ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2016.0169

FEATURES BASED ON NEIGHBORHOOD PIXELS DENSITY - A STUDY AND COMPARISON

Satish Kumar
Department of Computer Science and Applications, Panjab University Swami Sarvanand Giri Regional Centre, India
E-mail: satishnotra@yahoo.co.in

Abstract
In optical character recognition applications, the feature extraction method(s) used to recognize document images play an important role. The features are the properties of the pattern that can be statistical, structural and/or transforms or series expansion. The structural features are difficult to compute particularly from hand-printed images. The structure of the strokes present inside the hand-printed images can be estimated using statistical means. In this paper three features have been purposed, those are based on the distribution of B/W pixels on the neighborhood of a pixel in an image. We name these features as Spiral Neighbor Density, Layer Pixel Density and Ray Density. The recognition performance of these features has been compared with two more features Neighborhood Pixels Weight and Total Distances in Four Directions already studied in our work. We have used more than 20000 Devanagari handwritten character images for conducting experiments. The experiments are conducted with two classifiers i.e. PNN and k-NN.

Keywords:
Statistical Features, Hand-Printed Recognition, Devanagari Script, NPW (Neighborhood Pixels Weights), SND, LPD, RD, k-NN, PNN, Weighted Map

1. INTRODUCTION

An optical character recognition functions in different stages like scanning, pre-processing, segmentation, feature extraction, classification and post-processing. Each stage has a specific task to do. Among them, the feature extraction stage is a crucial stage as some useful features of an image is extracted during this stage. The given images are categorized into their respective classes using these features. The features used to represent the given image must be robust for classification purpose. Robust means, the minor variation present in a hand-printed character image due to writing or pen used is coped by feature itself. If we look at literature, there are many features used to express the images. According to Govindan et al [1], there are three categories of features i.e. statistical, structural and global transforms and series expansion.

In case of document, the character images are either black and white or gray. In structural features the geometrical and topological properties of a character image are used. In these, the various stroke primitives, junction points along with their types, present in a character image, are located. The relationships among these primitive components are set up. The approach requires multiple heuristics as it is based on matching the approximated primitive components with store prototypes. So, this is a complex approach. The structural features can be extracted from binary images only and are estimated from skeletonised or contour character images. The problems with skeletonization process are that it is computationally costly and sometimes it also removes some significant components of the character images. The difficulties arising due to structural features can be easily covered with statistical ones. On the other hand, the statistical features use the statistical distribution of B/W pixels in the given image. Such features can be estimated from the binary as well as the gray scale images. The features are not much variant to character deformation and writing variations. The features are not difficult to estimate. Moreover, these can be estimated with high speed as only a few arithmetic or logic functions requires to perform at each pixel that takes very small computational efforts. The various statistical feature extraction methods are available in literature. Each method has a unique way to estimate the distribution of B/W or gray pixels in a given character image. The overall purpose is to spot the existence of various stroke primitives in the given zone of an image. Some statistical features available in literature for recognition are zoning, projection profile, projection histograms, crossings, character loci, moments and n-tuple. The statistical features can be local or global [1]. Baird [2] and Anisimovich [3] used binary feature set for expressing topological or geometrical properties. In both cases, the existence of a particular stroke or topology in a particular zone is spotted and expressed through binary feature. This kind of representation gives a lot of deviation in feature location and thus produces discontinuity in feature space.

In case of off-line hand-printed character recognition, a comprehensive review is available in literature [10-14]. In this study, we are proposing three new features which are based on neighborhood pixels of a pixel in a character image. The features are statistical and different from each other the way we consider the neighborhood pixels of a pixel for extracting their properties. The methods of extracting features have been discussed in detail. The practical recognition performance of each feature on noisy as well as noise-less images has been investigated. The new features are Spiral Neighbor Density (SND), Layer Pixel Density (LPD) and Ray Density (RD). In addition to this two more features i.e. Total Distance in four Directions (TDIST) and Neighbor Pixels Weight (NPW) have been discussed. The Neighborhood Pixels Weight (NPWM) is modified version of NPW [5]. The TDIST feature has been already discussed [7] but its performance has not been compared individually. These features can be used as individual as well as supporting features in many applications depending upon the complexity of script. The proposed features can be extracted both from B/W and from gray images. In our earlier study we have compared the performance of about five features using two classifiers, the NPW feature was considered as best [5]. In consonance with earlier results we are comparing the performance of features under study with NPW features already published. There might be a question, why only these features have been studied and compared in this paper? The answer is that all these features are based on the neighborhood pixels of a pixel. The difference lies in the way they are extracted. It is essential to know that, which will be a better way of extracting properties of an image using neighborhood pixels for discrimination purpose.

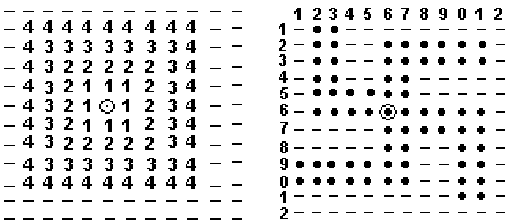
The detail experimental results of proposed features and their performance in comparison with other features is provided. The details of newly proposed features as well as existing, on which this study is conducted, are given in section 2 and section 3 covers the classifiers used to classify. The experimental results are covered in section 4. The experimental results are analysed in section 5. The research conclusion is given in section 6.

2. FEATURES BASED ON NEIGHBORHOOD PIXELS

The simplest form of a feature is pixel based where a normalized image of size N×N is fed to the classifier. In this case the size of the feature set is very large that is not only difficult to trained but computationally time consuming too. The second simplest method is zoning where a normalized image of size N×N is partitioned into various zones and the percentage of B/W pixels in each zone is computed and fed to the classifier. The feature size is equal to the number of zones created in a character image. The method is fast and easy to implement. In the proposed features, it is not only some statistical measures that has been considered but the neighborhood pixels of a pixel are also considered. This encodes the hidden characteristics of a character image that helps in taking classification decision efficiently.

2.1 LAYER PIXEL DENSITY (LPD)

Any pixel contains 8, 16, 24 and 32 neighboring pixels in its first, second, third and fourth level neighboring layers respectively. The four neighborhood levels are marked as ‘1’, ‘2’, ‘3’ and ‘4’ for first, second, third and fourth level neighborhood layers respectively as given in Fig.1(a). In LPD the average density of black pixels in each neighboring layer pertaining to a pixel under consideration is computed. Since first, second, third and fourth layer consists of 8, 16, 24 and 32 pixels respectively. The average density of black pixels in first layer is computed out of 8 pixels, in second layer is computed out of 16 pixels, in third layer is computed out of 24 pixels and in fourth layer is computed out of 32 pixels.



(a) (b)

Fig.1.(a) Four neighborhood levels marked as ‘1’, ‘2’, ‘3’ and ‘4’, (b) A Black and White image of size 12×12 pixels

For a B/W image, Fig.1(b), the average density of black pixels in first, second, third and fourth layer for pixel (6, 6) is 7/8 = 0.875, 8/16 = 0.5, 15/24 = 0.625, and 24/32 = 0.75 respectively. The image is considered as rolled over while considering neighborhood layers for a pixel on border or near border. For example in case of Fig.1(b), row 1 is considered after row 12 and row 12 is considered after row 1. Similarly with column pixels i.e. column 1 is considered after column 12 and column 12 is considered after column 1. The average density of black pixels on a pixel near boarder say (2, 6), read as row 2 column 6, in first, second , third and fourth layer is 3/8 = 375, 4/16 = 0.25, 11/24 = 0.458, and 21/32 = 0.656 respectively.

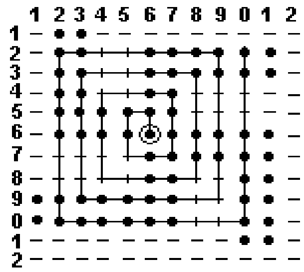


Fig.2(a). Spiral Neighbors of pixel (6, 6) in a binary image

The feature can be computed for a gray image. For a gray image, Fig.2(b), the average density of the pixels for first layer neighborhood pixels for pixel (6, 6) shall be 200 + 220 + 80 + 210 + 240 + 220 + 200 + 240 = 1610/(8 × 255) = 0.789. Similarly, the average density of pixels for second, third and fourth layer neighbour pixels for pixel (6, 6) shall be 0.62, 0.565 and 0.672 respectively.

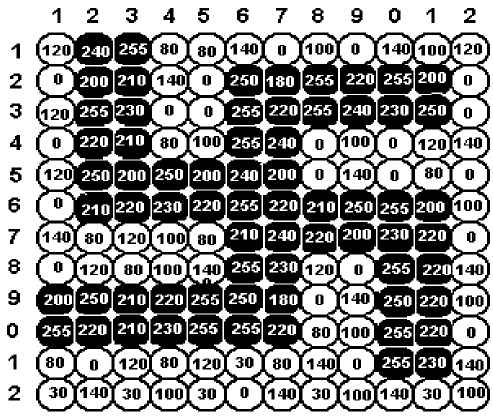


Fig.2(b). Gray image of size 12×12 pixels

For LPD, we have conducted our study on binary images only. Three layers per pixel have been considered i.e. 48 pixels per pixel. The feature is represented using three cells per pixel where each cell stands for a particular neighborhood pixel layer. A pixel density map (PDM) for various pixels in a given image is developed. The PDM contains three planes, each plane has same size as it is the size of the given normalized image. Each plane is due to average density of black pixels for the particular layer of neighborhood pixels for entire image. To extract feature set, each PDM plane is partitioned into X×Y zones and average density of black pixels out of total pixels in each zone is computed. The feature set size is 3×5×5.

In case of LPD, we studied two features i.e. LPD1_75 and LPD2_75. In LPD2_75, the pixel density has been computed for foreground pixels only and pixel density on background pixels at all the three level shall be zero whereas in LPD1_75 the pixel density has been computed for both foreground and background pixels.

2.2 SPIRAL NEIGHBOR DENSITY (SND)

In SND feature, the pixels used to calculate the features is given in Fig.2(a). In example the sequence of pixels for a given pixel (6, 6) shall be: (5, 6), (5, 5), (6, 5), (7, 5), (7, 6), (7, 7), (6, 7), (5, 7), (4, 7), (4, 6), (4, 5), (4, 4), (5, 4),....,(3, 9). In this way 48 pixels per pixel are considered to extract feature and partitioned into four bins of 12 pixels each. The average density of black pixels out of total pixels in each bin is computed. For pixel (6, 6) the pixels considered in each bin and their average density is given in Table.1.

Table.1. Pixels used in each bin at each pixel for a pixel (6, 6) in SND

Bin No.	Desired Pixels	Average Density
1	(5,6), (5,5), (6,5), (7,5), (7,6), … … (8,8)	13/20 = 0.65
2	(7,8), (6,8), (5,8), (4,8), (3,8) , … … (9,7)	14/20=0.7
3	(9,8), (9,9), (8,9), (7,9), (6,9), … … (6,2)	13/20 = 0.65
4	(7,2),(8,2),(9,2),(10,2),(10,3) … … (2,10)	14/20=0.7

The SND feature can also be computed for gray images. The average density of pixels in 1st, 2nd, 3rd and 4th bin for a gray image given in Fig.2(b) shall be 0.727, 0.654, 0.678 and 0.735. The image is considered as rolled over while considering spiral neighbor for a pixel on border or near border.

We have conducted our study on binary images only. A spiral neighbor density map (SNDM) for various pixels in a given image is developed. The SNDM contains four planes, each plane has same size as it is the size of the given normalized image. Each plane is due to average density of black pixels for a particular bin for all the pixels in an image. To extract feature set, each plane is partitioned into X×Y zones and the average density of black pixels out of total pixels in each zone is computed. In our study, we have partitioned each SNDM plane in 5×5 zones.

In case of SND, we studied two features i.e. SND3_100 and SND2_100. In SND3_100, the pixels of three neighborhood layers have been considered for computing density of black pixels and partitioned into four bins of 12 each where as in SND3_100, the pixels of two neighborhood layers have been considered for computing density of black pixels and partitioned into four bins of 6 each. In both cases the pixel density has been computed for both foreground and background pixels.

2.3 RAY DENSITY(RD)

In Ray Density (RD) feature, the density of black pixels along four tracks i.e. horizontal, vertical, left-diagonal and right-diagonal as shown in Fig.3 is calculated. The middle pixel is not considered in exercise. The density of black pixels along horizontal, vertical, left-diagonal and right-diagonal direction considering 8 pixels in each direction, for pixel (6, 6) for B/W image given in Fig.3 is 8/8 = 1.0, 8/8 = 1.0, 5/8 = 0.625 and 5/8 = 0.625 respectively. The number of pixels in each direction can be increased or decreased depending upon need or optimal value. The Ray Density feature can be calculated for a gray image too. The density of black pixels along horizontal (H), vertical (V), left-diagonal (LD) and right-diagonal (RD) direction considering 8 pixels in each direction, for pixel (6, 6) for gray image given in Fig.2(b) is 0.89, 0.97, 0.72 and 0.64 respectively. The details are given in Table.2.

Table.2. Gray level of each pixel in respective bins along with computed average density in each for RD

Bin No.	Desired Pixels	Average Density
H	210, 220, 230, 220, 220, 210, 250, 255	1815/(8×255) = 0.89
V	250, 255, 255, 240, 210, 255, 250, 255	1970/(8×255) = 0.97
LD	200, 80, 230, 200, 240, 120, 140, 255	1465/(8×255) = 0.72
RD	200, 0, 240, 255, 80, 100, 210, 220	1305/(8×255) = 0.64

We have conducted our study on B/W images only. A Ray density map (RDM) for various pixels in a given image is prepared. The RDM contains four planes, each plane has same size as it is the size of the given normalized image. Each plane is due to average density of black pixels for a particular bin for all the pixels in an image.

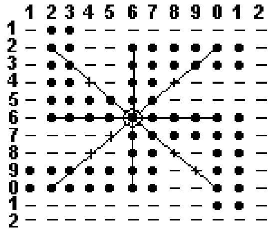


Fig.3. B/W image with horizontal, vertical, left-diagonal and right-diagonal tracks for pixel (6, 6)

To extract feature set, each RDM plane is partitioned into X×Y zones and average density of black pixels out of total pixels in each zone is computed. In our study, we have partitioned each RDM plane into 5×5 zones.

In case of RD, we studied four features i.e. RD10_1_100, RD10_2_100, RD12_1_100 and RD12_2_100. In case of above mentioned features, the number 10 and 12 means 10 and 12 pixels respectively used along each direction to compute black pixels density. The value _1 means the pixel density has been computed for foreground pixels only whereas _2 means the pixel density has been computed for both foreground and background pixels.

2.4 MODIFIED NEIGHBORHOOD PIXELS WEIGHT (MNPW)

In Neighborhood Pixels Weight (NPW) [5], the weights of neighboring pixels of a pixel are considered for feature computation as shown in Fig.4(a). The Figure shows, neighboring pixels of a pixel (6, 6) on four corners, due to first, second and third level neighborhood layers. The pixels which are on horizontal or vertical tracks marked in dotted rectangle, neighboring the pixel under consideration, are not considered for feature calculation. In Modified Neighborhood Pixels Weight (MNPW) all such pixels are also considered in balanced manner as given in Fig.4(b). In this study, the pixels of two level as well as the three level neighborhood layers are considered.

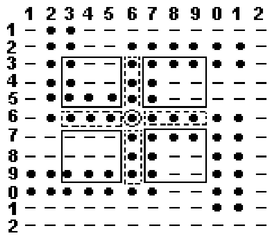


Fig.4(a). Neighboring pixels considered for a given pixel (6, 6), in case NPW feature

In this study, the weight on a pixel due to black pixels is considered. For a pixel, the feature set is represented with 4-cell array as explained in [5]. The NPWM weight for a pixel (6, 6), Fig.4(b), is given in Table.3.

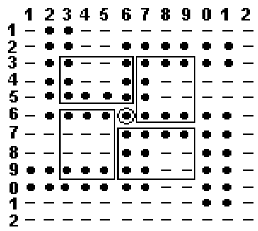


Fig.4(b). Neighboring pixels considered for a given pixel (6, 6) in case NPWM feature

The MNPW feature can be extracted from gray images as it is mentioned in [5]. Our study is carried on B/W pixels only. A modified weight map (MWM) for various pixels in a given image is developed. The MWM contains three planes, each plane has same size as it is the size of the given normalized image.

Table.3. Weights of neighboring pixels on four corners

Corner	Top Right	Top Left	Bottom Left	Bottom Right
Weights	8/12=0.75	8/12=0.75	8/12=0.75	6/12= 0.5

In our experiments we have partitioned each MWM plane in 5×5 zones.

We have conducted our study by considering two level as well as three level neighborhood layer pixels. The purpose is to know recognition ability of MNPW feature by considering two levels as well as three levels.

In case of both NPW and NPWM, we studied four features from each. In case of above mentioned features, the value 2 and 3 means 2 level and 3 level neighborhood pixels respectively have been used for feature computation. The value_1 means, the pixel density has been computed for foreground pixels only whereas_2 means the pixel density has been computed for both foreground as well as background pixels.

2.5 TOTAL DISTANCES IN FOUR DIRECTIONS (TDIST)

In DDD based feature [6], if we take the distance travelled by a ray in all 8-directions and encode feature set using 16-cell per pixel and subsequently divide each DDD plane into 4×4 zones, the size of feature set grows to 16×4×4 = 256. The feature vector is large in size. If we combine the same with another feature it will further grow in size. For combination purpose, the DDD feature is restructured to smaller size. Here, we take total distances in four directions and the feature is represented using 4-cell per pixel. The new feature is named as TDIST (Total Distances in Four Directions) [15]. In Fig.2(a), the total distances for a pixel (6, 6) along horizontal, vertical, left-diagonal, and right-diagonal directions are 10, 9, 3 and 2, respectively. In this case the distance considered is from a black to whites.

According to all pixels in an image there are 4 TDIST planes. The size of each TDIST plane is same as the size of original image. Each TDIST plane is partitioned into 5×5 zones giving 4×5×5 = 100 sized feature set. Here we extracted two features i.e. TDIST1_100 and TDIST1_100. In TDIST2_100, the distances of the foreground pixels to background pixels are computed only whereas in case of TDIST1_100 the distances of the background pixels to foreground pixels have been also considered. In TDIST2_100 total distances of background pixels will be zero in all 4-cell according to a pixel.

3. CLASSIFICATION METHODS

We have conducted various experiments in this paper on hand-printed characters. The various classifiers used are: PNN (Probabilistic Neural Networks), *k*-NN (Nearest Neighbor). The hand-printed character set belongs to Devanagari script. The classifiers used are available in MATLAB.

3.1 *k*-NEAREST NEIGHBORS CLASSIFIER

The *k*-nearest neighbour (*k*-NN) is an instance-based classifier in which the posterior probability of occurrence of foreign pattern is determined based on the rate of occurrence of its nearest *k* neighbors in the training sample set.

To classify a new sample u, the k samples, close to u, from the training sample set are identified. The new sample u is assigned the label of samples that appears most frequently out of k samples. The Euclidean distance $DE(u, u_i) = \|u-u_i\|$ between an unlabeled test sample u and a labeled training sample u_i is given as,

$$DE(u,u_i) = \sqrt{\sum_{j=1}^{n} u_j - u_{ij}} \tag{1}$$

where, n is the size of feature set of a sample pattern under consideration. *k*-nearest neighbors of an unlabeled test sample mean out of total training samples, k samples which have minimum value of Euclidean distance from an unlabeled sample are used for decision making.

3.2 PROBABILISTIC NEURAL NETWORK

In Bayes classifier, a new pattern u is assigned a class such that the decision function $\Psi_j(u)$ yields the maximum value [8].

$$\Psi_j(u) = p(u/\omega_j)P(\omega_j)\cdot\ j = 1,2,3,\ldots,q, \tag{2}$$

Probabilistic neural network (PNN) works on the basis of the Bayesian classification theory [9]. The said theory is implemented just like neural network. The output of this classifier is purely based on Bayes rule 2.

The bias value for first hidden layer is set to 0.8326/Sp. The parameter Sp mentioned here is called as Spread. This gives radial basis functions that cross 0.5 at weighted inputs of +/− Sp. This determines the width of an area in the input space to which each neuron responds. The response to the new unknown pattern wholly depends upon the Sp value and number of training samples. When Sp ≈ 0, the network acts as a nearest neighbor classifier. As Sp grows, the network considers several nearby design sets. We have conducted our experiments using various values of Sp. The training set size is fixed for various values of Sp.

4. EXPERIMENTAL RESULTS

The various experiments are conducted to check the performance of LPD, SND and RD. The experimental study is also made to know their performance in comparison to NPW, MNPW and TDIST features available in literature. The experiments are conducted on Devanagari hand-printed character dataset [7, 15]. The size of dataset used is 500 characters per class for 43 Devanagari alphabet letters (classes) only. Out of these first 100 characters from each class have been used for testing purpose and remaining 400 characters from each class have been used for training purpose. Both noise-less and noisy images are considered in study. Java codes are used for feature extraction. Images have been size normalized to 35×35 pixels. To know the effect on features in noisy situation, a random white noise which is about 10-15% to the normalized image size is added to the images. Some hand-printed Devanagari character images before and after noise are given in Fig.5.

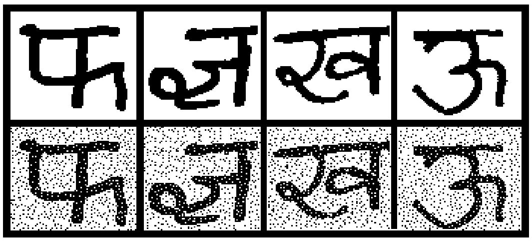


Fig.5. Devanagari handwritten character images before and after introducing random noise

4.1 FEATURE COMPUTATION STRATEGY

In section 2, we have discussed various features to be considered for examining. In case of all the features under study i.e. LPD, SND, RD, NPW, MNPW and TDIST, each weighted map (WM) is partitioned into 5×5 zones and average density of weight in each zone is calculated. The feature set is normalized by dividing the each feature with the maximum value of feature out of all the zones.

Table.4. Recognition results of different features using *k*-NN classifier on Devanagari handwritten character images

Feature Type	Devanagari Hand-printed Character Samples (Recognition Rate in %)									
	Without Noise					With Noise				
	1	3	5	7	9	1	3	5	7	9
NPW2_1_100	82.3	84.1	84.7	84.6	84.8	79.0	81.1	82.5	83.2	83.0
NPW2_2_100	74.7	77.0	78.0	79.0	78.8	73.6	75.7	76.9	77.6	77.5
NPW3_1_100	81.7	83.0	84.2	84.5	84.3	78.8	81.1	82.0	82.8	83.0
NPW3_2_100	76.9	78.8	80.0	80.1	80.2	74.5	76.4	77.8	78.7	78.7
NPWM3_1_100	82.2	83.9	84.7	85.1	84.7	79.1	81.3	82.8	82.8	83.0
NPWM2_1_100	81.9	83.9	84.3	84.5	84.7	78.9	81.2	82.2	82.6	82.1
NPWM2_2_100	75.7	77.8	78.6	79.2	79.0	72.1	74.8	76.5	76.4	76.6
NPWM3_2_100	76.4	77.9	79.4	79.6	79.6	73.3	76.0	77.7	77.9	78.2
RD101_100	79.3	81.3	82.2	82.0	81.8	74.8	77.2	78.5	78.7	78.8
RD121_100	79.4	81.6	82.7	82.5	82.1	75.7	77.9	78.9	79.3	79.5
RD102_100	79.7	81.7	82.7	83.7	83.3	76.1	79.0	80.3	81.4	80.7
RD122_100	79.8	81.6	82.7	82.7	82.6	75.8	78.7	80.3	80.6	80.6
LPD2_75	77.7	79.5	81.3	81.6	81.0	72.6	75.2	77.1	78.7	77.9
LPD1_75	72.0	74.1	75.2	75.6	76.1	68.3	71.3	73.2	73.7	73.2
SND3_100	80.6	82.4	83.3	84.4	83.6	77.1	80.8	80.8	80.9	82.2
SND2_100	81.2	83.3	83.2	84.5	83.5	77.6	79.5	80.8	81.1	82.3
TDIST1_100	74.0	76.3	77.0	76.8	76.6	56.6	59.7	62.0	62.4	63.6
TDIST2-100	75.3	77.1	78.4	78.7	78.5	63.0	66.7	69.6	70.5	70.1

Table.5. Recognition results of different features using PNN classifier on Devanagari handwritten character images

Feature Type	Devanagari Hand-printed Character Samples (Recognition Rate in %)									
	Without Noise					With Noise				
	0.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
NPW2_1_100	83.0	84.2	85.0	84.9	83.6	80.1	81.8	83.1	82.5	80.0
NPW2_2_100	75.7	77.4	78.7	78.6	77.4	74.4	76.2	77.3	77.6	76.2
NPW3_1_100	82.4	83.4	84.4	84.3	82.9	80.2	81.8	82.4	81.3	78.8
NPW3_2_100	77.8	79.3	80.0	79.8	78.0	75.6	77.5	78.3	77.7	76.1
NPWM3_1_100	83.1	84.1	85.1	85.1	83.3	80.5	82.2	83.1	81.8	79.1
NPWM2_1_100	82.7	83.7	84.7	84.7	83.3	79.9	81.5	82.4	81.9	79.4
NPWM2_2_100	76.6	77.7	78.9	79.1	77.1	73.1	74.6	76.4	76.9	75.7
NPWM3_2_100	77.4	78.8	79.4	79.2	77.4	74.3	76.4	77.5	77.1	75.8
RD101_100	80.5	81.9	82.4	81.1	78.8	76.1	77.9	79.3	78.1	75.9
RD121_100	80.9	82.1	82.6	81.1	78.9	77.4	78.7	79.3	78.0	75.9
RD102_100	81.4	83.1	83.9	81.2	78.4	79.0	81.2	79.3	76.4	73.0
RD122_100	81.4	83.0	82.8	80.1	77.3	79.2	80.4	78.8	75.6	71.7
LPD2_75	79.4	80.9	80.2	78.5	75.9	76.4	78.0	76.7	74.3	71.2
LPD1_75	74.2	75.9	75.6	73.3	70.6	71.1	73.1	73.6	72.4	69.6
SND3_100	81.6	82.9	84.2	82.9	81.2	78.8	80.7	81.8	79.7	72.6
SND2_100	81.9	83.0	82.1	84.3	82.1	79.2	80.9	81.9	80.7	78.4
TDIST1_100	77.8	75.3	70.7	66.9	63.3	63.7	60.1	55.9	52.0	49.2
TDIST2-100	78.9	77.1	73.6	73.6	64.9	69.6	69.8	67.2	63.1	60.1

1. Results with *k*-NN Classifier: The results on noisy and noise-less Devanagari hand-printed character images with *k*-NN are listed in Table.4. The experiments are conducted only on odd values of k in order to avoid conflict occurring as a result of majority rule. The Euclidean metric is considered for experiments.
2. Results with PNN Classifier: The output of PNN is dependent upon spread i.e. Sp. The recognition results with PNN for different values of Sp on all features under study are listed in Table.5.

5. RESULT ANALYSIS AND DISCUSSION

The performance of both the classifiers is almost same on all features except LPD2 on noise-less images. The performance of both the classifiers is almost same on all features except LPD2 on noisy images. The NPW2_1 and NPWM3_1 features are most performing whereas TDIST2 is least performing in case of both noise-less and noisy images. There is lot of difference between the performance of NPW2_1 or NPWM3_1 and TDIST2. The performance of NPWM3_1 is a little better as compared to NPW2_1 but this on the cost of considering more pixels i.e. thrice the number on each pixel. The features SND2 and RD102 perform almost neck to neck. One point which is noteworthy is that in case of NPW2_1 there are 16 pixels on each pixel which contribute to feature extraction whereas in all other features its number is about 48 except TDIST2 where it is not possible to estimate the exact number. The analysis of results with various features using PNN and *k*-NN classifiers in noisy and noise-less situation is detailed in Fig.6.

The error rate on noisy and noise-free images using *k*-NN and PNN is analyzed in Fig.7. The error difference is very small in case of NPW2_1, NPWM3_1 and RD102 with *k*-NN. It is maximum 2.2% in all cases. It is more in case of SND2 i.e. about 2.9% and much more in case of TDIST2 i.e. 8.2% with *k*-NN. Almost same trends have been observed with PNN classifier with small variation. On noisy situation NPW2_1 feature is most performing and TDIST2 is least performing and there is huge difference between there discrimination ability on noisy situation.

6. CONCLUSION

In this paper, three new features are proposed and these are Spiral Neighbor Density (SND), Layer Pixel Density (LPD) and Ray Density (RD). In addition to this two more features i.e. Total Distance in four Directions (TDIST) and Neighbor Pixels Weight (NPW) has been discussed. In case of all features, the neighborhood pixels of a pixel are also considered for features extraction. All the features under study can be computed from binary as well as gray images. In this study, the experiments are conducted by considering different number of pixels at each pixel for all the features under study.

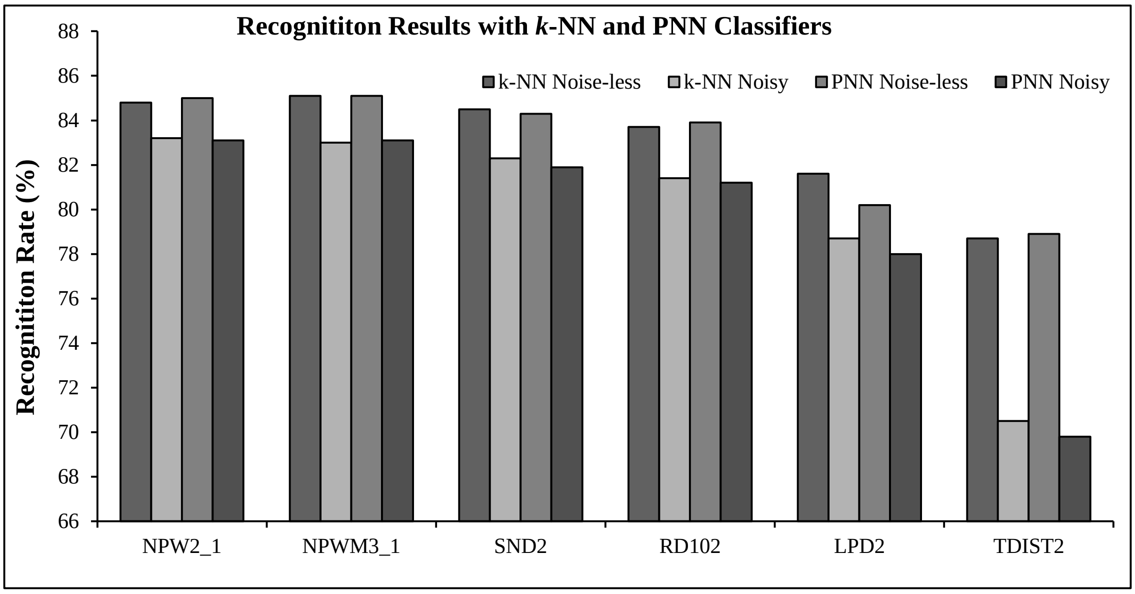


Fig.6. Analysis of results with various features using PNN and *k*-NN classifiers in noisy and noise-free situation

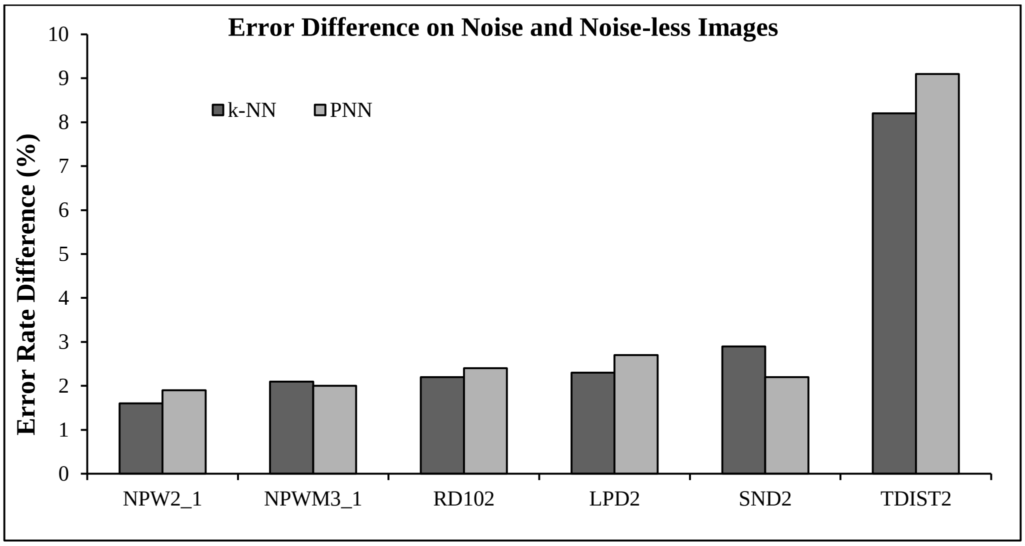


Fig.7. Error difference analysis in noisy and noise-less situation using *k*-NN and PNN

As per our observation, NPW2_1 and NPWM3_1 are most performing and SND2 and RD102 less performing whereas TDIST2 is least performing. In earlier study [5], the NPW(named as NPW2_1 in this study) is tested against some features which have been used as secondary features in many optical character recognition applications on noise-less and noisy images. So NPW2_1 or NPWM3_1 are better than other features. The feature work well on noisy images as its recognition rate is not much affected in comparison to other features. Though the performance of Spiral Neighbor Density (SND), Layer Pixel Density (LPD) and Ray Density (RD) is not higher but these features can be used as primary or secondary feature in pattern recognition applications. The performance of these features needs to be investigated either individually or in combination with other features and/or classifiers for various pattern recognition applications. Also the performance of these features needs to be tested on gray images.

REFERENCES

[1] V.K. Govindan and A.P. Shivaprasad, "Character Recognition - A Review", *Pattern Recognition*, Vol. 23, No. 7, pp. 671-683, 1990.

[2] Henry S. Baird, "Feature Identification for Hybrid Structural/Statistical Pattern Classification", *Computer Vision, Graphics and Image Processing*, Vol. 42, No. 3, pp. 318-333, 1988.

[3] K. Anisimovich, V. Rybkin, A. Shamis and V. Tereshchenko, "Using Combination of Structural, Feature and Raster Classifiers for Recognition of Hand-printed Characters", *Proceedings of Fourth International Conference of Document Analysis and Recognition*, Vol. 2, pp. 881-885, 1997.

[4] L. Heutte, T. Paquet, J.V. Moreau, Y. Lecourtier and C. Olivier, "A Sturctural/Statistical Feature Based Vector for

Handwritten Character Recognition", *Pattern Recognition Letters*, Vol. 19, pp. 629-641, 1998.

[5] S. Kumar, "Neighborhood Pixels Weights-A New Feature Extractor", *International Journal of Computer Theory and Engineering*, Vol. 2, No. 1, pp. 1793-8201, 2010.

[6] Il-Seok Oh and Ching Y. Suen, "Distance Features for Neural Network-based Recognition of Handwritten Characters", *International Journal on Document Analysis and Recognition*, Vol. 1, No. 2, pp. 73-88, 1998.

[7] S. Kumar, "Recognition of Pre-Segmented Devanagari Handwritten Characters using Multiple Features and Neural Network Classifier", Ph.D. Thesis, Panjab University, 2008.

[8] Richard E. Woods and Rafael C. Gonzalez, "*Digital Image Processing*", 2nd Edition, Prentice Hall Professional Technical Reference, 1992.

[9] Philip D. Wasserman, "*Advanced Methods in Neural Computing*", Van Nostrand Reinhold, 1993.

[10] Vikas J. Dongre and Vijay H. Mankar, "A Review of Research on Devanagari Character Recognition", *International Journal of Computer Applications*, Vol. 12, No. 2, pp. 8-15, 2010.

[11] Debashis Ghosh, Tulika Dube and Adamane P. Shivaprasad, "Script Recognition - A Review", *Pattern Analysis and Machine Intelligence*, Vol. 32, No. 12, pp. 2142-2161. 2010.

[12] Ahmed Lawgali, "A Survey on Arabic Character Recognition", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 8, No. 2, pp. 401-426, 2015.

[13] N. Arica and F.T. Yarman-Vural, "An Overview of Character Recognition Focused on Off-line Handwriting", *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, Vol. 31, No. 2, pp. 216-233, 2001.

[14] O.D Trier, A.K. Jain and T. Taxt, "Feature Extraction Method for Character Recognition – A Survey", *Pattern Recognition*, Vol. 29, No. 4, pp. 641-662, 1996.

[15] S. Kumar, "A Three Tier Scheme for Devanagari Hand-printed Character Recognition", *World Congress on Nature & Biologically Inspired Computing*, pp. 1016-1021, 2009.