6 RECOGNITION

The recognition module of OGHTR system is divided into three parts 1) Recognition of diacritic marks 2) Recognition of core characters and 3) Text generation. This chapter describe the recognition approach used for both. The diacritic marks are few in numbers as well as during segmentation its position also available the decision tree approach is most suitable to recognize it. For recognition of core characters, we need trained model as per our discussion in section 3.1. Discussion on architecture of IGHCR module is described in section 6.2. Finally text generation module will merge the recognized core character and diacritic mark together and generate text.

6.1 THE RECOGNITION OF DIACRITIC MARK

The Appendix – I shows list of diacritic mark used in Gujarati language text. The frequently used number of diacritic marks are 12 amongst them 8 appears as single component vowel symbols and 4 appears as multiple component as shown in Table 2. Here note that we are not including processing for diacritic mark “□ ◌□,“ □ ◌□“ and “□: ◌□”.

Table 2 Diacritic mark with respect to its position

Position	Diacritic marks	
	Single component	Multiple component
Left	□◌	
Bottom	◌□ ◌□◌□	
Right	◌□ ◌□	◌□ ◌□ ◌□
Top	◌□◌□	◌□◌□ ◌□

The OGHTR segmentation process is based on connected component based, diacritic marks are segmented as single unit. It means that the diacritic mark which are having multiple component are segmented as more than one CCs. For example, if diacritic mark is “◌□ *kāno ek mātra*” then it is segmented into two CCs 1) “◌□ *kāno*” and 2) “◌□ek mātra”. Therefore, while generating text; arrangement is required to carry out for diacritic marks which have multiple components. Following Table 2 shows position of diacritic marks with respect to its associated character. Note that multiple component diacritic mark appears at both the position. Also, we need to consider that majority of the time writer writes diacritic mark “◌□*be mātra*” in such a way that it appears as single CCs due to join from the bottom part.

Table 3 Structural analysis of diacritic marks

Diacritic Mark	Location of diacritic mark w.r.t core character				Number of end points	Number of branch points	Presence of loop	Reservoir location			
	Top	Bottom	Left	Right				Top	Bottom	Left	Right
◌ે	√	X	X	X	2	0	X	X	X	X	X
◌ં	√	X	X	X	1	0	X	X	X	X	X
◌ૈ	√	X	X	X	2	1	X		X		X
◌ુ	X	√	X	X	1/2	0/1	0/1	√	X	√	X
◌ૂ	X	√	X	X	1/2	0/1	0/1	X	√	X	X
◌ૃ	X	√	X	X	2	X	X	X	X	X	√
િ◌	X	X	√	X	2	X	X	X		X	X
◌ા	X	X	X	√	2	X	X	X	X	X	X
◌ી	X	X	X	√	2	X	X	X		X	X

To recognize single component diacritic mark using decision tree they are studied for their unique characteristics using structural features. The structural feature is extracted such that each value contains some information about structure of the character shape. Feature values are calculated from the structural and geometrical properties of the character. it have many advantages such as font independent, size independent, works well even with shape distortion [65,67].

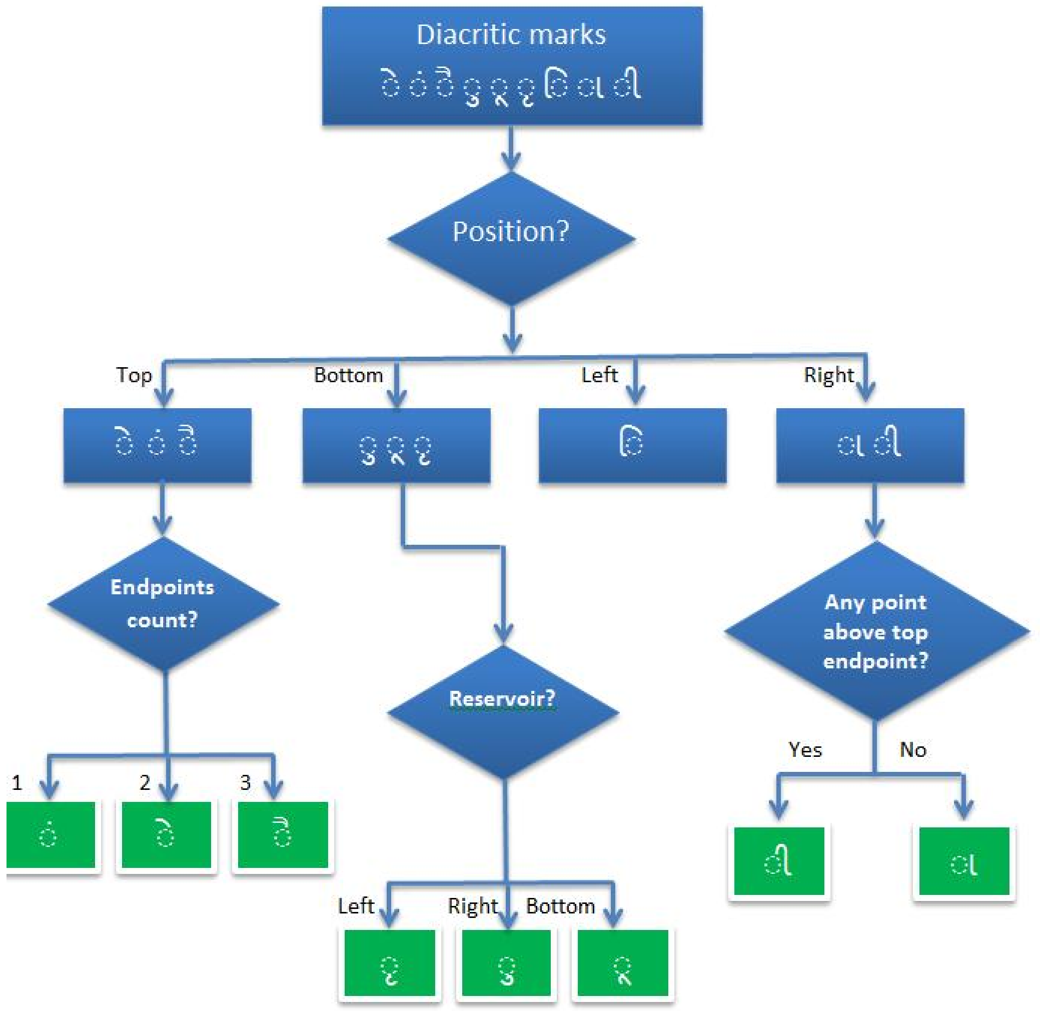


Figure 32 Tree classifier for diacritic mark recognition

Each of the diacritic mark carefully studied for structural characteristics. These characteristics are location of diacritic mark with respect to core character, number of endpoints, number of branch points, loop in character, and location of reservoir. Discussion related to these structural features is available in section 6.2.2.

The analysis of these structural features is shown in Table 3. Based on structural features classification tree is designed to recognize diacritic mark. The tree is also known as decision tree as at every step based on answer of the key question set of possibility reduces [27,67,134]. Processing diacritic mark this way reaches to leaf node yield the resultant diacritic mark. The classification tree is shown in Figure 32.

6.2 RECOGNITION OF CORE CHARACTERS

For recognition of core characters Gujarati alphabets and numerals, OGHTR system uses trained model by training the neural network. The segmented core character component is preprocessed and presented for feature extraction. The extracted features are inputted to trained neural network for recognition. To get trained neural network the isolated character recognition module is implemented.

In this section, the model isolated Gujarati handwritten character recognition (IGHCR) is described. Based on literature review and study of Gujarati character set, initial architecture of isolated character recognition consists of preprocessing, feature extraction and recognition stage. The model IGHCR is developed to achieve objective of recognizing character as Gujarati alphabet and numerals.

The function of preprocessing is to enhance character image so that character features are highlighted and feature extraction task can be carried out. This includes steps to convert image into grayscale, noise removal, binarization, thinning, removal of extended line and cropping. The feature extraction step extracts meaningful features from the character image for the recognition engine.

Based on literature survey, it has been noted that neural network is robust and widely used for HCR and hence we choose neural network for classification [12,108,114]. Based on this, the IGHCR module is divided into two sub modules 1) Training module and 2) Testing module as shown in Figure 33. The objective of the training module is to obtain trained model for recognition engine. The testing module is used to test trained model obtained from the training module.

In training module, the set of isolated sample character images feed into the system along with the labelling of corresponding character. The inputted image required to process thorough preprocessing steps so that features are extracted. These features are then fed into the recognition engine for building knowledge. The recognition engine intended to train with this set of labelled images. In the testing module, an isolated testing character image feed into the system and the system attempt to identify the corresponding character.

The testing module is used to classify the test image to its appropriate class using knowledge created in training module. The test image is fed into the system for preprocessing and feature extraction which are exactly like training module. With the help of extracted set of features, recognition engine classifies the character image to its character class. If classification output of recognition engine matches with the input test image class then it is correct. Now classified image is match to the test image class

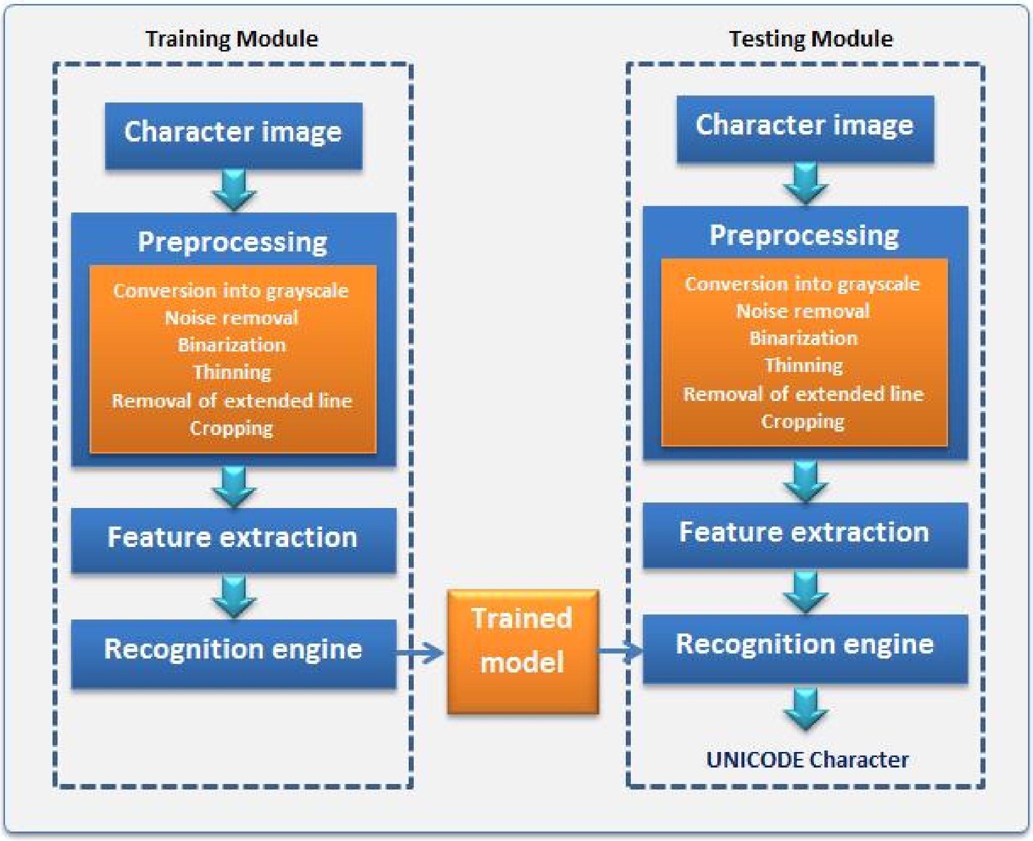


Figure 33 The model for isolated Gujarati handwritten character recognition (IGHCR)

The IGHCR recognize all 49 characters units as per Table 1 in section 3.2.2. Here note that Gujarati character “ ra rə” and numeral “ 2 *bagaṛo”*, character “ pa pə“ and numeral “ 5 *pācharo”* are exactly similar looking characters. Due to exact similarity it is very difficult for machine to recognize these characters as numerals or alphabet. To recognize these characters as numeral or alphabet, post processing is essential using linguistic rule based on natural language processing [135]. As our focus is towards generating text, post processing is currently not in the scope of this thesis.

In Gujarati script 32 consonants are used from which characters "□ ṇa ɳə", "□ la lə", "□ śha ʃə" and "□ ga gə" having more than one component. Also, character "□" is used as part in character "□". Due to this during process of segmentation the first part of character appears as core character and process first for recognition. For example, if for character "□ śha ʃə", the segmentation algorithm segment its first part and labelled it as core character and its second part considered as diacritic matra "◌□ ā ɑ̈ (*kāno*)". So if recognition engine trained with character "□ śha ʃə" as whole then first part of it remains unrecognized or misclassified.

To solve this problem the recognition system trained using part of the character too. It means as soon as segmentation algorithm segment first part of character "□ śha ʃə" and processed for recognition will result in character class for "□ śha ʃə". For second segmented part, the presence of diacritic matra "◌□ ā ɑ̈ (*kāno*)" is verified. Similar strategy is used for characters "□", "□", and "□".

For training neural network, we need samples for part of these characters. Instead of collecting samples for these characters we decided to extract part using connected component method. After the character is converted into binary the connected component extraction is applied. After applying CCs, it is sorted on its X value. Now component are arranged in the order of its part and extracted. First part of "□ ga gə" "□ ṇa ɳə", "□ la lə", "□ śha ʃə" and second part of "□ la lə" are extracted. For each of these character parts 250 extracted samples are added into dataset. Due to these, number of character class reaches to 54. The overall size of dataset is 13500 character samples.

The detail discussion on preprocessing, feature extraction and recognition engine is presented from section 6.2.

6.2.1 Preprocessing

The aim of preprocessing step is to enhance character image so that features can be extracted. Below given Figure 34 shows preprocessing steps IGHCR module. Based on the objective to extract features preprocessing steps consist of grayscale conversion, noise removal, binarization, thinning, extended line removal, and cropping.

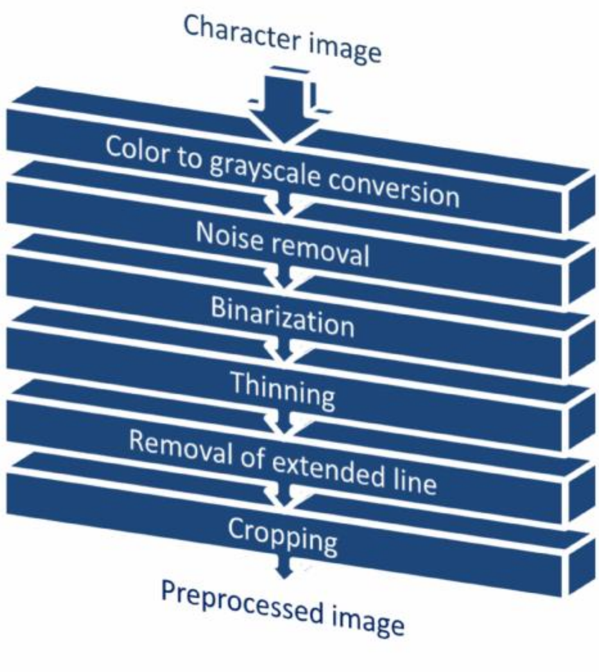


Figure 34 Preprocessing steps of IGHCR

I have previously discussed grayscale conversion, noise removal and binarization in sections 4.2, 4.4 and 4.5 respectively. So in this section we will discuss remaining steps required for preprocessing of character image. After processing these three steps character image is transform to binary image as shown in Figure

35. Now the binary character image is processed for thinning as described in next section.

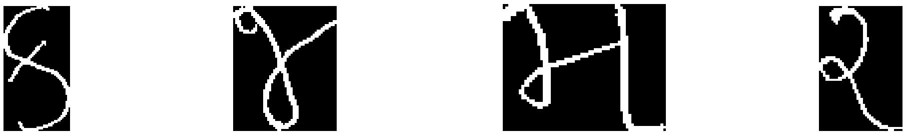


Figure 35 Character image after the binarization for Gujarati “ક ka kə”, ”જ ja dʒə”,”મ ma mə” & “ર ra rə”

6.2.1.1 Thinning

Due to pen, ink, paper quality and noises the scan character image has different stroke width. The stroke width is uneven, some places it is thin, and some places it is thick. The stroke width does not contribute information in features so treated as noise.

To make standardization of trained dataset, thinning is very much crucial & important step. The morphological thinning operation is used to thin the stroke width of character image to single pixel width [65]. It removes excess pixels so that character image will have unique 1-pixel thickness. Thinned characters are free from uneven thickness and it does not affect the results of the feature extraction process. After the thinning process binary character image is shown as in Figure 36.

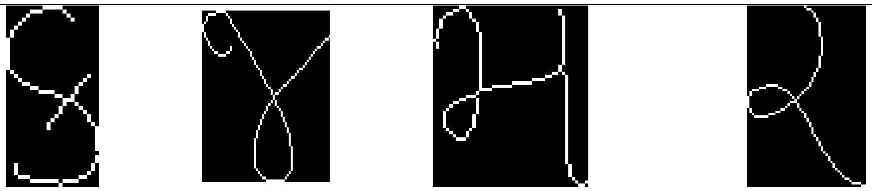


Figure 36 Character image after the thinning process for Gujarati “ક ka kə” ,”જ ja dʒə” ,”મ ma mə” & “ર ra rə”

6.2.1.2 Removal of extended line from the bottom-right part

Many time writers extends stroke at an appropriate position of the character. In Gujarati such extension occurs at bottom right part which causes extra information in character image data. Also this extended line causes increase in width of the character and ultimately affect the recognition process. This again can be treated as noise as it does not add information in character recognition process. The extended lines of some of the Gujarati characters are shown in Figure 37.

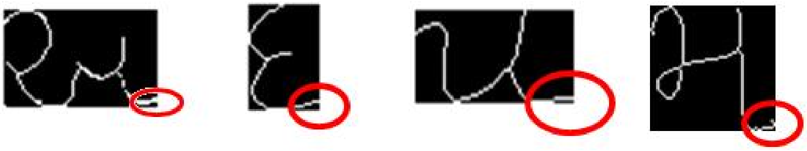


Figure 37 Gujarati characters “અ a ə” “દ da d̪ə” “પ pa pə” and ”મ ma mə” with extended line

To remove extended lines from the character image vertical histogram projection of bottom half of the character image is obtained. The vertical histogram projection array is traced from the last column to the first column and looking for the column which contains more than one object pixels. The part of the image after recorded column is discarded which is extended line. The result of extended line removal step is shown in Figure 38.

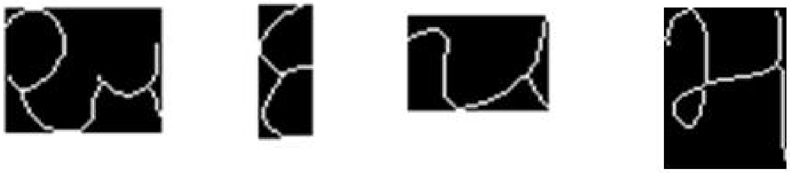


Figure 38 removal of extended line from Gujarati characters “અ a ə”, “દ da d̪ə”, “પ pa pə” and ”મ ma mə”

6.2.1.3 Cropping

Thinned character is then cropped to its optimum available size using bounding box extraction.

The preprocessed character image is then stored and processed for the feature extraction step. The extracted features saved into feature library. The discussion on features is available in next section.

6.2.2 Feature extraction

The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least number of elements. In feature extraction stage each character is represented as a feature vector, which becomes its identity [63,65,84,110,118,136,137]. Features are broadly classified into three categories: 1. Statistical features, 2. Structural features and 3. Global transformation and moments based [65].

The statistical methods based on a planning of how data collected and selected, which helps to make a hypothesis about the type of data. It is based on the probability theory and hypothesis. Statistical distribution of pixels of an image takes care of variations in writing styles [136]. Profiles, projection, zoning, distance and crossings are examples of statistical features. Based on literature review, many researchers used combination of these features for character recognition. For printed characters these features give good results [136,137].

The structural feature is extracted such that each value contains some information about structure of the character shape. Feature values are calculated from the structural and geometrical properties of the character. Examples of structural based features are horizontal lines, vertical lines, aspect ratio, cross points, loops, branch points, curves at top, bottom, left or right etc.

The different feature extraction techniques employed in this work is presented in sub sequent sections. Two novel features water reservoir area statistics (WRA) and partial radial histogram (PRH) which is described in detail in section 6.2.2.5 and 6.2.2.6 respectively.

6.2.2.1 End points and its location (EL)

The end point of character is identified as place where character stroke starts and stops. At this place the character pixel has only one connecting neighbour. The end points of character “□ kɑ kə” and “□ ra ɾə” is depicted in Figure 40.

The hit-or-miss transform is a binary morphological operation that is used to look for bit pattern of corner pixels in an image [65,138]. The hit-and-miss operation is performed in much the same way as other morphological operators, by translating the origin of the structuring element to all points in the image, and then comparing the structuring element with the underlying image pixels. If structuring element is completely match with 1 and 0 value of pixels in the image, then the image pixel that coincides with the centre of the structuring element is set to 1. If it doesn't match, then that pixel is set to 0.

The structuring element used to probe an image for finding end point is depicted in Figure 39. The empty value in structuring element is representing no care value. Other three structuring element is obtained by rotating it in 90, 180 and 270 degrees. Each of the structuring elements is applied and result is combined using logical OR to get the resultant image that identify endpoints.

0	1	0
0	0	0

Figure 39 Structuring element use for detecting end points

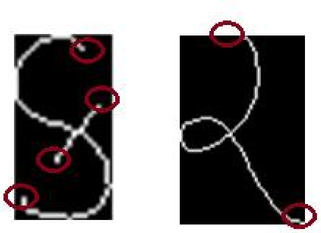


Figure 40 End points of Gujarati character “ક ka kə” and “ર ra rə”

To find the location of end points the character image is visualized into four quadrant and location of the endpoints are labelled with respect to quadrant number as top-left, top-right, bottom-left and bottom-right. Four more features are retrieved based on number of endpoints at top, bottom, left and right in the image. These make total nine features including total number of endpoints. The sample entries of nine features are shown in Table 4 for character “□ ka kə” and “□ ra rə”.

Table 4 Sample entry of features end points and its location

Character	Number of end points	Location with respect to four quadrants.				Location with respect to			
		Top-left	Top-right	Bottom-left	Bottom-right	Top	Bottom	Left	Right
ક	4	0	1	1	0	2	2	2	2
ર	2	1	0	0	1	1	1	1	1

6.2.2.2 Number of branch points (NB)

In character image if a pixel having more than two neighbours are considered as branch point. To locate the branch point again hit-and-miss transformation is used with different structuring element. The structuring element is used for branch point identification is depicted in Figure 41. Other nine structuring element is obtained by rotating each of these in 90, 180 and 270 degrees. Each of the structuring elements is applied and result is combined using logical OR to get the resultant image that identify branch points.

	1	
	1	
1		1

1		
	1	
1		1

	0	1
1	1	0
	1	

Figure 41 Structuring element used for branch point detection

After finding branch points of a thinned character image, total number of branch points are counted and added to the features vector. The branch points for character “□ ma mə” shown in Fig.

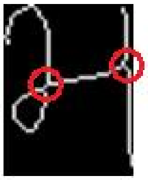


Figure 42 Branch points in Gujarati character " મ ma mə"

6.2.2.3 Number of loops (NL)

The loop is the close regions or holes formed in character shape. The morphological shrink is performed on thinned image to find loops in the character. It removes pixels so that objects without holes shrink to a point, and objects with holes shrink to a connected ring halfway between each hole.

The operation remains with more than one pixel if loop is present. Using morphological branch points operation, the number of loops in the character image is calculated. Number of loops is added to the features vector for the recognition.

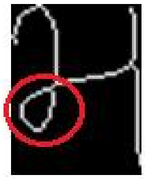
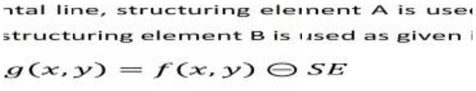


Figure 43 Loop in character " મ ma mə"

6.2.2.4 Length of vertical and horizontal line (LV, LH)

The shape of the character creates horizontal or vertical lines. To identify line in character image morphological erosion is used with line as structuring element. For detecting horizontal line, structuring element A is used as given in Figure 44 (a) and for vertical line, structuring element B is used as given in Figure 44 (b).



ntal line, structuring element A is use

structuring element B is used as given i

$$g(x, y) = f(x, y) \ominus SE$$

Equation 7

Where, f(x, y) is thinned character binary image, SE is structuring element used to probe character binary image and g(x, y) is the result of erosion operation.

1	1	1

a.

1
1
1

b.

Figure 44 a) Structuring element for horizontal line detection b) Structuring element for vertical line detection

After applying morphological erosion with vertical line structuring element, the resulting character image remains with vertical lines. The vertical projection is calculated for resulting image. The length of vertical line is obtained by maximum values of vertical projection after applying convolution operator of stroke width area. To normalize the value, number of line pixels is divided by character height. Similar steps are performed for horizontal line detection by changing structuring element to horizontal line. The steps for vertical line detection is presented here:

Algorithm:	VerticalLine
Input:	Binary thinned image, stroke width
Output:	VL feature

```
Perform erosion with 90˚ line structuring element
Calculate vertical projection of eroded image
Find maximum frequency of character pixels and its location
Apply convolution operation with mask size of stroke width
maxLinePixel = Find maximum value from convolution data
VL=maxLinePixel/character_height
END
```

6.2.2.5 Height to width ratio (HWR)

The character height to width ratio also known as aspect ratio is obtained by calculating character height divided by character width. This feature is useful due to unconstrained size of character, also note that in this work the size of character is not normalised or scaled to fix size.

6.2.2.6 Water reservoir area (WRA)

One of the novel features used in this work is water reservoir area (WRA). The water reservoir concept is used in [11,23,107] for detecting touching character in Oriya. The concept of water reservoir for character image is as follows. If water is filled from the top of the character image, the catchment regions of the character where water is stored are considered as the top reservoirs as depicted in Fig. 39.

The Gujarati characters are very peculiar in the nature and having this catchment area in different characters at different positions with different sizes. This motivates us to use it as feature in feature vector. The catchment area caused by character is used as feature of the character. To normalize the feature value the number of pixels of catchment region is divided by total number of pixels of an image i.e. height X width.

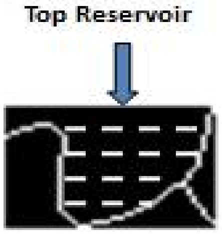


Figure 45 Top reservoir for character “પ pa pə”

To find catchment area simple yet robust technique is used. To find top reservoir, labelling mechanism is used in three steps for character image. In first step, starting from the top to bottom for each column labelled the pixel till the on pixel is found.

In second step, from left to right for each row start removing label till the on pixel founds. After the on pixel, all pixels are labelled up to next labelled pixel or on pixel. The same process is repeated for right to left direction.

In third step connected component analysis is performed on all labelled character image. The component with maximum number of pixels considered as catchment area. To normalize value of catchment area, it is divided by total area of the character image. This gives percentage of area covered by catchment region. The algorithm steps for the obtaining water reservoir area feature are as follows:

Algorithm:	getTopWRA
Input:	Binary thinned image
Output:	Top reservoir area

```
FOR each column in character image
    Assign label up to on pixel
END FOR
FOR each row in labeled image
    Remove labeled pixel up to on pixel from left
    Remove labeled pixel up to on pixel from right
    Assign label to pixel after labeled pixel up to on pixel in left direction
    Assign label to pixel after labeled pixel up to on pixel in right direction
END FOR
```

```
10.    Apply connected component analysis
11.    Top reservoir area=Find object with maximum area/ (height X width)
12.    END
```

Similarly, three more feature values are obtained as bottom reservoir, left reservoir and right reservoir. The reservoir statistics for some of the Gujarati characters and numerals are shown in Table 5.

Table 5 Reservoir area for Gujarati characters

Character	Reservoir area			
	Top	Bottom	Left	Right
ક	0.057937	0.047388	0.291137	0.19462
ખ	0.314705	0.108443	0.018025	0.025316
ગ	0.03431	0.023416	0.489738	0.000080
ઘ	0.415831	0.050098	0.009764	0.047098
ચ	0.275407	0.029503	0.129578	0.016208
૦	0.029509	0.115119	0.017145	0.081715
૧	0.008506	0.009988	0.075724	0.187632
૮	0.00527	0.001866	0.000723	0.379616
૯	0.085754	0.012129	0.001836	0.300334

6.2.2.7 Partial radial histogram (PRH)

Another novel feature used in this work is partial radial histogram (PRH). The partial radial histogram is based on radial histogram. The radial histogram also known as polar histogram is calculated using finding angle for each of the object pixel locations and sum up its pixels with the group of θ˚ angle [39,65] as shown in Figure 46.

The radial histogram is normally obtained from the centre of the character and can be processed for 0˚ to 360˚. The size of Gujarati handwritten characters are varies, so the centre of the character is going to be different for the same characters with small variation.

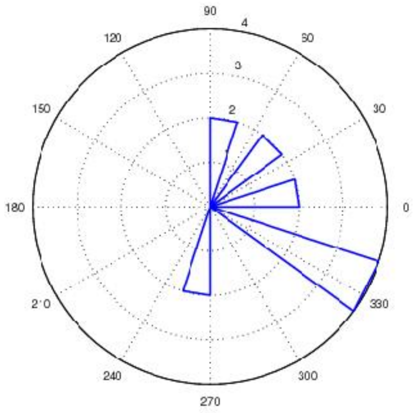


Figure 46 Radial histogram

To eliminate this issue, radial histogram is computed from the corner of the character image. Taking radial histogram from corner of the character image allows to process histogram for only up to 90˚ so we called it as partial radial histogram (PRH).

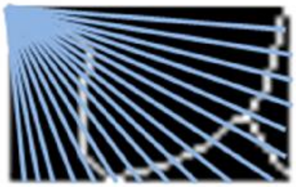


Figure 47 Partial radial histogram for character image ”પ pa pə” from top-left corner

The PRH is obtained with grouping 5˚ angle so total 18 feature values are obtained for top-left corner of the character image as shown in Figure 47. These 18 feature values of one PRH are normalized by dividing pixel frequency value by total number of object pixels. Similarly, three more PRH are obtained from bottom-left, top-right, and bottom-right corners as features.

6.2.3 Feature vector

It is accepted that one of the crucial factor influencing performance of handwritten character recognition is the selection of an appropriate set of features [65,138]. This allows different researchers to use different features for recognition of character which we have reviewed in section 2.2.

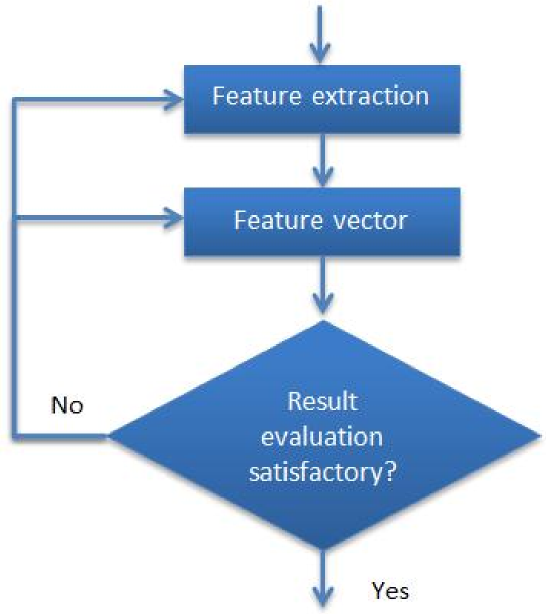


Figure 48 Feature vector selection process

The feature vector or pattern vector is a vector that contains the features that are extracted [138]. As described in earlier section different types of features are extracted which are relevant to recognise Gujarati character image. This feature are stored in feature vector and feed into the recognition system for classification. The choice of a good feature subset is crucial step in any classification process. The selection of features in feature vector is learned as shown in given Figure 48.

The features which are extracted earlier are modelled into 7 feature vectors by categorising features based on type and combining different set into one feature vector as shown in Table 6.

Table 6 Feature vector model

Set	Number of features	Feature description
Set1	14	Number of endpoints in a character image.
		The endpoint location with respect to four quadrant top-left, top-right, bottom-left and bottom-right along with number of endpoints at top, bottom, left and right.
		The number of branch points in character image.
		The number of loops in character image.
		Ratio of length of vertical line with respect to height of character.
		Ratio of length of horizontal line with respect to width of character.
		Ratio of height and width.
Set2	4	The water reservoir area in percentage for top, bottom, left and right of the character image.
Set3	72	The partial radial histogram (PRH) computed from top-left corner of image.
		The PRH computed from top-left corner of X - flipped image.
		The PRH computed from top-left corner of Y- flipped image.
		The PRH computed from top-left corner of 180˚ rotated image
Set4	18	Set1 + Set2
Set5	86	Set1 + Set3
Set6	76	Set2 + Set3
Set7	90	Set1 + Set2 + Set3

6.2.4 Recognition engine

The recognition and classification of a character based on the extracted feature is carried out using different methods by different researchers. The template matching methods, structural methods are not robust for handling variation of handwriting [6].

The neural network is configured for a specific application like pattern recognition, data classification, etc., through a learning process. It has great potential for parallelism as computations of the nodes are largely independent of each other. They can adapt to changes in the data and learn the characteristics of unknown inputs [4,76,137]. Hence it is suitable for building recognition engine for IGHCR. In the next section we will give brief introduction of neural network.

6.2.4.1 A Formal neuron

Artificial Neural Network (ANN) is popular statistical classification techniques widely used for classification in many areas [78]. It is based on human nervous system as shown in Figure 49. ANN is relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is divided into three stage receptors, effectors and the flow as shown in block diagram.

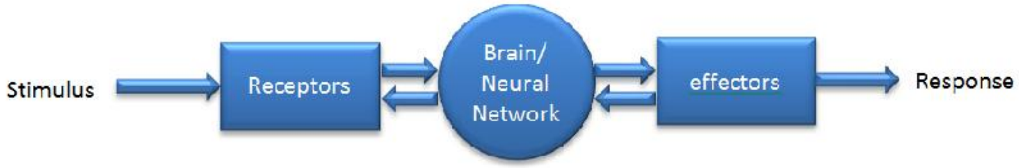


Figure 49 Human nervous system

The function of receptor is to collect information from the environment. The effectors generate interaction with environment and the flow of information or activation is feedforward or feedback. The human brain can be visualised as a large number of interconnected processing units and as activation function that fires when the neurons cross a threshold value.

A formal neuron, which is obtained by re-formulating a simplified function of biological neuron into a mathematical formalism, will be the basis of the mathematical model of neural network. Its schematic diagram of formal neuron is shown in Figure 50.

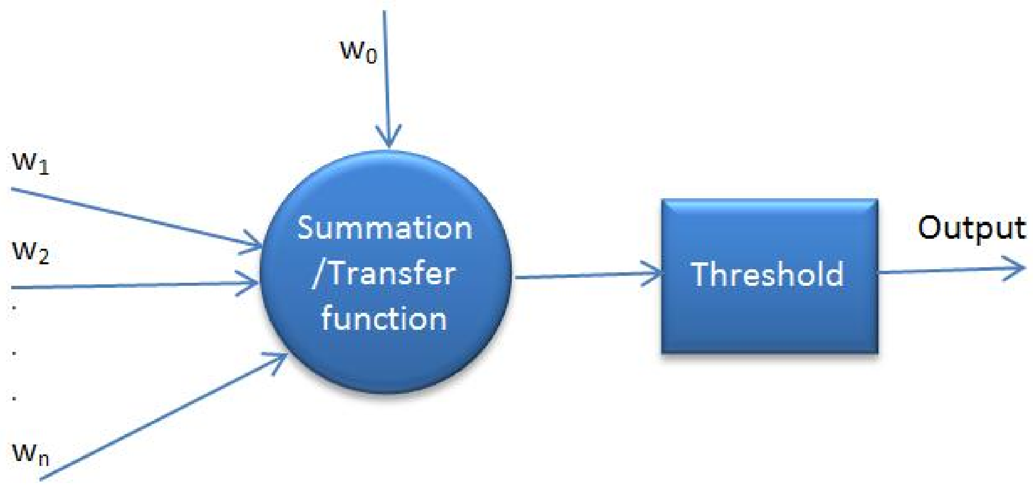


Figure 50 Schematic diagram of formal neuron

The formal neuron has n, generally real, inputs x_1, x_2,… x_n that model the signals coming from dendrites. The inputs are labelled with the corresponding, generally real, synaptic weights w_1, w_2… w_n. The transfer function of basic neural network can be described as $y = F(u)$ where:

$$u = w^0 + x^1 w^1 + x^2 w^2 + x^3 w^3 + \ldots \ldots \ldots x^n w^n$$ Equation 8

$$= w^0 + \sum_{i=1}^{n} x_i w^i$$ Equation 9

The neuron fires if at-least one of the following conditions is satisfied.

1. $\sum_{i=1}^{n} x_i w^i \geq T$ threshold value or
2. $f(u) \geq T$ for some non − linear activation function.

One can visualize an activation function as a neuron regulator and it work by ensuring that the responses of neuron are bounded, that is within the permissible range. Some of the useful activation functions that are widely used in ANN are listed as follows:

1. Identity function

$$f(x) = x \text{ for all } x \qquad \text{Equation 10}$$

2. Step function

$$f(x) \begin{cases} 1 \; if \; x \geq T \\ 0 \; if \; x < T \end{cases} \qquad \text{Equation 11}$$

3. Sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}} \qquad \text{Equation 12}$$

6.2.4.2 Neural Network

A neural network consists of formal neurons which are connected in such a way that each neuron output further serves as the input of generally more neurons similarly as the axon terminals of a biological neuron are connected via synaptic bindings with dendrites of other neurons. The architecture of neural network is determined by the number of neurons and the way they are interconnected.

Typical neural network architecture consists of three layers input, hidden and output based on their purpose. Input layer receives an external signal and simply transfers it to the neurons of the next layer without performing any operation. It represents receptors. In output layer, the neurons receive the signal from either the hidden layer or the input layer represents effectors. The hidden layer consists of connected working neurons create the corresponding channels between them to propagate the respective signals.

The way these neurons are interconnected distinguished in two types of architectures: cyclic (recurrent) and acyclic (feedforward) network. In the cyclic topology, there exists a group of neurons in the network, which are connected into a ring cycle. The simplest cycle is a feedback of the neuron whose output serves simultaneously as its input. The maximum number of cycles is contained in the complete topology in which the output of each neuron represents the input for all neurons. An example of a general cyclic neural network is depicted in Figure 51 a where all the cycles are indicated.

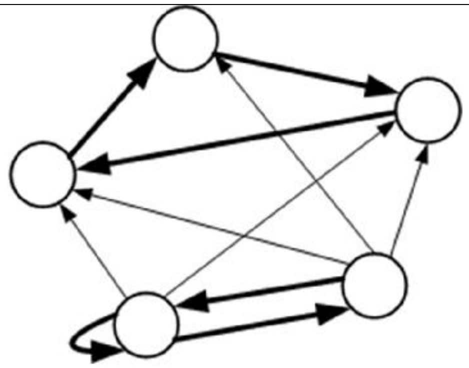


Figure 51 Cyclic architecture

On the contrary, the feedforward neural networks do not contain any cycle and all paths lead in one direction. An example of an acyclic neural network is in Figure 52 b where the longest path is marked.

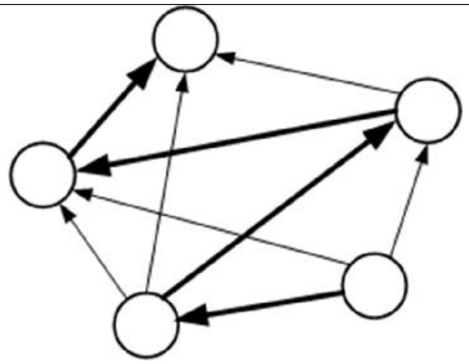


Figure 52 Acyclic architecture

6.2.4.3 Multilayered Feedforward Neural Network

Feed-forward neural network the connection amongst the neurons lead only one direction which allow signals to travel one way only; from input to output. The output of any layer does not affect that same layer. Feed-forward neural network tend to be straight forward networks that associate inputs with outputs.

Typical feedforward neural network architecture consists of three layers input, hidden and output as shown in Figure 53. Input layer receives an external signal and simply transfers it to the neurons of the next layer without performing any operation. In output layer, the neurons receive the signal from either the hidden layer or the input layer. The hidden layer connects the input and output layers.

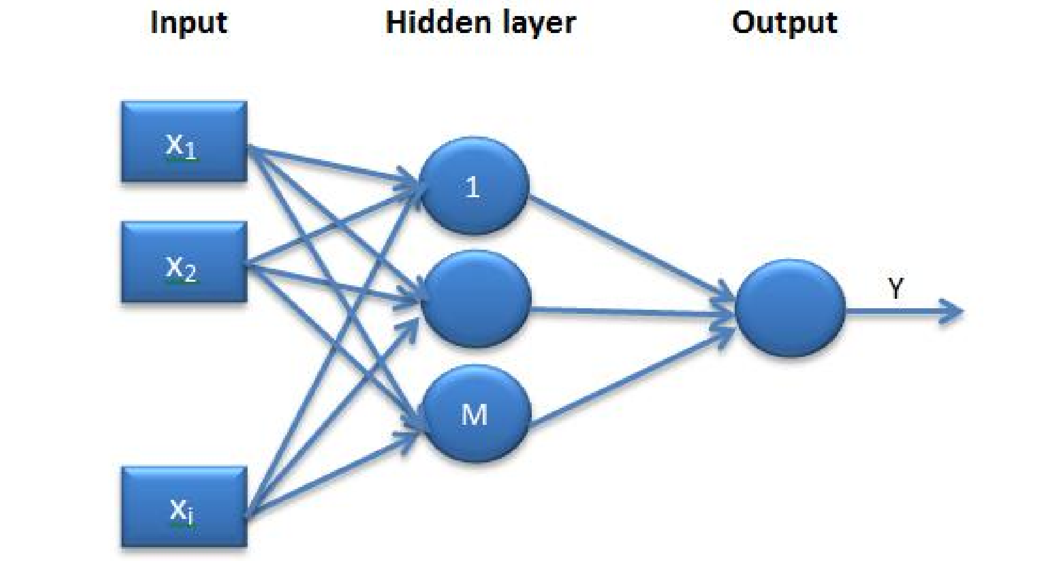


Figure 53 A typical feedforward neural network

Based on the number of layers, neural networks can be classified as single and multilayer networks. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech and language processing, vision, and control systems.

In the topology of a multilayered network, each neuron in one layer is connected to all neurons in the next layer. Therefore, the multilayered architecture can be specified only by the numbers of neurons in particular layers, typically hyphened in the order from input to output layer. In addition, any path in such a network leads from the input layer to the output one while containing exactly one neuron from each layer.

The most prominent and widely applied model of neural network is the multi-layered neural network i.e. feed forward with backpropagation learning algorithm. It is used in many neural network applications. This model generalizes the network of perceptron for the architecture with hidden layers.

An important characteristic of the ANN is its learning ability. The adjustment of weights based on the input data is known as learning in neural networks. The learning schemes are supervised learning or unsupervised. In supervised learning the target is known. The neural network computes the output and compares it with the target response. The error computed using the difference between the target and the actual output. Based on error the adjustment factor to the weights used to matches the target value as shown in Figure 54.

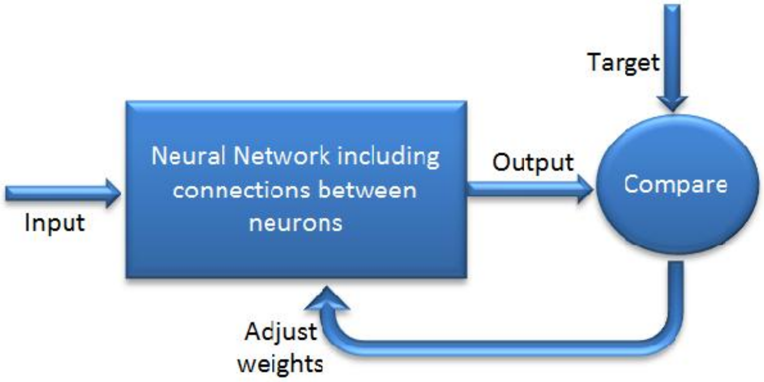


Figure 54 Training in neural network using supervise learning

In unsupervised learning, desire output is not known. The input vectors are grouped based on similarity. The test pattern is taken by the ANN and the output response indicates the class to which the input pattern belongs to.

The important parameters of the neural network involve correct decision of neural network architecture, the number of nodes in each layer, weighted functions, and appropriate learning techniques for adjusting the weights during iterations.

The neural networks are robust to the handwritten variations and learn from the stable features over a large training set [65]. So researcher decided to use two layers feed forward neural network as recognition engine in IGHCR for classification of Gujarati characters which is effective for pattern recognition problems.

A two-layer feed-forward network with sigmoid hidden and output neurons, can classify vectors arbitrarily well, given enough neurons in its hidden layer. The network is trained with scaled conjugate gradient backpropagation.

The architecture diagram of two-layer feed forward neural network is depicted in Figure 53. The neural network is trained with different set of feature vector obtained for each character in the training set. The set of feature vectors are shown in table 6.

The network randomly divided 13500 input samples into training, validation and testing set in the ratio of 70% (9450), 15% (2025) and 15% (2025) respectively. The training set is used for training, validation set are used to validate that the network is generalizing and to stop training before over fitting. The testing set use to test network independently to check network generalisation.

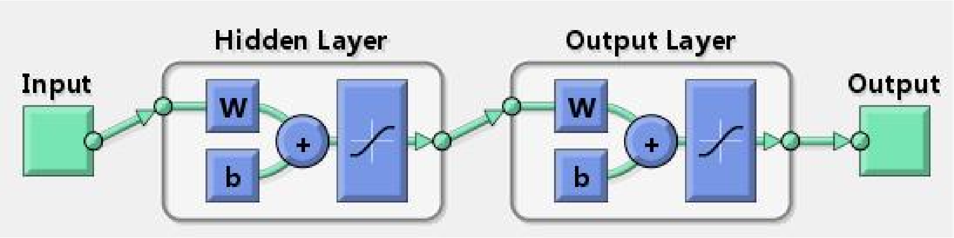


Figure 55 Two layer feed forward neural network

The input neuron of neural network is set to number of input element i.e. number of features in feature vector. The number of output neuron is set to the number of output class. The hidden layer neuron, if set to appropriate number gives good results.

6.2.4.4 Using trained model for core character recognition

The trained network obtained from the IGHCR is used to recognize core character which is segmented by segmentation phase of OGHTR system. The network accept feature vector for recognition of character. Hence, feature extraction process need to be done for segmented core character.

The feature is extracted on preprocessed character, which is thinned binary image in IGHCR. The segmented core character is binarized and thinned but required remaining process of preprocessing those are removal of extended line. So extracted core character is processed for it and then features are extracted from it.

This extracted feature vector is inputted into the trained network gives output as best matches with character class. The best match class is selected as recognized character.

6.3 TEXT GENERATION

The main objective of text generation is to combine the result obtained from diacritic mark recognition and core character recognition and generate text. It is essential also due to it need transformation from recognize character code to Unicode characters.

Also as per our approach of segmentation we segmented each unit as connected component. So it is possible that two diacritic marks are identified separately for “◌▯ o, ɔ kano ek matra” and “◌▯ Au əʊ kano be matra” vowels. That means for “◌▯ o, ɔ kano ek matra” the diacritic mark extracted as right matra “◌▯ ā ɑ̈ kāno” and top matra “◌□e, ɛ ek mātra” separately. Similarly, for “◌▯ Au əʊ kano be matra” the diacritic mark extracted as right matra “◌▯ ā ɑ̈” kāno” and two single top matra “◌□e, ɛ ek mātra” detected separately. So recognized classes for “◌▯ o, ɔ kano ek matra” are two and for “◌▯ Au əʊ kano be matra” are three. In Unicode these diacritic mark represent only one Unicode, hence conversion is required.

During the recognition of characters, rules for characters “□ ṇa ɳə”, “□ la lə”, “□ śha ʃə” and “□ ga gə” are constructed for recognition as they occurs in more than one components. Also, we recognize them using first component only and then checking for rest of the component. This requires to do not consider rest of the part for conversion in Unicode. And if other part not found in that case it may represent conjunct. For example, if character “□” is present in the text then it is recognized by first part of it, and right diacritic mark presence is verified. If right diacritic mark is absent then it represent conjunct. To make it conjunct “◌”□is added after the Unicode of character ““□ śha ʃə”. If Therefore, text generation rules are created. They are:

- If recognized character is “ક્ષ kṣa kʃə” and “ gña gnə” then converted into Unicode character □ + ◌□+ □ and □ + ◌□+ □ respectively.
- If recognized character is first part of “□ ṇa ɳə”, “□ la lə” then next right component is ignored if it is “□ ga gə” or second part of “□ la lə”.
- If recognized character is first part of “□ śha ʃə” and “□ ga gə” then do not consider right diacritic mark “◌l ɑ̈ kāno” if available. Otherwise add “◌□”after the character to make it as conjunct.
- If recognized character is second part of “□ la lə” and previous recognized character is “□ ra ɾə” or numeral “□bagaṛo or bay” then consider it as “□ sa sə”.
- If recognized characters are from numeral “□ bagaṛo” or “□ pāchaṛo” and have diacritic marks then consider it as alphabet “□ ra ɾə” or “□ pa pə” character.
- Consider recognized character as “□ ra ɾə” or “□ pa pə” as numeral if they appear single as word or previous or next characters are numerals.
- If character followed by diacritic mark “◌□ Ā ɑ̈ kāno” and “◌□e, ɛ ek mātra” then convert it to “◌□ o, ɔ kāno ek mātra”.
- If character followed by diacritic mark “◌□ Ā ɑ̈ kāno” and “◌□Ai əj be mātra” then convert it to “◌□ Au əʊ kāno be mātra”.
- If character followed by diacritic mark “◌□e, ɛ ek mātra” and “◌□e, ɛ ek mātra” then convert it to “◌□Ai əj be mātra”
- If character followed by diacritic mark “◌□ Ā ɑ̈ kāno” and two “◌□e, ɛ ek mātra” then convert it to “◌□ Au əʊ kāno be mātra”.
- If left diacritic mark “□◌ I i hrasva-ajju” is recognized then place it after the recognized core character.
- If recognized character is alphabet “□ A ə” and has diacritic mark “◌□ ɑ̈ kāno” / “◌□e, ɛ ek mātra” / “◌□Ai əj be mātra” / “◌□ o, ɔ kāno ek mātra” / “◌□ Au əʊ kāno be mātra” / “◌□ṁ anusvār” then convert it to “ □ Ā ɑ̈” / “□ e, ɛ” / “□ Ai əj” / “ □ o, ɔ” / “ □ Au əʊ” / “□ ṁ”

6.4 DISCUSSION

The recognition process of OGHTR is divided into two parts i) diacritic mark recognition and ii) core character recognition. For diacritic mark recognition decision tree classifier is used as number of diacritic marks is less. To recognize core character IGHCR module is implemented to train the neural network which is capable of classifies 54 character classes which includes Gujarati alphabet, numerals and few more symbols.

To train the neural network dataset consists of 250 character image for each class is used. The character image is preprocessed and feature extraction is carried out for recognition. Two novel features are presented in this research they are water reservoir area and partial radial histogram.

The two layers feed forward back propagation neural network is used for classification using extracted features. The different feature vector has been created for finding the best selection amongst the features. In next chapter result of recognition phase is discussed along with the result of segmentation phase.

To recognize conjunct is not in the scope of research but through experiment we are able to recognize conjunct whose first character is “□ śha ʃə” and “□ ga gə”. This is due to the recognition process is train with half form of these characters. This encourages us to add recognition of conjunct character in future work.