CHAPTER 4

HANDWRITTEN TAMIL CHARACTER RECOGNITION SYSTEM USING OCTAL GRAPH

4.1 INTRODUCTION

Machine simulation of human functions has been a very challenging research field since the advent of digital computers. In some areas, which entail certain amount of intelligence, such as number crunching or chess playing, tremendous improvements have been achieved. On the other hand, humans still outperform even the most powerful computers in the relatively routine functions such as vision Arica et al (2001).

In this Overview, Character Recognition (CR) is an umbrella term, which has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. Now, the rapidly growing computational power enables the implementation of the present CR methodologies and also creates an increasing demand on many emerging application domains, which require more advanced methodologies. Optical Character Recognition (OCR) deals with the recognition of optically processed characters rather than magnetically processed ones. OCR is a process of automatic recognition of characters by computers in optically scanned and digitized pages of text Pal et al (2004). OCR is one of the most fascinating and challenging areas of pattern recognition with various practical applications. It can contribute immensely to the advancement of an automation process and can improve the interface between man and machine in many applications proposed by Mantas (1986), Govindan et al (1990).
Character and handwriting recognition has a great potential in data and word processing. For instance, automated postal address and ZIP code reading, data acquisition in bank cheques, processing of archived institutional records and more. Combined with a speech synthesizer, it can be used as an aid for people who are visually handicapped. As a result of intensive research and development efforts, systems are available for English language Bozinovic et al (1989), Jianying Hu et al (1996). Chinese/Japanese character recognition system was developed by Deng et al (1994), Chang et al (1996), Yamada et al (1988) and handwritten numeral recognition system was proposed by Lee (1996). However, less attention had been given to Indian language recognition. Some efforts have been reported in the literature for Devanagari characters Bansal et al (1999), Tamil Chinnuswamy et al (1980) and Bangla scripts Chaudhuri et al (1997). The need for OCR arises in the context of digitizing Tamil documents from the ancient and old era to the latest, which helps in sharing the data through the Internet.

4.2 OCTAL GRAPH APPROACH

The proposed approach ventures a solution for offline handwritten recognition, which converts the letter written into an octal graph, by representing each pixel of the given character as a node of a graph. Each node has eight fields so termed as octal graph. The graph tries to represent the basic form of a letter independent of the style of writing. Using the weights of the graphs and by the appropriate feature matching with the predefined characters, the written characters are recognized. Figure 4.1 shows the octal graph representation of a Tamil character. An octal graph unlike a normal graph has a node with eight pointers and a data field. Based on the neighboring pixels the pointer values are assigned to the various fields of the octal node. These octal nodes are connected to the other nodes based on the
threshold value. Figure 4.2 shows the Octal Node Representation of a Tamil Character.

![Octal Graph Representation of a Tamil Character](image)

**Figure 4.1 Octal Graph Representation of a Tamil Character**

| Null | Null | Null |
|------|------|------|
| Null | Data | Null |
| Null | Null |     |

| Null | Null |
|------|------|
| Null | Data |
| Null | Null |

| Null | Null |
|------|------|
| Null | Data |
| Null | Null |

| Null | Null |
|------|------|
| Null | Null |

**Figure 4.2 Octal Node Representation of a Tamil Character**

### 4.3 RECOGNITION PROCEDURE

The system uses octal graph conversion to recognize the handwritten characters. The major phases of the recognition are:

![Work Flow Diagram of Octal Graph Approach](image)

**Figure 4.3 Work Flow Diagram of Octal Graph Approach**
For the proper working of the algorithm, the following issues are taken under consideration:

1) To convert the pattern to an exact similar octal graph, the normalized image should be cleaned such that cells in a single line do not have more than two set cells as neighbours.

2) For the given input the features of the octal graph such as loops, horizontal lines, and vertical lines should be identified correctly.

These factors are taken into consideration while developing the handwriting recognition system.

### 4.3.1 Segmentation

Text line segmentation is an essential pre-processing stage for off-line handwriting recognition in many Optical Character Recognition (OCR) systems. It is an important step because inaccurately segmented text lines will cause errors in the recognition stage. Text line segmentation of the handwritten documents is still one of the most complicated problems in developing a reliable OCR Likforman-Sulem et al (2007).

Handwriting text line segmentation approaches can be categorized according to the different strategies used Nicolas et al (2004). These strategies are projection based, smearing, grouping, Hough-based Louloudis et al (2006), graph-based and Cut Text Minimization (CTM) approach Shi et al (2004). Figure 4.4 show the segmentation representation of a character.
Algorithm

The segmentation process separates the individual characters from the given input. This is done by following steps:

Input : Preprocessed Image

Output : Segmented Image

Step 1: The image is checked for inter line spaces.

Step 2: If inter line spaces are detected then the image is segmented into sets of paragraphs across the interline gap.

Step 3: The lines in the paragraphs are scanned for horizontal space intersection with respect to the background.

Step 4: Histogram of the image is used to detect the width of the horizontal lines.

Step 5: Then the lines are scanned vertically for vertical space intersection.

Step 6: Here histograms are used to detect the width of the words.

Step 7: Then the words are decomposed into characters using character.

Figure 4.4 Segmentation
4.3.2 Normalization

While normalizing images of various sizes into a single standard size, there may be unwanted pixels that are set in a single stroke. This, if passed to the next stage as such, would result in complications in octal graph construction. The complications may be because two points in a line may be connected by more than a single path. This would result in duplicate linkages in two consecutive nodes of a graph which would result in an unwanted loop, which is a critical feature used in recognition of the letters from the learning set. Figure 4.5 shows the normalization of a character. It is done by following steps:

**Algorithm**

Normalization is the conversion of images of various dimensions into fixed dimensions. It is done by following steps:

Input : Segmented Image

Output : Normalized Image

Step 1: The horizontal and vertical ratios of the corresponding dimensions are found.

Step 2: The pixels of these images are grouped into cells with respect to these ratios.

Step 3: Then the pixels in each cell are read.

Step 4: If any pixel is set in a cell, the corresponding cell is set. If none of the pixels are set in a cell, the cell is not set.

Step 5: This normalized image map is subject to cleaning to remove any pixels such that no pixels that form a single line have more than two neighbouring set pixels.
Step 6: This is done by looking for pre-defined pattern of pixels and removing them for cleaning.

![Figure 4.5 Normalization](image)

**4.3.3 Octal Graph Conversion**

The algorithm used for converting the normalized image into an octal graph is critical because the octal graph should represent the features of the letter to a good extent. There are two factors that represent the extent to which the graph represents a letter. Figure 4.6 shows the octal graph conversion of a character. They are:

- The distance between two nodes should be high enough to represent the features of the letter correctly and also low enough not to take up much memory.

- The number of directions in which linkages are possible is chosen as eight. This is because the number of directions in which the linkages are possible must be high enough to express the curvature of the letters correctly. Also it must be low enough to avoid a highly sparse direction pointer array.
Algorithm

The normalized image is converted into an octal graph. This is done by:

- **Input**: Normalized Image
- **Output**: Octal Graph Image

**Step 1**: Count the number of set neighboring cell of each set cell.

**Step 2**: Mark the connecting points and junction points as node.

**Step 3**: The nodes are connected with respect to the direction of the strokes.

**Step 4**: Connect all the nodes created so far with proper direction linkages.

Figure 4.6 Octal graph Formation

4.3.4 Feature Extraction

The next step is feature extraction where individual image glyph is considered and extracted for features. Each character glyph is defined by the following attributes:

1. Height of the character
2. Width of the character
3. Numbers of horizontal lines present (short and long)
4. Numbers of vertical lines present (short and long)
5. Numbers of circles present
6. Numbers of horizontally oriented arcs
7. Numbers of vertically oriented arcs
8. Position of the features
9. Pixels in the various regions.

4.3.5 Recognition

The various features of the input graph are identified. This includes the height and width of the character, number of loops, number of lines (horizontal and vertical), number of curves, etc. These features help in finding the desired match between the input graph and the character that is in the repository. Match the input graph with that of the character in the repository by considering various features such as loops, horizontal lines, vertical lines, curves etc. Compare the features of the input graph with that of the characters in the repository. Compare the level of confidence for each character. If the confidence level is matched with that of the characters in the repository, then the character is recognized.

4.4 RESULTS AND DISCUSSION

The recognition ranks the letters and displays the top 3 ranked letters. If the letter is in the first spot, then it is 100% success, if it is in the second, it is 75% success, if it is in the third, then it is 50% success or else its failure. Based on this ranking an evaluation was done for each letter in the learning set and efficiency of recognition was evaluated for each letter. The following bar graph displays the result of the evaluation. The overall efficiency of the system was found to be 82% (Table 4.1). The evaluation cases consisted of ten samples which consist of 40 good samples, 40 misaligned samples and 20 extremely disfigured samples. This performance of our system is high considering the fact that the existing systems do not recognize disfigured inputs and misaligned inputs at all.
Table 4.1 Recognition Rate Comparison

| S. No | Character Samples | Total No. of Samples | No. of Samples Recognized | Recognition (%) |
|-------|-------------------|----------------------|---------------------------|-----------------|
| 1     | m                 | 100                  | 88                        | 88 %            |
| 2     | M                 | 100                  | 96                        | 96 %            |
| 3     | ,                 | 100                  | 83                        | 83 %            |
| 4     | <                 | 100                  | 83                        | 83 %            |
| 5     | C                 | 100                  | 75                        | 75 %            |
| 6     | C                 | 100                  | 81                        | 81 %            |
| 7     | v                 | 100                  | 78                        | 78 %            |
| 8     | V                 | 100                  | 76                        | 76 %            |
| 9     | I                 | 100                  | 78                        | 78 %            |
| 10    | x                 | 100                  | 89                        | 89 %            |
| 11    | X                 | 100                  | 75                        | 75 %            |

Overall Efficiency 82 %

4.5 SUMMARY

In this chapter the recognition of Tamil Characters was improved using octal graph conversion to get the maximum possible efficiency. Segmentation and Normalization of handwritten characters has been proven to be efficient on application of octal graph conversion, which improves slant correction. Significant increase in accuracy levels has been found with comparison of our method with the others for character recognition. The experimental results show that the accuracy is really improved than the previous study. With the addition of sufficient pre processing the approach offers a simple and fast structure for fostering a full OCR system. The experimental results show that the accuracy is really improved.