3, pp.602-607.

Liu, Y. M. (1993), Introduction of Cang-Jie Coding Method. Run Lin Ltd. (Chinese)

Ren, M. (1994), *An Associative Artificial Neural Network System with a Combined-Radical Structure for Chinese Character Recognition*, internal report, Department of Computing, The Nottingham Trent University, UK.

Wang, T. (1994), *Improving Recall in Associative Memories by Dynamic Threshold*, Neural Networks, Vol. 7, No. 9, pp.1379-1385.

Xu, N. and Ding, X. (1992), *Printed Chinese Character Recognition Via The Cooperative Block Neural Networks*, IEEE International Symposium on Industrial Electronics, Vol. 1, pp.231-235.

Proceedings of the European
Chinese Automation Conference
London, 16-17th September 1995

An Associative Memory Artificial Neural Network System

Manling Ren*, Daizhong Su** and David Al-Dabass*

* Department of Computing
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel: +44 (0)115 941 8418 Ext. 2761
Fax: +44 (0)115 948 6518
E-mail: mar@doc.ntu.ac.uk
dad@doc.ntu.ac.uk

** Department of Mechanical Engineering
The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
UK
Tel: +44 (0)115 941 8418 Ext. 2306
Fax: +44 (0)115 948 6506
E-mail: mec3sud@ntu.ac.uk

Abstract

This paper presents an artificial neural network system using Hopfield and Kosko associative memory algorithms. Both of algorithms are used for recognising patterns from partial/noisy input or from other associated patterns. The recognition rate of the system has been effectively improved by adjusting the network's parameters and by modifying the associative memory function for converging to a global minimum. The system developed has been tested using a set of data and the results obtained has proved its efficiency and robustness.

1. Introduction

Artificial neural networks have been used as associators, classifiers, and optimisers in many fields (Wang et al, 1993). Associators implement mappings from one field to another for the training pairs. Usually, the neural network is expected to remember all training pairs; namely, the mapping scheme has to perform mappings for all associations. This kind of application is called associative memories (Wang et al, 1993).

Associative memory neural networks presented in this paper include two types: the Hopfield (Hopfield, 1982) and the bidirectional networks (Kosko, 1987 & 1988). The former is an auto-associative neural network that associates an incomplete pattern with an identical learning pattern. The latter is a hetero-associative neural network, which can recognise patterns different from the input patterns. Figure 1 illustrates the major feature of both neural networks.

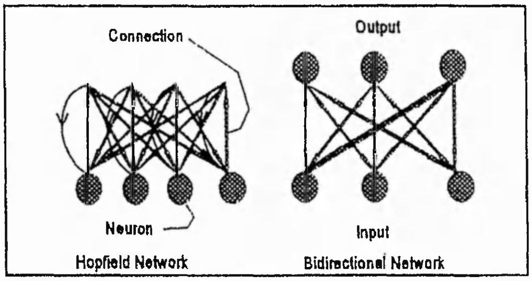


Figure 1. The Architecture of Hopfield and Bidirectional Neural Networks

The advantage of the Hopfield network can be seen as eliminating noise to recognise a pattern from an incomplete input data (Nelson, 1991): Its disadvantages lie in the following two problems. The first problem is the limitation of associative memory, which can be formulated as follows.

$$M = \frac{n}{4\ \log_2 n}$$

The memory M is limited as a constant value when the number of learning patterns n increases to a big value. For instance, M is equal to 25 when n is 1000. The second problem is that the network is just as likely to converge to a local solution as to a global minimum (Dayhoff, 1990).

On the other hand, the benefit of the bidirectional network is that it can set up a relationship between two individual patterns. However, the bidirectional network has the same problems as those that occur in the Hopfield network.

The neural network system presented in this paper tries to use benefits from both the Hopfield and bidirectional networks effectively. By applying the Hopfield network, the system ensures that, for each input pattern, there is either a good training pattern or nothing as an output pattern. To build up a relationship between different patterns by using the bidirectional network, the system only learns several learning patterns and is capable of outputting various combinations of these patterns in terms of the recognition requirements. Meanwhile, the system aims to reduce the number of learning patterns and eliminates the associative memory limitation.

2. The Neural Network Architecture and Algorithms

The neural network in the system consists of three layers: input, inner and output. The number of neurons in the input and inner layers is determined by one of the learning patterns. Neurons in the inner and output layers are dynamically created as a result of an input pattern which can be divided into several individual sub-patterns.

The system employs two methods for dealing with Hopfield and Kosko associative algorithms. A static method using the Hopfield auto-associative algorithm is applied to the input and inner layers. A dynamic method based on the Kosko bidirectional hetero-associative algorithm is used in the inner and output layers. The structure of the system is shown in Figure 2.

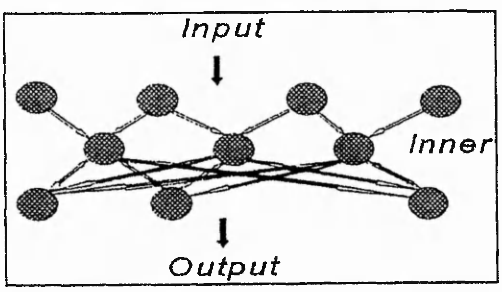


Figure 2. An Associative Memory Artificial Neural Network System

In the input and inner layers, the Hopfield associative memory function offers an important advantage of recalling a stored pattern from its partial or noisy input. The mathematical formula for the associative memory function is established using the energy equation shown below.

$$E = -\frac{1}{2}\sum_i\sum_j X_i W_{ij} X_j + \sum_i \theta_i X_i \quad \text{-- (2.1)}$$

Where X_i and X_j stand for input and its output patterns respectively; W_{ij} is a weight factor; θ_i is a constant parameter; and i = 1, 2 ... N; j = 1, 2 ... N.

Its value is always reduced or remains constant during the recall procedure if the following updating conditions are satisfied.

$$X_i^{n+1} = \begin{cases} 1 & \sum_j W_{ij} X_j^n - \theta_i > 0 \\ X_i^n & \sum_j W_{ij} X_j^n - \theta_i = 0 \\ -1 & \sum_j W_{ij} X_j^n - \theta_i < 0 \end{cases} \quad \text{-- (2.2)}$$

For convergence to an auto-associative value, the Hopfield algorithm uses the Hebbian rules (Wang, 1994) as a special assumption as shown below.

$$\begin{aligned} W_{ij} &= \sum_u (X_i(u) X_j(u)) \\ W_{ii} &= 0 \\ \theta_i &= 0 \end{aligned} \quad \text{-- (2.3)}$$

Where u = 1, 2 ... M and M ≤ N.

Plugging the assumption into the equation in formula (2.1), the energy equation can be defined as follows.

$$E = -\frac{1}{2}\left\{\sum_i\sum_j X_i \left[\sum_u X_i(u) X_j(u)\right] X_j\right\} \quad \text{-- (2.4)}$$

Using this equation, the Hopfield algorithm can achieve the aims of converging to a minimum value. This is the Steepest Gradient Descent Algorithm. The proof omitted here is that equation (2.4) converges to a minimum value.

In the updating procedure shown in (2.2), there are two updating methods: the series and the abreast. The feature of the series method is that each updated variable affects other variables once updating occurred. The method is stated as follows:

$$\left[\begin{array}{l} Do \\ \left[\begin{array}{l} \textit{Update each variable } X_i \textit{ as follows.} \\ X_i^{n+1} = \begin{cases} 1 & \sum_j W_{ij} X_j^n - \theta_i > 0 \\ X_i^n & \sum_j W_{ij} X_j^n - \theta_i = 0 \\ -1 & \sum_j W_{ij} X_j^n - \theta_i < 0 \end{cases} \end{array}\right. \end{array}\right. \quad (2.5)$$

In the abreast method, all variables are updated in the procedure simultaneously. The method is explained as follows:

$$\begin{array}{l} Do \\ \quad \left[\begin{array}{l} \textit{Update each variable } X_i \textit{ as follows.} \\ new\_X_i^{n+1} = \begin{cases} 1 & \sum_j W_{ij} X_j^n - \theta_i > 0 \\ X_i^n & \sum_j W_{ij} X_j^n - \theta_i = 0 \\ -1 & \sum_j W_{ij} X_j^n - \theta_i < 0 \end{cases} \quad (2.6) \end{array} \right. \\ X_i^{n+1} - new\_X_i^{n+1} \end{array}$$

Compared to the series method, the abreast method has the advantage of saving processing time and is employed by the system for the auto-associative memory.

In the inner and output layers, the system employs the bidirectional hetero-associative memory algorithm. The algorithm is based on the convergence of the following energy equation.

$$E = -\sum_i \sum_j X_i W_{ij} Y_j + \sum_i \theta_i X_i + \sum_j \varphi_j Y_j \quad (2.7)$$

Where X_i and Y_j stand for two different patterns respectively; θ_i and φ_j are constant parameters of the equation.

The equation will converge to a hetero-associative value if the updating conditions shown below are satisfied.

$$\begin{aligned} Y_j^{n+1} &= \begin{cases} 1 & \sum_i W_{ij} X_i^n - \varphi_j > 0 \\ Y_j^n & \sum_i W_{ij} X_i^n - \varphi_j = 0 \\ -1 & \sum_i W_{ij} X_i^n - \varphi_j < 0 \end{cases} \\ X_i^{n+1} &= \begin{cases} 1 & \sum_j W_{ij} Y_j^n - \theta_i > 0 \\ X_i^n & \sum_j W_{ij} Y_j^n - \theta_i = 0 \\ -1 & \sum_j W_{ij} Y_j^n - \theta_i < 0 \end{cases} \end{aligned} \quad -- \ (2.8)$$

The Hebbian rules required by equation (2.7) as special assumption are given by (2.9).

$$\begin{aligned} W_{ij} &= \sum_u (X_i(u).Y_j(u)) \\ W_{ii} &= 0 \\ \theta_i &= 0 \\ \varphi_j &= 0 \end{aligned} \quad -- \ (2.9)$$

Similar to the auto-associative memory function, the bidirectional hetero-associative memory function in the system applies the abreast method for its updating procedure.

The system employs two phases for its recognition: the learning phase and the training phase. In the learning phase, the associative memory function is used to form the network's connection matrix W. In the training phase, the function is employed to converge the input pattern to either its own pattern or another one.

Two methods are applied to the system: a static and a dynamic methods. The static method requires that the network is trained in advance with specific patterns, and retrieves them from portion or distorted inputs. When the learning and retrieving are embedded in the training process randomly, the method is termed as dynamic. For the auto-associative memory function, the matrix W is determined by the rules shown in (2.3) using the static method. For the hetero-associative memory function, the matrix W is constructed by the rules shown in (2.9) using the dynamic method. The dynamic bidirectional method used in the system aims to recognise a pattern that has a relationship with the other pattern.

3. Modification of the Associative Memory Function

The modification of the associative memory function in the system is aimed at solving the second problem mentioned in section 1, i.e finding a global minimum value for an input pattern. The objective of the modification is to enhance the recognition rate of the system. This modification has been made in both of the learning and training phases. In the learning phase, the modification centres on choosing a reasonable parameter for building up the matrix W. The choice can ensure the accuracy of recognition and reduce mis-recognition cases that occur in the training phase. There are two steps involved: i) to select a suitable u, shown in (2.3) and (2.9), for the matrix W; ii) to divide learning patterns with different characteristics into the same group for creating a neuron. The aim of creating such neurons is to ensure that each neuron has its own easily distinguishable feature. In the training phase, the modification is centred on how to make local minimum values of an input pattern converge to a global minimum. The convergence quality takes into account for global comparison of different recognition results and makes reference to the original learning patterns. In the recognition process, the network also provides its recognised result with a parameter, which is termed the reliability rate, for further reference. There are three possibilities of recognising a result: recognition, mis-recognition and failure. The reliability rate produced depends on the converging quality and recalling quantity of a pattern. Any failure or error result is saved for further analysis.

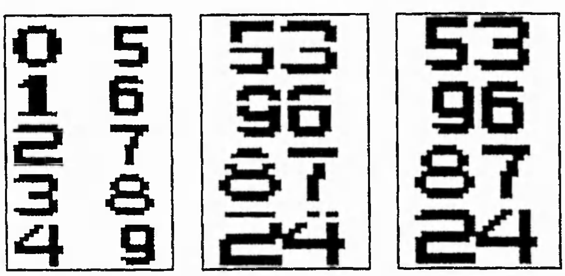
4. Examples and Discussion

The system has been tested using several cases. The following are two examples: number recognition and Chinese character recognition. In the first example, the effect of different weight factors and the error tolerance of the system were examined. In the second example, optimising the neurons' structure and analysing results of convergence were investigated.

The system is carried out by using X view functions and the C language supported by the Unix operating system version 4.1. The learning and testing patterns are input in a bitmap format.

4.1 Number Recognition

Number recognition is implemented for selecting a reasonable weight factor and testing its error tolerance. Figure 3 shows an example of learning patterns, input patterns and their recognising results.



(a) Learning Patterns

(b) Input Patterns with 10% Error

(c) Recognised Results with Weight M=3

Figure 3. Number Patterns Recognition

Using the learning and input patterns presented in Figure 3, Table 1 shows results of different weight factors and their effect on recognition.

Weight M	2	3	4
Input Number	4	4	4
Error Rate	10%	10%	10%
Recognition	4	4	4
Mis-Recognition	426	0	1
Failure	0	0	0

Table 1. Results of Different Weights

Three groups of test data with different error rates were used for examining the error tolerance of the system. The error rates were 10%, 20% and 60% respectively. Table 2 gives the recognition result for this trial.

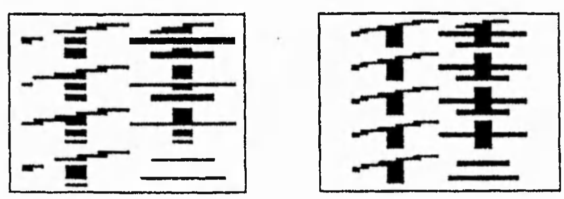
Weight M	3	3	3
Input Number	4	4	4
Error Rate	10%	20%	60%
Recognition	4	4	3
Mis-Recognition	0	1	2

Table 2. Results of Test Data with Different Error Rates

The results in tables 1 and 2 show that better recognition is achieved when the weight M is 3.

4.2 Chinese Character Recognition

Chinese character recognition is used to test the consequence of changing neurons' structure and hence to improve its global convergence effectively. Two groups of data, similar and different characters, were chosen for the trial. Figure 4 shows input patterns and recognition results in a group of similar characters.

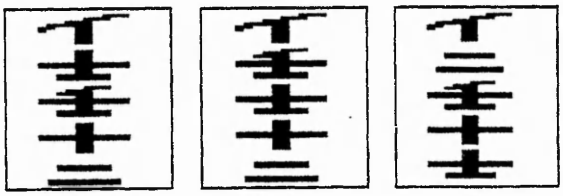


(a) Input Patterns with 20% Error

(b) Results using Learning Patterns (c)

Figure 4. Similar Character Groups

Figure 5 shows three groups of learning patterns in different order for recognition.



(a) (b) (c)

Figure 5. Different Learning Patterns for Recognising Similar Characters

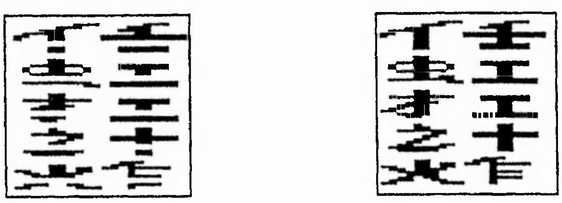
Table 3 gives recognition results of using different structures of neurons created separately using the learning patterns shown in Figure 5.

Learning Patterns	(a)	(b)	(c)
Weight M	3	3	3
Input Number	4	4	4
Error Rate	20%	20%	20%
Recognition	3	3	4
Mis-Recognition	2	2	1

Table 3. Results of Different Neuron Structures

It is clear that there is an optimal scheme, such as learning patterns (c) in this case, for the neurons structure, even if the difference between the three groups of learning patterns is quite small.

Figure 6 is a set of test Chinese characters which are quite different from each other, namely they belong to different character groups.



(a) Input Patterns with 20% Error

(b) Recognition Results

Figure 6. Different Character Groups

Table 4 shows results for the convergence of local minima onto global minimum for an input pattern with different error rates. Several important processing parameters, which refer to the reliability rate, are listed in Table 4.

Analysing the simulation results shown in the above figures and tables, there is a problem, regarding the associative memory function.

Although the modification of the associative memory function has achieved a good result, some of the mis-recognition still exist. Clearly, further optimisation of the associative memory algorithm or a combination of the algorithm with other algorithms is needed to overcome this weakness.

Input Number	5	5	5
Error Rate	0%	10%	20%
Weight M	3	3	3
Number of Iterations	1	3	2
Local Minimum	5	14	7
Global Minimum	5	7	5
Reference to Learning Patterns	5	8	7

Table 4. Test Results of Modifying Associative Memory Function

Nevertheless, it should be stated that the algorithm possesses the advanced features of a quicker convergence speed and the capability of recognising the error patterns. According to the statistics of output patterns from the above trials, the number of iterations is less than 4 for reaching recognition. The capability of recognising the error patterns can be high, up to as much as 60%.

5. Conclusions

An associative memory neural network system with Hopfield and Kosko associative memory algorithms was presented. The system's capabilities of associative memory and recognising patterns were tested using a set of different simulation data. The modification of the associative memory function in several stages enhanced the recognition rate of the system. Further research is needed to modify the associative memory algorithm for converging to a global minimum value while maintaining its advantage of faster convergence speed.

References

Dayhoff, J. E. (1990), Neural Network Architectures: An Introduction, Van Nostrand Reinhold, USA, pp.37-57.

Hopfield, J. (1982), *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proc. Ntl. Acad. Sci. USA, Vol.79, pp.2554-2558.

Kosko, B. (1988), *Bidirectional Associative Memories*, IEEE Transactions on Systems, Man, and Cybernetics, 18(1), pp.49-60.

Kosko, B. (1987), *Adaptive Bidirectional Associative Memories*, Applied Optics, 26(23), pp.4947-4960.

Nelson, M. M. (1991), A Practical Guide to Neural Nets, Addison Wesley Publishing Company, Inc., USA, pp.67-71.

Wang, T. (1994), *Improving Recall in Associative Memories By Dynamic Threshold*, Neural Networks, Vol.7, No.9, pp.1379-1385.

Wang, Y. F., Cruz, J. B. Jr. and Mulligan, J. H. Jr., (1993), *Multiple Training Concept for Back-Propagation Neural Networks for Use in Associative Memories*, Neural Networks, Vol.6, pp.1169-1175.