

# A Review of Feature Extraction Methods for Handwritten Character Recognition

Pragati Sharma<sup>1\*</sup>, Dr. Jitender Rai<sup>2</sup>

<sup>1</sup> Research Scholar, Sunrise University, Alwar, Rajasthan

<sup>2</sup> Professor, Sunrise University, Alwar, Rajasthan

**Abstract - This work provides a survey of feature extraction strategies for recognizing segmented (de-contextualized) characters when no internet connection is available. Perhaps the most crucial step toward optimal recognition performance in character recognition systems is settling on a suitable feature extraction strategy. Each of the many possible character representations from solid binary characters to character contours to skeletons (thinned characters) to gray-level subimages of each character requires a unique set of feature extraction algorithms. We address the invariance qualities, reconstructability, and predicted distortions and variability of the characters of the various feature extraction approaches. The challenge of knowing which feature extraction technique to use when is also covered. Once a few of interesting feature extraction approaches have been found, they must be experimentally examined to determine which is best for the specific application at hand.**

**Keywords - handwritten, character, extraction method**

-----X-----

## INTRODUCTION

Feature extraction refers to the process of deciding which characteristics of patterns will be the most useful in solving a certain classification issue. It entails reducing the complexity of tools needed to correctly characterize a huge data collection. Numerous variables in the feature set are a major source of difficulty for performance analysis of complicated data. In this case, a lot of memory and processing power are required. The greatest outcomes may be anticipated by focusing on a subset of traits that are specific to the target application. To recognize patterns quickly and accurately, professionals must choose or create features with strong discriminating capacity. Many characteristics have been found and evaluated for use in pattern recognition. In this chapter, we will look at how to effectively describe the data via the use of a few different feature extraction methods.

## FEATURE EXTRACTION SURVEY

Researchers have spent decades perfecting efficient feature extraction methods. Some recent efforts have been summarized briefly.

Within the framework of a segmentation-based handwritten word recognition system, Blumenstein et al. describe and analyze the performance of a unique feature extraction approach for the identification of segmented/cursive characters.

Based on the original direction feature (DF) extraction method, the modified direction feature (MDF) method uses the shape of characters to deduce their directions of motion. This basic idea was developed further to include direction information with a method for identifying pixel transitions between the character's background and foreground.

There were a lot of changes made to the DF extraction method in an effort to enhance it. With the goal of providing a more precise description of the character outline, the method for determining the direction numbers was rethought and redesigned. In addition, a new global feature was included to boost identification precision for the most often confused characters. The MDF was evaluated against the DF and TF extraction methods using a neural network-based classifier. Using a benchmark dataset, MDF performed better than DF and TF methods and was competitive with the best findings in the literature. In tests using the CEDAR dataset, an accuracy of over 89% in character recognition was found.

In their study, Arora et al. demonstrated an OCR for handwritten Devnagari characters. Neural classifier can decipher elementary symbols. Intersection features, shadow features, chain code histograms, and straight line fitting features are the four feature extraction methods employed. Global shadow features are calculated for the

character image, whereas local features such as intersection, chain code histogram, and line fitting are calculated by breaking the character image up into smaller parts. The classification decision made by four multilayer perceptron based classifiers is combined using a weighted majority voting approach. They ran experiments using a dataset of 4,900 samples, and found that taking the top five options resulted in a 92.80% identification rate. The author says his technique has a higher success rate for recognizing handwritten Devnagari characters when compared to other current approaches.

## EXISTING FEATURE EXTRACTION TECHNIQUES

Both statistical and structural characteristics are used in the feature extraction processes used for handwritten character recognition. The statistical characteristics, such as zoning, moments, projection histograms, and direction histograms, are obtained from statistical distributions of pixels. Topological and geometrical aspects of the character, such as strokes and their orientations, endpoints, or the junction of segments and loops, inform structural features. To better handle the wide range of variance seen in photographs of handwritten characters and to bring out their many qualities, it is preferable to integrate structural and statistical data.

In the preceding paragraph, we covered a review of the major works as a whole. It's possible that one approach won't work for all the languages. What works well and is effective in one situation may not work so well in another. The best collection of features for recognizing Malayalam handwriting has to be found. That's why we carefully choose out the features we'll be using for this task. Here are some of the ways that have been tried.

### Meshing Techniques

According to the findings, there is room for improvement in meshing approaches; as a result, many meshing methods were adopted to break down character pictures. What follows is a breakdown of these elements:

- **Fixed Meshing (FM)**

The construction of a fixed mesh involves dividing an  $ab$  picture into  $N$  ( $M1 \times N1$ ) equally sized chunks. Clear partitioning is achieved by dividing the space in half horizontally and vertically. FIGURE 4.1: An  $ab$  picture is fixed-meshed into 9 ( $3 \times 3$ ) blocks of size  $pq$ .

It is possible to do preprocessing and then partition. Aspect ratios larger than one are the norm for standalone Malayalam letters due to their generally wider form. Therefore, prior to partitioning, the photos are size-normalized.

It's common for block sizes to be tiny and data contained inside them to be sparse or nonexistent when the total number of blocks is quite big. In order to fix this issue, we use variable size partitioning and increase the total number of blocks. However, the smaller the block, the less information is contained inside it. There must be a compromise between the block size and the block count. The empirically determined partitioning limit is addressed along with the findings for a variety of approaches and characteristics.

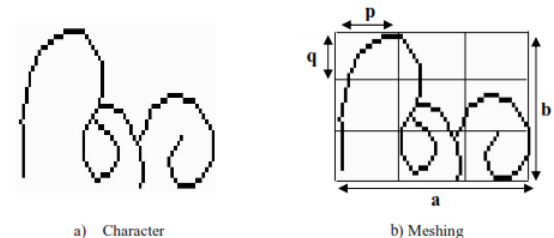


Figure 1: Fixed Meshing

- **Global Elastic Meshing (GEM)**

To create a global elastic mesh, we divide the character's horizontal and vertical histogram into  $N$  intervals, where  $N$  is the number of intervals that include  $N$  histograms each. In a picture, the black pixels are distributed equally among sub-images using an elastic meshing algorithm (Fig.2). This allows for division of varying sizes, with about equivalent information included inside each horizontal and vertical slice. As a result, feature extraction from each block is more accurate than using the fixed meshing approach. Having more partitions means more blocks. The partitioning levels may also be determined empirically. Figure 2 depicts the GEM technique being used to split into two tiers. It is possible to divide a picture into four parts at each level.



Figure 2: Global Elastic Meshing

- **Local Elastic Meshing (LEM)**

After constructing a set of global elastic meshes based on the global distributions of the horizontal and vertical histograms of the given character, local elastic meshing proceeds to refine the meshes at the character level. Next, horizontal and vertical histograms are calculated locally in each block. Finally, in each block, a set of elastic meshes are built by partitioning the local histograms (Fig. 3). The issue with fixed meshing is that there may be no pixels or information included inside individual

blocks. When using local elastic meshing, a minimum number of pixels is guaranteed, or at least one pixel is present in every block. As the number of partitions is increased, each block may consist of a single pixel. Reducing the number of partitions may prevent this from happening. Fig. 3 depicts a 2-tiered partitioning scheme that yields 16 blocks/sub pictures.



Figure 3: Local Elastic Meshing

- **Fuzzy Zoning (FZ)**

First, similar to fixed meshing, an ab picture is segmented into N (M1 x N1) sub images with size pq. The border is shifted away from crisp treatment and toward the inclusion of fuzzy ideas. With fuzzy areas in mind, we assign a membership value of 0.5 to each pixel in the block boundary (region B). Region A's border pixels have a membership value of 0.75, while region C's border pixels have a membership value of 0.25. One pixel is added to the width and height of the subimage, bringing the total to (p+1) (q+1). Fuzzy zoning is seen in Fig. 4.

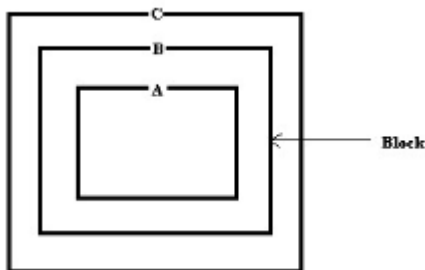


Figure 4: Fuzzy Zoning

### Normalized Vector Distance (NVD)

Counting the amount of background (black) pixels in a block is the quickest way to extract a feature from it. Contrarily, it does not accurately depict how the foreground pixels are distributed. Each foreground pixel's distance from a fixed origin may be used to describe the distribution. Taking the pixel r's (relative to the center) x and y coordinates as input, we can calculate its vector distance  $d_r$  as follows:

$$d_r = \sqrt{x^2 + y^2} \quad \dots\dots\dots 1$$

Then the normalized vector distance of a block p is:

$$NVD_p = \frac{1}{N_p} \sum_{r=1}^{N_p} d_r \quad \dots\dots\dots 2$$

where  $N_p$  is the number of foreground pixel in the block.

The NVD may be determined for FM, GEM, and LEM using the aforementioned eq. (2). However, in FZ, the expanded (fuzzy) border pixels each have a unique membership value. The number of fuzzy boundaries ranges from 4 to 3, or even 2, depending on where the block is placed. For FZ, we may adapt eq. (2) as follows, assuming that there are  $N_a$  pixels in region-A,  $N_b$  pixels in region-B, and  $N_c$  pixels in region-C.

$$NVD_p = \frac{1}{NC} \sum_{r=1}^{N_c} d_r + \frac{1}{N_a} \sum_{r=1}^{N_a} d_r * 0.75 + \frac{1}{N_b} \sum_{r=1}^{N_b} d_r * 0.5 + \frac{1}{N_c} \sum_{r=1}^{N_c} d_r * 0.25 \quad \dots\dots\dots 3$$

where  $NC = N_p - (N_a + N_b)$ .

### Simple Features (SF)

Various feature types are used to represent various visual characteristics. When paired with additional powerful characteristics, certain basic traits demonstrate improved discriminatory skills. These details may be used to enhance recognition systems. The experiment makes use of the following components.

- **Aspect Ratio (AR)**

One of the most fundamental characteristics is the aspect ratio of the character picture after it has been cropped into the smallest possible rectangle. That proportion between the width and height of a picture. When used in conjunction with other features, identification accuracy is shown to be increased. Most individual Malayalam letters are wider than their neighbors. The aspect ratio of an  $a \times b$  image is given by,

$$\text{Aspect ratio} = a / b \quad \dots\dots\dots 4$$

- **Centroid (C)**

Another aspect is the location of the character image's centroid. The centroid will be the location of the central focus (unit value) of all on-pixels. A weighted (unit value) average of all on-pixels positions is used to arrive at this value. The coordinates ( $x_c$ ,  $y_c$ ) of centroid are calculated by eq. (5) and eq. (6).

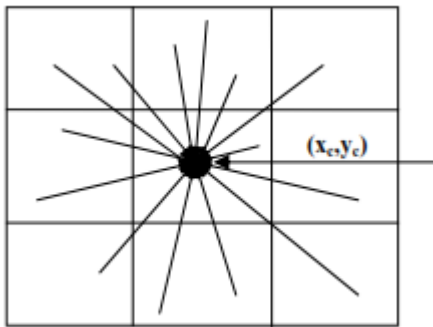
$$x_c = \frac{\sum_{i=1}^n x_i}{n} \quad \dots\dots\dots 5$$

$$y_c = \frac{\sum_{i=1}^n y_i}{n} \quad \dots\dots\dots 6$$

The boundary information is modeled by the distance from the centroid to the boundary points, which is represented by the centroid distance function. To calculate how far away from the centroid a particular block point, denoted by its coordinates  $(x_p, y_p)$ , is, the centroid distance function is used.

$$d_p = \sqrt{([x_p - x_c]^2 + [y_p - y_c]^2)} \quad \dots\dots\dots 7$$

where  $(x_c, y_c)$  is the shape's centroid.



**Figure 5: Image Centroid**

### Thinning Directional Decomposition (TDD)

Normal and directional character pictures are used to extract the NVD feature. Thinning directional decomposition creates four-dimensional (4 D) images—a horizontal pattern (HP), a vertical pattern (VP), a left-slanted pattern (LP), and a right-slanted pattern (RLP) (RP). Imagine that  $p$  is a black pixel in a binary character representation. In Fig. 4.6, we can see the eight neighboring pixels of  $p$ .

$p4$	$p3$	$p2$
$p5$	$p$	$p1$
$p6$	$p7$	$p8$

**Figure 6: Eight neighborhood of pixel  $p$**

Decomposition of a black pixel  $p$  into corresponding sub-pattern is given by:

If  $p1$  or  $p5$  is black pixels, then refine  $p$  into horizontal pattern;

If  $p2$  or  $p6$  is black pixels, then refine  $p$  into left slant pattern;

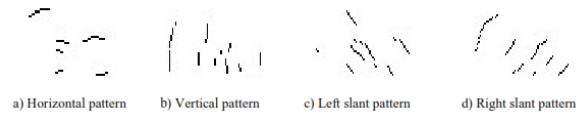
If  $p3$  or  $p7$  is black pixels, then refine  $p$  into vertical pattern;

If  $p4$  or  $p8$  is black pixels, then refine  $p$  into right slant pattern.

Using the above rules, a normal character image (Fig 7) can be decomposed into four directional images (Fig.8).



**Figure 4.7: Character image**



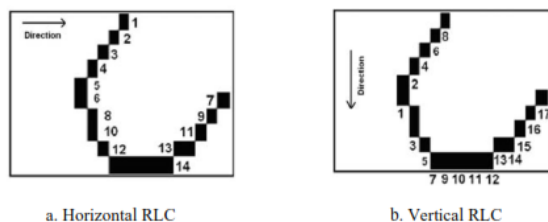
**Figure 7: Four directional sub patterns of a character image**

### Run Length Count (RLC)

This method uses a count of consecutive runs of pixels to represent each row or column of a picture. From an edge detection perspective, only the changes from 0 to 1 and 1 to 0 in a binary picture are meaningful. Thus, they provide a crucial contribution to the procedure of object recognition. The blocking ensures that there will be no duplicates and that each block will include at least one unique element. Transition is measured in RLC units. When doing horizontal RLC, the picture or block is scanned from left to right, and the number of 1s in each contiguous block across all rows is tallied. For vertical RLC, we look for rows and columns where there are nothing but ones, and then count how many of such rows and columns there are. An image's feature vector may be derived from the RLCs of its blocks, both vertically and horizontally. The number of RLCs in each direction of a block is shown in Figure 8a and 8b. With RLC, we may simulate the form based on how its pixels are distributed. RLC can accommodate varying writing styles since precise numerical accuracy is unnecessary. To some extent, the RLC



approach is analogous to zero crossing.



**Figure 4.8: Run Length Count**

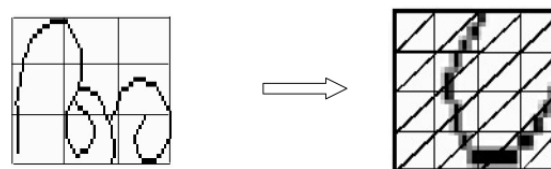
### Direction Features (DF)

Lines, curves, or poly lines with a certain orientation make up a character's stroke. The direction of strokes is critical for distinguishing between distinct characters. In stroke-based character recognition, stroke orientation or direction has been considered for quite some time. In order to use statistical classification based on feature vector representation, characters have also been represented as vectors of orientation/direction statistics. For this purpose, we divide the stroke orientation/direction angle into a defined number of ranges, and use the fraction of the stroke that falls inside each range as a feature value. Thus, the distribution of segment counts is a histogram, which we refer to as the "orientation" or "direction" histogram. The local orientation/direction histogram is the result of calculating the histogram for local zones of the character image to improve the discrimination capacity.

It is common practice to use the term "direction feature" to refer to both orientation and direction histogram characteristics. Directional pattern matching was the original name for character recognition when it was first developed. Distance measures are calculated between the planes extracted from the character picture and the class template, with each plane storing the pixels that correspond to a certain stroke direction. Many factors, such as the character's skeleton orientation, stroke segment, contour chain code, gradient direction, and so on, may be used to establish a stroke's local orientation and direction. It's easy to apply the contour chain code and the gradient direction features, which are why they've become so popular.

### Diagonal Based Feature (DBF)

The extraction of features is often performed in a horizontal and/or vertical plane. Using a diagonal path through the picture, we can also extract character traits (Fig. 9). In order to recognize handwritten letters while offline, Pradeep et al. offer a diagonal feature extraction approach.



**Figure 9: Diagonal based feature extraction**

The  $a_1 \times b_1$  resized character images are divided into  $N$  ( $M_1 \times N_1$ ) zones, resulting in a zone size of  $p \times q$  pixels. The features are extracted from each zone by moving along the diagonals of its respective pixels. Number of diagonal lines,  $D$  in each zone with  $p \times q$  zone size is given as,

$$D = M_1 + N_1 - 1 \quad \dots\dots\dots 11$$

It is shown by adding up the number of "on" pixels along the diagonals of the different "zones." In this case, the feature represented by the zone is the geometric mean of the  $D$  values. Each region is analyzed for its features. There may be certain areas where no on-pixels are present. There is no feature value specified for these areas. Every character has its own unique set of characteristics calculated using the formula  $Z = (M_1 \times N_1)$ . By adding up the totals across rows and down columns, in addition to the  $Z$  features, we may get  $Z_3$  ( $M_1 + N_1$ ) features for each character; this is denoted by the notation DRC.

### PROPOSED FEATURE EXTRACTION TECHNIQUES

The preceding section provided an overview of the various feature extraction techniques. Here, we offer some methods and characteristics that will help boost Malayalam HCR significantly.

#### Modified Meshing Techniques

Modification on GEM, LEM and FZ meshing techniques are proposed. The methods are explained below:

- **Modified Global Elastic Meshing (MGEM)**

The technique is based on the GEM approach, with the following tweaks made. The GEM technique begins by dividing the picture into horizontal  $M_1$  meshes. After that, the picture is captured vertically once again, this time into  $N_1$  meshes, with a mesh size of ( $M_1 \times N_1$ ). Horizontal partitioning comes first in the MGEM approach. The original picture is not captured at the following step, which involves vertically slicing it. Instead of

taking a single horizontally split picture, M1 sub images are separated vertically.

- **Modified Local Elastic Meshing (MLEM)**

The GEM technique is alternatively used in the horizontal and vertical directions in local elastic meshing. To break it down further, 4 distinct pictures result from only one tier of subdivision. These 4 subimages are then further subdivided into 16 smaller pictures. In contrast, in modified local elastic meshing (MLEM), the GEM approach is first performed horizontally only, resulting in 2 sub pictures. These two halves are then divided vertically to provide a further two halves, for a grand total of four. The same procedure is then used to the subsequent level of partitioning. These subimages are partitioned first horizontally and then vertically.

- **Modification on Fuzzy Zoning: Fuzzy Boundary Based Meshing (FBM)**

Lajish has tweaked the original idea of fuzzy zoning in this way. Here, similar to fixed meshing, an ab picture is first partitioned into N ( $M_1 \times N_1$ ) subimages with size pq. The border is shifted away from crisp treatment and toward the inclusion of fuzzy ideas. Then, we assume that the sub picture is  $(p+3)$  by  $(q+3)$  in size. The enlarged border is divided into categories A, B, and C. A more accurate depiction is achieved by assigning varying weights to individual pixels rather than considering them all equally. Values of membership are given to pixels depending on how near an area is to the block. The membership values are scaled such that the closer blocks have higher values and the farther ones have lower values. Membership values of 0.75 are given to all pixels in fuzzy region-A, 0.5 to all pixels in region-B, 0.25 to all pixels in region-C, and 1 to all pixels within the block.

### FEATURE EXTRACTION ALGORITHM FOR MESHING METHODS

Input: Resized ( $72 \times 72$ ), binarized and thinned image.

1. Repeat step 2 for different values of N ( $M_1 \times N_1$ ) namely 9 ( $3 \times 3$ ), 16 ( $4 \times 4$ ), 36 ( $6 \times 6$ ), 64 ( $8 \times 8$ ), 81 ( $9 \times 9$ ) and 144 ( $12 \times 12$ ).
2. Repeat steps 3 to 9 for all input images:
3.
  - a. Apply FM, dividing image into N equal blocks.
  - b. For each block calculate NVD with CO, LO, IC and LC (FMCO, FMLO, FMIC and FMLC).

### FEATURE EXTRACTION ALGORITHM FOR 4 DIRECTIONAL IMAGES

1. For each direction image (HP, VP, LP, VP), calculate NVDs after dividing the image into 4 blocks using

- a. Fuzzy boundary based meshing

- b. Modified global elastic meshing and in each case measuring the distance from:

1. Common origin
2. Local origin

2. Repeat step 1 for extracting 20 features, where the vertical directional image is divided into 8 blocks and other directional images into 4 blocks.

### FEATURE EXTRACTION ALGORITHM FOR DIRECTION FEATURES

The characters are scaled to  $72 \times 72$  and thinned for the experiment. By assigning 0 and 255 as the lowest and maximum values, respectively, the grayscale level is standardized.

Use this process on every picture in your database. Determine the feature values and write them down.

1. Find the horizontal gradient  $g_h$  at each pixel
2. Find the vertical gradient  $g_v$  at each pixel
3. Find the gradient strength  $G$ , at each pixel
4. Map the gradient direction into 12 directions
5. Repeat step 6 for different values of N namely 9 ( $3 \times 3$ ), 16 ( $4 \times 4$ ), 36 ( $6 \times 6$ ), 64 ( $8 \times 8$ ) and 81 ( $9 \times 9$ ).
6. Apply FM:
  - a) Divide the image into N equal sized blocks.
  - b) For each block, find the twelve direction codes (12 DC).
  - c) For each block, find the sum of 12 direction codes (SDC).

### CONCLUSION

In this section, we'll discuss the various feature extraction strategies that were used during the creation of the Malayalam HCR system. There are four distinct meshing methods that are discussed. It is suggested to make certain changes to the global, fuzzy, and local meshing. An alternative to the traditional "common origin" for distance calculations is presented, which uses a "block" or "local" origin. Binary pictures and four-dimensional images are both taken into account when calculating distances between points. For each of the 5 classes into which the binary pictures may be divided, a new character code is added. Extraction of transition information from a binary picture is performed using horizontal and vertical run length counts. Another popular HCR element, the gradient, is also broken out here. Decomposing the gradient direction into 12 directions helps provide more precise results. By adding up these twelve

direction codes inside each block, we may minimize the number of dimensions involved. Features are extracted from pictures in a novel method by first traversing them on the diagonal rather than the conventional horizontal or vertical paths. Feature fusion GBF-RLC is suggested for improved feature representation.

## REFERENCES

A K Jain et al., (2000) A K Jain, R P W Duin and J Mao, "Statistical Pattern Recognition: A review", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), 2000.

A.M.Namboodiri and A. K. Jain, "Online script recognition", IEEE Transactions on Pattern Analysis and Intelligence, Vol.26, No.1, pp.124-130, January 2004.

B V Dhandra et al., (2010) B. V. Dhandra, Mallikarjun Hangarge, Gururaj Mukarambi, "Spatial Features for Handwritten Kannada and English Character Recognition", IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition, RTIPPR, 2010.

Bhowmik, T.K., U. Bhattacharya, and S.K. Parui. Recognition of Bangla handwritten characters using an MLP classifier based on stroke features. in Neural Information Processing.2004. Springer.

C Liu, (2006) Cheng-Lin Liu, "High Accuracy Handwritten Chinese Character Recognition using Quadratic Classifiers with Discriminative Feature Extraction", The 18th International Conference on Pattern Recognition (ICPR'06) 0-7695-2521-0/06 IEEE, 2006.

Das, N., et al. A Novel GA-SVM Based Multistage Approach for Recognition of Handwritten Bangla Compound Characters. in Proceedings of the International Conference on Information Systems Design and Intelligent Applications (INDIA 2012) held in Visakhapatnam, India, January 2012. Springer.

F Enríquez et al., (2012) Fernando Enriquez , Fermin L. Cruz, F. Javier Ortega, Carlos G. Vallejo, Jose A. Troyano, "A Comparative Study of Classifier Combination Applied to NLP Tasks", Information Fusion, Elsevier, Vol. 14, Issue 3, July 2013, pp 255 - 267, doi:10.1016/j.inffus.2012.05.001.

G Nagy, (2000) G. Nagy, "Twenty Years of Document Analysis in PAMI", IEEE Transactions on PAMI, Vol. 22(1), pp 38 – 61, 2000.

H Liu and X Ding, (2005) Hailong Liu and Xiaoqing Ding, "Handwritten Character Recognition using Gradient Feature and Quadratic Classifier with Multiple Discrimination Schemes", Proceedings of

the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05), 2005.

J Hirayama et al., (2007) Junichi Hirayama, Hidehisa Nakayama, and Nei Kato, "A Classifier of Similar Characters using Compound Mahalanobis Function based on Difference Subspace", Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), IEEE, Vol. 1, pp432-436, 2007.

Kim, H.-Y. and J.H. Kim, Hierarchical random graph representation of handwritten characters and its application to Hangul recognition. Pattern Recognition, 2001. 34(2): p. 187-201.

Lajish V L, (2007b) Lajish V. L., "Handwritten Character Recognition using Perpetual Fuzzy Zoning and Class modular Neural Networks", Proceedings of the 4th International Conference on Innovations in IT, pp188 – 192, 2007,978-1-4244-1841-1/08/ 2008 IEEE.

---

## Corresponding Author

**Pragati Sharma\***

Research Scholar, Sunrise University, Alwar, Rajasthan