CHAPTER VII - CLASSIFICATION

7.1 INTRODUCTION

The various aspects of Image Coding, Features used for Filtering and Primitive Extraction were discussed in detail in the last four chapters. This chapter discusses the classification procedure adopted in this work. Initially character description schemes are described, followed by classification methods. An interaction exists between the various stages of the scheme. The new ideas presented are:

1) Hierarchy maintained between the Main and Auxiliary Primitives and the character description, and

2) The provision of openended inventories which makes possible the addition of new members to the group and also handles new variations in the description of the existing characters.

A set of rules is used for the build up of the description of a character from an inventory of primitives. The rules are made as simple as possible and flexible, to add new descriptions of characters or even to add new groups of character sets.

Recognition is also a result of proper description of the
image. This is very true for characters also. In the case of Kanji characters, characters represent a "picture" of the object it stands for. At the lower level where symbols which are seemingly unconnected with the object it represents, the character description should aim at classification or recognition. The character image is described in terms of "phrases" consisting of primitives. Classification is completed by applying rules or heuristically. The recognition process in this work consists of the comparison of the "description" of the character being investigated with a reference description of all the characters in the inventory. The description of the character is built up from its primitives extracted at the primitive extraction stage. With the addition of each new primitive to the description, the character class is reduced and with all the primitives specified the character is classified unambiguously.

Primitives are more purposeful in character description when compared to features. Some primitives may themselves be features while others could be combinations of features. The features proposed in this work, numbering 69, may be appropriately used for extracting primitives.
CHARACTER DESCRIPTION AND PRIMITIVES

Images are made up of structural parts like often occurring lines of definite shape or junctions of two or more lines ... etc.. These structural parts and the interconnections between them determine and describe a character.

Some of the standard primitives used in image description, but not necessarily in character description, are symmetric, like a fork (,<,>), a trident (\), bend (,<,>), cross (\) ... etc.. In this work it is proposed to use the above set of primitives along with a new set defined for the purpose of describing the characters. In the proposed scheme the parts or the primitives are called the MAIN PRIMITIVES and the interconnections between them as the AUXILIARY PRIMITIVES.

An example will show how description is built up employing the two types of primitives. Consider, for example the following two combinations:

```
000 3A1 000 3A2
000 3A1 3A2 000
000 3A1 000 000
000 3A1 000 000
3A0 000 000 3A2
000 3A0 3A2 000
000 3A2 3A0 000
3A2 000 000 3A0
```

The first is a Y junction whereas the second combination stands for a X junction. It can be seen from these two
examples that, in the case of a Y junction two lines, main primitives, join at a point, where as in the case of the X junction, two main primitives cross at a particular point. The elements of these kinds were discussed in detail in the previous chapter. All the characters can be described by choosing the appropriate elements from these classes.

The character images were limited to lower case English cursive script in order to achieve a coalescence in Features and Primitives. As mentioned in Chapter IV, features are defined and extracted at the cell level, whereas primitives are identified and used at the paragraph level. It can be established that this process does not introduce any performance degradations, as long as the number of loops and crosses are small.

7.3 HIERARCHY IN CHARACTER DESCRIPTION

Intuitively it can be felt that a hierarchy exists between the Main Primitives, Auxiliary Primitives and the description of the character in the classification procedure. The description of a character implies that the Auxiliary Primitives can occur only in particular combinations in a certain character and that a primitive can occur only at a particular position. To quote an example, the second
auxiliary primitive GC at column 4 can occur only with the second main primitive QLHL at row 1 and in character "p". This fact can be made use of in the step-by-step recognition scheme implemented here.

The recognition of the auxiliary primitives concurrently indicates the host Main Primitive. Therefore the task of primitive extraction also directly extracts the main primitives. Normally these two decisions, that is Auxiliary Primitive recognition and Main Primitive location concur. In the event of a disagreement, the latter decision prevails. A disagreement, for instance, can occur if an auxiliary primitive is not recognized correctly or if an auxiliary primitive exists at a location where it should not exist. That is, an auxiliary primitive can occur only at designated places in a main primitive and cannot occur at any other place in a character. Conversely, the combinations of auxiliary primitive and the main primitive are fixed in any character.

7.4 RULES ADOPTED FOR CHARACTER RECOGNITION

A set of rules is generated for the structural description of the characters in terms of the primitives. These rules specify the order in which the primitives concatenate. The interconnections of the main primitives are indicated by
the auxiliary primitives residing in them. An analysis of
the characters reveals that:
1). One and two limb junctions may occur at line ends.
2). Three and four limb junctions reside at the waist of
at least one main primitive.
3). A main primitive may have up to four auxiliary primiti-
tives residing in it.

Hence the standard description of a character takes the
form:
\[ MP_1 (AP_{11}, AP_{21}, AP_{31}, AP_{41}) \ast MP_2 (AP_{12}, AP_{22}, AP_{32}, AP_{42}) \ast \ldots \]
\[ \ast MP_n (AP_{1n}, AP_{2n}, AP_{3n}, AP_{4n}) \]

where * stands for a relational operator and "n" may vary
from 1 to 4.

7.4.1 GENERATION OF CHARACTER DESCRIPTION

The following examples show how these rules can be em-
ployed in generating the characters. The notations used in
generating the descriptions are described as and when they
are encountered.
1). A main primitive is accompanied by the auxiliary
primitives residing in it.
2). The contact among primitives is denoted by a " + ",
(plus). These can be seen in the following.
The description of the character "O" in which two main primitives are connected is:

\[ O = TLVL \left( C_1, LC, FC \right) + TLHL \left( R_1, FC, JC \right) + TLVL \left( C_4, GC, JC \right) + TLHL \left( R_4, JC, FC \right) \]

where \( C_1, C_2, C_3, C_4 \) stands for columns 1 to 4 and \( R_1, R_2, R_3, R_4 \) stands for rows 1 to 4.

3. In the case of some of the characters where a main primitive is connected to more than one main primitive the descriptions will be in two forms. The main primitives may be mutually connected or disconnected. Typical examples for the two classes are given below. Consider the first type where the character is described by the following:

\[ A = HLVL \left( C_1, LC, FC \right) + TLHL \left( R_1, FC, GC \right) + HLVL \left( C_3, GC, IJ \right) + TLHL \left( R_4, IJ, LC \right) \]

\[ D = QLVL \left( C_1, LC, FC \right) + QLHL \left( C_3, FC, LB \right) + FLVL \left( C_3, LB, QC, JC \right) + QLRS \left( R_4, QC \right). QLHL \left( R_4, LC, JC \right) \]

The operator "\cdot\" indicates a separator, indicates the fact that these two main primitives are non-touching.

7.5 ILLUSTRATIONS OF CHARACTER DESCRIPTIONS

A sample of lower case English language characters is given in Fig: 4.1. A different set with distortions is given in Fig: 4.2. The description is built up in the order
in which the primitives are to be recognized. Fig:7.1 shows the labeling of the characters employing the primitives. In this figure the descriptions of the characters are also added. Table - 7.1 give the description of all the characters in Fig: - 4.1. The description of characters in Fig: 4.2 will also belong to one among this as normalization takes place at the filtering stage. This is an advantage of the new proposal made in this work.

7.6. OPEN - ENDED INVENTORIES

The inventories for descriptions are kept openended with a view to include more character descriptions. This facility permits the extension of the existing inventory to include more character definitions of the already present classes or adding new classes of characters to the inventory.

A powerful feature of the scheme is its versatility to adapt to situations in which more than one description is permitted for some of the characters. Whenever there is more than one description for a character in the inventory, the recognition becomes easier. More new characters can be included in the inventory by using suitable main primitives and auxiliary primitives from their inventories. By keeping the inventories of both main primitives
See the "+" junction at row 2 column 2 and the F Corner at row 1. The 3A1+3A0+3A1 in C2 is the FLVL, 3A3+3A0 in R1 is the H L at top and the 3A3+3A0+3A3 in R3 is the second H L.

See the L C at row 4, F C at row 3, G C at row 1, and the I J at row 4. The 3A1+3A1+3A1+3A1 in column 1 is the first vertical line in Cl, 3A3 in row 1 is one of the H L's, 3A1+3A1+3A1 in C3 is the second V L and 3A3+3A3+3A3 in row 4 is the second H L.

See 9 J at row 4, L C at row 4, R B at row 2, GC at row 2 and the J C at row 4. 3A1+3A1+3A1+3A1 in C2 is the first V L, 3A3 in R3 is one of the two H L's, 3A1 in C4, is the second V L and 3A3+3A3+3A3 in R4 is the second H L. The 3A2 is R4 is the tail at the bottom left of the character which is not necessary to be identified.

See the F C and G C in row 1, L C in row 2, the L in row 2 and the Q C in row 4. 3A1 in C1 is the connecting link between the two H L's in R1 and R2, 3A1+3A1+3A1 in C3 is the V L and the Q L R S. 3A2 in is the Q C.

Fig: 7.1

LABELING OF THE CHARACTER
DESCRIPTION OF THE FOUR CHARACTERS

CHARACTER "f"  FLVL [C2, PJ, FC] + HLHL [R1, FC].  
                QLHL [R2, PJ] + QLHL [R2, PJ].

CHARACTER "a"  FLVL [C1, LC, FC] + QLHL [R1, FC, GC] +  
                TLVL [C3, GC, IJ] + FLHL [R4, LC, IJ].

CHARACTER "b"  FLVL [C2, 9J, LC, FC] + HLHL [R4, LC,  
                JC] + QLVL [C4, JC, GC] + QLHL [R2,  
                GC, FC) . QLRS [R4, 9J].

CHARACTER "q"  QLVL [C1, LC, FC] + TLHL [R1, FC, GC,  
                PR2] + TLVL [C3, GC, LB, QC] +  
                QLHL [R2, PR1, LC, LB] . QLRS [R4, QC].

NOTE: In the description of the character "q", the  
notation PR1 and PR2 means parallel lines in row1  
and row2 respectively.

Fig: - 7.1.a

LABELING OF CHARACTERS
TABLE 7.1

DESCRIPTION OF THE CHARACTERS

<table>
<thead>
<tr>
<th>CHARACTER</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>a</em></td>
<td>F/T/HLVL [C1, LC, FC] + HL [R1, FC, GC] + H/T/FLVL [C3, GC, IJ] + HL [R4, LC, IJ]</td>
</tr>
<tr>
<td><em>c</em></td>
<td>F/TLLVL [C1, LC, FC] + F/T/HLHL [R1, FC] + F/T/HLH [R4, LC]</td>
</tr>
<tr>
<td><em>e</em></td>
<td>F/TLLVL [C1, LC, RB, FC] + F/T/HLHL [R1, PR2 FC, GC/7C] + QLVL [C4, GC, JC]/QLRS [R2, 7C, BC] + H/QLHL [R2, PR1, JC/BC, RB]. F/T/HLH [R4, LC]</td>
</tr>
<tr>
<td><em>f</em></td>
<td>FLVL [C2, RC, FC] + T/HLHL [R1, FC], Q/HLHL [R2/3, PJ] + Q/HLHL [R2/3, PJ]</td>
</tr>
<tr>
<td><em>g</em></td>
<td>QLVL [C1, LC, RB, FC] + Q/HLHL [R1, PR2, FC, GC] + FLVL [C3, GC, LB, 9J, 6J, JC] + Q/HLHL [R2, PR1, JC, LC]. QLHL [R4, JC, 2C] + QLRS [R4, JC, 9J]. QLRS [R2, 6J]</td>
</tr>
<tr>
<td><em>h</em></td>
<td>FLVL [C1, RB] + HLHL [R2/3, RB, GC] + H/TLLVL [C4, GC]</td>
</tr>
<tr>
<td><em>j</em></td>
<td>FLVL [C3, MC/GC, JC, 9J, 6J, LC] + QLHL [R4, JC, 2C] + HLRS [R3R4, 2C, 9J] + QLRS [R1, 6J]. QLRS [R1, MC]/QLHL [R1, GC]</td>
</tr>
<tr>
<td><em>k</em></td>
<td>FLVL [C2, KJ, MC/GC] + H/QLRS [R1R2, KJ] + H/QLLS [R3R4, KJ]. QLRS [R1, MC]/QLHL [R1, GC]</td>
</tr>
</tbody>
</table>


CHARACTER "n" - F/TLVL [C2, TJ] + H/TLHL [R1, TJ, GC] + F/TLVL [C4, GC]


CHARACTER "p" - FLVL [C2, RB, FC, MC] + H/QLHL [R1, PR2, FC, GC] + QLVL [C4, GC, JC] + H/QLHL [R2, PR1, JC, RB] . QLRS [R1, MC]

CHARACTER "q" - QLVL [C1, FC, LC] + H/QLHL [R1, PR2, FC, GC] + FLVL [C3, GC, LB, QC] + H/QLHL [R2, PR1, LC, LB] . QLRS [R4, QC]

CHARACTER "r" - FLVL [C2, MC, FC] + QLRS [R1, MC] + T/HLHL [R1, FC]

CHARACTER "s" - QLVL [C1, FC, LC] + TLHL [R1, PR2, FC] . T/HLHL [R2, PR1, LC, GC] + H/QLVL [C4, GC, JC] + TLHL [R4, JC]


**TABLE 7.1
DESCRIPTION OF CHARACTERS**
CHARACTER "x" - FLVL [C2/C3, TJ, 1J] + H/QLHL [R1, TJ] + H/QLHL [R1, TJ]. H/QLHL [R4, 1J] + H/QLHL [R4, 1J]


CHARACTER "z" - HLVL [C3, GC, JC, 9J, 6J] + H/QLHL [R2, PR1, GC, 2C] + QLRS [R2, 2C, 7C] + HLHL [R1, PR2, 7C]. HLHL [R4, JC, 2C] + QLRS [R3, 2C, 9J]. QLRS [R2, 6J].

**TABLE - 7.1**

**DESCRIPTION OF CHARACTERS**

**NOTE:**
1) In the case of some characters more than one choice at some of the limbs is already given. However, this is not included in Fig. 7.2. The Decision Tree. However in the computer implementation these and other variants are also accounted for.

2) F/T/H/QL indicate Full Length / Three Quarter Length / Half Length / Quarter Length Lines
and auxiliary primitives openended new primitives can be added to their respective inventories.

7.7 CHARACTER CLASSIFICATION SCHEME

A decision tree is used for the classification scheme. This decision tree enables the interaction between the various elements in the classification procedure namely, the primitive extraction and character classification. The description of a character that is being classified, is built up with the location of each new primitive. Progress along any branch of the decision tree is made with the comparison of the built up description of the character with respect to a standard inventory of all the members of the lower case English language characters. With the addition of each new primitive, the comparison narrows down to the particular character in question and an unambiguous decision is reached finally. Because of the implementation of openended inventories the difficulty of many a problem like that of primitive separation, and extra primitives are avoided. This is achieved by dynamically adding more branches wherever necessary to the decision tree. The use of openended inventories permits this and helps avoid wrong classifications and minimize rejections. In a subsequent section of the chapter the
computer implementation of the classification scheme is presented where a typical example is chosen to show the primitive by primitive extraction and the final unambiguous classification of the character. The variants in the decision tree is also indicated.

7.6 THE DECISION TREE

The classification scheme described in this work uses a decision tree for the interconnection of the various parts like modified image scanning, primitive extraction and the building up of the character description which in turn helps in the correct classification of the unknown character sample. J.R. Ullman [UMAN] in his famous book describes the use of decision tree in pattern recognition. In [NAR1] the use of decision tree is described in the context of hand printed English letters. Many authors have used the decision tree for the classification of particular group of character sets.

An introduction to multistage classification systems has been offered in [KURZ] and [SMAN]. It is stated in [SMAN] that the multistage classification system closely resembles the human recognition system.

The three main components that are involved in the design
of a tree classifier as listed out in [KURZ] are:

1). The choice of the tree structure,
2). The choice of the primitives to be used and
3). The decision rules to be used at each nonterminal node for performing the classification.

The design of the decision tree used in this scheme which is a modification of the classical approach is presented below.

The structure of the decision tree employed in this classification scheme is shown in Fig:7.2, and is based on the character description scheme described earlier in Section 7.5. Any branch lying between the ROOT (designated as start in the figure) and any one of the terminal nodes (which is the identification of the character) is the description of the character. Each node indicates the presence of an auxiliary primitive. Except for the first row of auxiliary primitives, all other nodes show the discovery of a main primitive. It can be seen that every main primitive is represented by a branch in the decision tree. With the extraction of each primitive the membership in the character class decreases in number of members who satisfy the specified characteristics. The weakness of a conventional decision tree is the fact that a wrong
branching taken at any node would affect the performance of the scheme adversely. The hierarchy of the character description scheme in this work avoids such situations. A new feature of this decision tree is the provision for introduction of new branches which lead to the original terminal node. The provision of additional branches leading to the same node allows simple techniques for handling variations in the same character.

Listing the primitives along a branch gives the description of a character which will be finally encountered at the terminal node of the branch. The build up of the description starts with the extraction of the first auxiliary primitive at the first node. The description of the character is enriched with the addition of every new main primitive and/or auxiliary primitive as shown by the successive nodes. The description of the character is thus built up in the form shown below:

\[ AP_1 \ast MP_1 . AP_2 \ast MP_2 \ldots \ldots \ldots AP_{n-1} \ast MP_n \ldots \ldots \ldots (7.2) \]

In a few characters the auxiliary primitives may repeat. The above description is similar to the one shown earlier in section 7.4. The classic form is given in Eq:7.1, whereas the actual form of generation of characters in real applications is of the form of Eq:7.2.
THE DECISION TREE
The decision tree shown in Fig: 7.2 represents the standard form. The actual tree will contain additional branches to accommodate variants of the same character and new characters.

Recognition of characters by the decision tree is shown in Fig: 7.3. Successive reduction in class with each Auxiliary Primitive and the ultimate arrival at the target character, are also shown in Fig: 7.3

7.9 **INTERACTION AMONGST SCANNING, PRIMITIVE EXTRACTION AND CHARACTER CLASSIFICATION**

The subsystems namely, scanning, primitive extraction and character classification interact through the decision tree. The branches of the tree describes all the characters in the inventory. In the normal operation when a change is encountered the primitive is extracted and identified. This output is used to build the description of the character. During every such operation identification of a primitive leads to a decision at the node of the tree. The question of the character class is answered by referring to the hierarchy of the character description, Main Primitives and Auxiliary Primitives.

Hence the description dictionary is used for primitive
## A Character

### It's Extracted Primitives

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Main Primitives</th>
<th>Action Primitives</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 3A1 3A3 3A1</td>
<td>3A1+3A1+3A1+3A1 (FLVL) at C2</td>
<td>FC at R1</td>
</tr>
<tr>
<td>3A2 3A1 000 3A1</td>
<td>3A3 (QLHL) at R1</td>
<td>GC at R1</td>
</tr>
<tr>
<td>000 3A1 3A3 3A3</td>
<td>3A1+3A1 (HLHL) at C4</td>
<td>JC at R3</td>
</tr>
<tr>
<td>000 3A1 000 000</td>
<td>3A3+3A3 (HLHL) at R3</td>
<td>RB at R3</td>
</tr>
<tr>
<td>000 3A1 000 000</td>
<td>QLRS at R2</td>
<td>RB at R3</td>
</tr>
</tbody>
</table>

### Character Description:

- FLVL (C2, RB, MC, FC) + QLHL (R1, FC, GC) + HLVL (C4, GC, JC) + HLHL (R3, JC, RB) + QLRS (R2, 9J)

### Possible Characters:

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Possible Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST MP</td>
<td>f, i, j, k, l, m, n, p</td>
</tr>
<tr>
<td>FIRST AP</td>
<td>f, p, r</td>
</tr>
<tr>
<td>SECOND MP</td>
<td>p</td>
</tr>
<tr>
<td>SECOND AP</td>
<td>p</td>
</tr>
<tr>
<td>THIRD MP</td>
<td>p</td>
</tr>
<tr>
<td>THIRD AP</td>
<td>p</td>
</tr>
<tr>
<td>FOURTH MP</td>
<td>p</td>
</tr>
<tr>
<td>FORTH AP</td>
<td>p</td>
</tr>
<tr>
<td>FIFTH MP</td>
<td>p</td>
</tr>
<tr>
<td>FIFTH AP</td>
<td>p</td>
</tr>
</tbody>
</table>

**NOTE:** Although from the extraction of the second main primitive onwards the character was unambiguously classified as 'p', the procedure was continued till the last step for the sake of completeness.

**Fig: 7.3**

**Recognition Employing the Decision Tree**
by primitive recognition of a character. By adopting this type of interaction the computer emulates human recognition process. As can be seen from Fig:7.2 at any node the class of later possible primitives is reduced in number. This fact itself is utilised in the primitive extraction procedure. It can be said with certainty that for a few characters, if the characters are written normally, their classification can be done with a partial scan of the character image. However even in cases where a partial scan yields correct classification, the image is scanned completely to corroborate the earlier classification decision. For example consider the character recognition scheme in Fig: 7.3. Even though at the third stage, that is extraction of the second Main Primitive "QLHL" at R1, the character can be classified as "p", the routine continues until all the primitives are extracted and the character classification is unambiguously reiterated.

7.10. COMPUTER IMPLEMENTATION OF THE PROPOSAL

In this section the computer implementation of the new proposal is discussed. The section starts with the preparation of the data, digitization of the character, followed by a flow diagram. The primitive by primitive articulated classification of the character, through a
A manual sample collection formed the first stage of the experimental work. A number of subjects were asked to write lower case English language characters on a fine graph sheet. By scaling, the size of each character was brought to a 12 x 12 modified grid size. No constraint was placed on the use of the writing equipment. The choice of the 12 x 12 grid ensures that line continuity will be maintained, whatever be the writing aid used. For binarizing the following rule is used:

"Assign a value of 1 to a cell if the character outline passes through it. Otherwise assign the value of 0."

The binarized image data is entered through a keyboard (this work was started in the year 1987), and is scanned for further processing. Currently all this can be realized using an image scanner / camera combination.

The different units of the programme are shown in the flow chart, depicted in Fig: 7.4. In the scheme envisaged there are three basic operations:

1). Scanning of the modified image

2). Primitive extraction and unifying feature and primitive definitions.
3). Classification of the character through the build up of the primitives.

From the implementation point of view these three operations are further subdivided into many smaller units.

The search for the character is made as follows. The data is fed into the computer. The search subroutine sweeps the character image by checking the value of each pixel. The search begins with the left top pixel, proceeds along the row and when finally the top right pixel is reached the scan moves over to the next row. The procedure is repeated until the values of all the pixels are noted.

The program now moves over to the next subroutine which groups the pixels into cells. There are nine pixels in each cell. A naming subroutine assigns word names to all the cells.

The image is now ready for data reduction, for which a scheme called FILTERING is adopted. A modified image is available at the output of the Filter. All further processing is done on this modified image.

Another subroutine which extracts primitives scans the modified image starting with the left bottom cell (cell 40). Once this subroutine encounters a cell where a non-
Divide the Image into 16 Cells

Name the 16 Cells

Is Naming Over?

Yes

Locate the Replacement Word for Cell NN

No

Fig: 7.4
Fig: 7.4

FLOW CHART FOR COMPUTER IMPLEMENTATION
zero word name is detected the primitive extractor is brought into operation. The primitive is extracted and recorded. If no word name is detected then the program goes to the north neighbor of the first cell. The search continues until a cell with a non-zero word name is detected. The scanning of the image continues until all the cells are scanned, and all the primitives, both main and auxiliary, are extracted. The classification module now takes over and character description starts forming and finally the character classification is effected.

The description of the output is shown below. The character was classified primitive by primitive. The classification procedure is also presented in Fig: 7.3. Fig: 7.5 illustrates the fact that the characters do not loose their recognizability.

7.11 REALISATION OF THE DECISION TREE

The decision tree of this scheme can be realised by scanning the standard description inventory in a particular mode. The character description is written as a string from left to right. In the scan operation also, the build up of the description of the unknown character is made in the same fashion, that is from left to right. With the
addition of each new main or auxiliary primitive the new description is compared with the standard inventory. The members with which this new description does not agree are ignored. With the growth in the addition of each new primitive resulting in the development of the description, the number of members with which the description agrees decrease and finally the agreement comes down to just one character. Even in such cases, the character search is continued along the decision tree until the final node is reached.

7.12 THE GRAPHICAL REPRESENTATION OF THE DECISION TREE

A part of the decision tree is presented in Fig: 7.6. Here the recognition of the characters "a" and "o" is presented. The decision process is explained below for the purpose of explaining the graphical representation of the decision tree.

On spotting the first primitive, a QLVL at the cell 41, the scan moves in a vertical direction moving on to the cell 31 and it spots the first vertical line, a FLVL or a TLVL. By the identification of this Main Primitive the character class reduces to a, b, c, e, h, o, u and w. The HL [R1] does not eliminate any member from the group.
However, the extraction of 'GC' eliminates the character "c" from the class. Discovery of VL [C3] eliminate both "o" and "e" which leaves behind only the character "a". Although only one or two more auxiliary features are required for the final classification of the character, all the remaining auxiliary primitives are extracted.

7.13 CONVENTIONS ADOPTED FOR CHARACTER SCANNING

A set of conventions are formulated for the scanning of the modified character image. These are not invariant. Other conventions are also possible. The details given below cover all the characters. The scan procedure starts from the left bottom cell.

Generally scanning conventions are needed at junctions. At the first junction the system may encounter 1, 2, or 3 limbs. In the case of 1 limb junctions there is no difficulty. In the case of 2 or 3 limb junctions a helical scan will be followed. This means that the limb encountered first by the scan in its operation will be followed only at the end. The scan will proceed to the top of the vertical limb and continue from there to the next primitive moving in a helical path. The scan will come down to the second top primitive if the helical scan is not possible
Images of a few characters to show that the filtering does not affect recognizability.
It may be noted that the scanning of the modified image can be done in more than one way. Description of the character is scan mode dependent, that is, if scanned differently, even if the basis, that is the primitives, are kept the same, the description of the character will change.

7.14 ADDITIONAL BRANCHES IN THE DECISION TREE

Fig: 7.2 shows a decision tree which can be used for the classification of lower case English language characters. If a different set of data is to be classified using the decision tree, the decision tree should be modified to accommodate the members of this new class. Because of the openended inventories used in this scheme the decision tree can be made to accept more members in it, which include both new characters and variants of the existing characters. The extension of the decision tree to include both these types is shown in Fig: 7.7. A comparison of Fig: 7.7 with Fig: 7.2 show that the variants shown are new parallel branches added to the original decision tree.

7.15 RECOGNITION OF CHARACTER VARIANTS WITHOUT ALTERING THE DECISION TREE

This approach provides for the recognition of character
Fig: 7.6

REPRESENTATION OF THE DECISION TREE
variants without altering the decision tree. The two types of variants generally seen in character recognition problem are:

1) Separation of primitives in characters.
2) Extra primitives in characters.

In the case of characters with separated primitives a rule is formulated for an acceptable limit in the distance of separation. In the case of extra primitives another rule is laid down for the number of extra primitives in one character. The logic of classification can be altered slightly to accommodate these variants also.
Fig: 7.7

VARIANTS IN THE DECISION TREE
Fig: 7.7

VARIANTS IN THE DECISION TREE