International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075, Volume-8 Issue-6S2, April 2019

Modified K-Means and Symbolic Representation in Kannada Character Recognition

B N Ajay, C Naveena

Abstract*: This paper describes an OCR system for printed text documents in Kannada, a South Indian language. Many commercial OCR systems are now available in the market, but most of these systems work for Roman, Chinese, Japanese and Arabic characters. There are no sufficient number of works on Indian language character recognition especially Kannada. In this work we proposed kannada character recognition system using texture features. Here we fuse the texture features like Local Binary Pattern, and Gray Level Local Texture Pattern using concatenation rule and the texture fea- tures are represented using symbolic representation. Finally, Weighted K-means is explored for the purpose of clustering. This method is simple to implement and realize, also it is computationally efficient

***Keywords*: LTP, GLTP, Weighted KNN, Segmentation**

I. INTRODUCTION

Optical character recognition (OCR) lies at the core of discipline of pattern recognition where the objective is to interpret scanned images of handwritten or typewritten text into computer processa- ble format. Due to the advancement of digital computers many tools are available for word processing and large amount of paper documents are processed for various reasons all over the world. Extracting information from paper documents manually would add significant cost in terms of human labour. So, making computers able to read would allow for substantial savings in terms of the costs for data entry, mail processing, form processing and many other similar situations. This real world problem can be made easy by OCR. Character recognition can be divided into two major categories according to mode of the document given to it i) Online character recognition ii) Offline character recognition. Further, Offline character recognition is divided into i) Printed character recognition ii) Handwritten character recognition. Kan- nada, the official language of the south Indian state of Karnataka, is spoken by about 48 million people. The Kannada alphabets were developed from the Kadamba and Chalaukya scripts, descendents of Brahmi which were used between the 5 th and 7 th centuries A.D. The basic structure of Kannada script is distinctly different from the Roman script. Unlike many north Indian lan- guages, Kannada characters do not have shirorekha (a line that connects all the characters of any word) and hence all the charac- ters in a word are isolated. This creates a difficulty in word seg- mentation. Kannada script is more complicated than English due to the presence of compound characters. However, the concept of upper/lower case characters is absent in this script. Hence in this work we investigate the suitability of texture features in designing a system for kannada character classification. Character is seg- mented using a threshold based method and texture features viz., Local Binary Pattern and Gray Level Local Texture Pattern. These features are used for training and classification using weighted K-means.

II. RELATED WORK

For the past few decades, intensive research has been done to solve the problem of printed and handwritten character recognition. Various approaches have been proposed to deal with this problem. Challenging problems are being encountered and solutions to these are targeted in various ways to improve accuracy and effi- ciency. A brief overview of the numeral recognition work done on Indian scripts is presented below. M. Hanmandlu et al.,[1] pro- posed a method for recognition of handwritten Hindi numerals, written in Devanagari script. The method is based on a kind of exponential membership function fitted to the fuzzy sets derived from features consisting of normalized distances obtained using the box approach. The exponential membership function is modi- fied by two structural parameters that are estimated by optimizing the entropy subject to the attainment of membership function to unity. The overall recognition accuracy is found to be 96%. How- ever, the experiments are carried out on a limited database with size of 3500 samples. Rajashekaradhya et al.,[2] used zone cen- troid and image centroid based distance metric feature extraction method for recognition of Kannada, Telugu, Tamil and Malayalam numerals. The character centroid is computed and the numeral image is further divided into n equal zones. Average distance from character centroid to the each pixel present in zone is computed. Similarly zone centroid is computed and average distance from the zone centroid to each pixel present in zone is computed. Nearest neighbour and feed forward back propagation neural network classifiers are used for subsequent classification and recognition purpose. In [3], U Pal et al. have proposed a modified quadratic classifier based scheme towards the recognition of off-line hand- written numerals of six popular Indian scripts.

One of the scripts is Kannada. The features used in the classifier are obtained from the directional information of contour points of the numerals. For feature computation, the bounding box of a numeral is segmented into blocks and the directional features are computed in each of the block. These blocks are then down sampled by a Gaussian filter and features obtained from the down sampled blocks are fed to modified quadratic classifier for recognition.

Revised Manuscript Received on December 22, 2018.
B N Ajay, Department of Computer Science, VTU RRC
C Naveena , Department of Computer Science, VTU RRC

Retrieval Number: F22290486S219/19©BEIESP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A five-fold cross validation technique has been used for result computation and they have reported 98.71% accuracy for Kannada scripts obtained by performing experiments on their own data set. Benne at al. [4] proposed recognition system for isolated handwritten numerals recognition for three popular Indian scripts namely, Kannada, Devanagri and Telugu numeral sets. The proposed method is thin- ning free and without size normalization. The structural features viz. directional density of pixels, water reservoirs, maximum pro- file distances, and fill hole density are used for handwritten nu- merals recognition. A Euclidian distance criterion and K-nearest neighbour classifier is used to classify the handwritten numerals. A limited data set consisting of 5250 numeral images are consid- ered for experimentation, and the overall accuracy of 95.40%, 90.20%, and 98.40% is reported for Kannada, Devanagari and Telugu numeral respectively. Dinesh Acharya and et al. [5] have used 10-segment string concept, water reservoir, horizontal and vertical strokes, and end points as features and k-means to classify to classify the Kannada handwritten numeral. They have reported the overall recognition accuracy of 90.5%. G.G. Rajput et al. [6] described a system for isolated Kannada handwritten recognition using image fusion method. Several digital images corresponding to each handwritten numeral are fused to generate patterns, which are stored 8x8 matrices, irrespective of the size of images. The numerals to be recognized are matched using nearest neighbour classifier with each pattern, and the best match pattern is consid- ered as the recognized numeral. The average recognition rate of 95.62% is reported by them obtained by performing experiments on the data set generated locally. In [7], handwritten Kannada character recognition system based on spatial features is proposed. Directional spatial features viz stroke density, stroke length and the number of stokes are employed as features to characterize the handwritten Kannada vowels using K-NN classifier. The average recognition accuracy of 90.1% is reported for vowel characters. In [8], moment based features are used for recognition of Kagunita (the Kannada compound characters resulting from the consonant and vowel combination). These features are extracted using Gabor wavelets from the dynamically pre-processed original image. Mul- ti-Layer Perceptron with Back Propagation Neural Networks are employed for character classification. Average recognition rate of 86% is reported for vowels and for consonants the average recog- nition reported is 65%. From the work reported in the literature, we note the following. In many of the proposed methods, the dataset size is small in terms of number of samples used for each character/numeral. There is no clear explanation about training and test images in case of supervised algorithms used [9]. Experi- ments are not being performed on entire character set of the script/language, eg. not all 49 characters of Kannada script are included.

III. PROPOSED METHOD

The proposed method has training and classification phases. In training phase, from a given set of training images the texture features (LTP/GLTP) are extracted and later the features are represented using interval features. Final the system is queried to weighted K-means clustering to label an unknown Character. The block diagram of the proposed method is given in Figure 1.

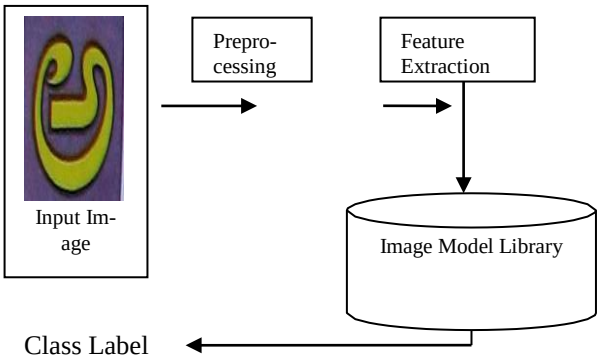

Figure 1: Block diagram of the proposed method using K-means

3.1.1 FILTERING

Character data might have been corrupted due to jitters, redundant and extraneous wild points due to erratic pen movements. In addi- tion to this, the hardware limitations in the data collection equip- ment may also introduce noise to the raw data. So, it is essential to isolate the collected data from these redundant noise components to improve the performance of the recognition system. In practice, a moving average filter or a Gaussian filter is commonly used to remove these noise segments. But, a careful choice of a filter is necessary in order to avoid the loss of important structural fea- tures like cusps, dots, etc. In the present work, a Gaussian filter with a sliding window size of seven [10] is used to filter the noisy images.

3.1.2 Morphological Processing

Binary images contain numerous imperfections. In particular, the binary regions produced by simple thresholding are distorted by noise. Morphological image processing pursues the goals of re- moving these imperfections by using morphological erode and dilate operations as shown in the below algorithm [11]. Algorithm

1) Define a square structuring element of size 3
2) Erode the image m number of times to remove spur.
3) Dilate the image m+n number of times to fill the gaps in the edges and restore the structure.
4) Erode the image n number of times to restore the image

3.2 Texture Based Proposed Model

The proposed texture based model consists of four stages: fea- ture extraction, feature level fusion, feature selection and classi- fication. Prior to the feature extraction stage, we required apply- ing the segmentation process.

3.2.1 Feature Extraction

The roughness is reflected by transitions in intensity levels on the surface of a

Character in the form of uniform and non- uniform patterns. To exploit this, we recommend extracting texture features from gray scale images of segmented characters using the various texture based models viz., LBP and GLTP which are explained in the following subsections.

$$F_1^{(ij)} = \left\{f_{11}^{(ij)}, f_{12}^{(ij)}, f_{13}^{(ij)}, \dots, f_{1u}^{(ij)}\right\}$$
$$F_2^{(ij)} = \left\{f_{21}^{(ij)}, f_{22}^{(ij)}, f_{23}^{(ij)}, \dots, f_{2u}^{(ij)}\right\}$$
$$F_3^{(ij)} = \left\{f_{31}^{(ij)}, f_{32}^{(ij)}, f_{33}^{(ij)}, \dots, f_{3u}^{(ij)}\right\}$$
$$\vdots$$
$$F_r^{(ij)} = \left\{f_{r1}^{(ij)}, f_{r2}^{(ij)}, f_{r3}^{(ij)}, \dots, f_{ru}^{(ij)}\right\}$$

3.2.1.1 Local Binary Pattern (LBP)

The LBP is a gray-scale and rotational invariant texture operator which characterizes the spatial structure of the local image tex- ture [12]. The gray-scale invariance is achieved by assigning a unique pattern label to every pixel in an image depending on binary pattern generated by comparing its value with those of its neighborhoods. A pattern label is computed by

$$\mathrm{LBP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)\, 2^p$$

$$\text{where, } s(g_p - g_c) = \begin{cases} 1, & (g_p - g_c) \geq 0 \\ 0, & (g_p - g_c) < 0 \end{cases}$$

Here, g c is the gray value of the central pixel of circularly sym- metric neighborhood g p (p = 0,1.....P-1) , g p is the gray value of its neighbors, P is the number of neighbors and R is the radius of the neighborhood.

3.2.1.2 Gray Level Local Texture Pattern (GLTP)

The LBP model is computationally efficient but inadequate to represent a local region whereas the texture spectrum (TS) mod- el will reveal more textural information but it is computationally burden. The GLTP was developed by combining the advantages of the TS and the LBP (Ojala et al., [12]). The GLTP is compu- tationally acceptable and is robust against variations in the ap- pearance of the texture to meet the real world applications [14]. These variations may be caused by uneven illumination, differ- ent viewing angles and resolving power of the sensor system. Since it is a rotational invariant and gray-scale shift invariant, it is robust against variable illumination. The GLTP model detects the number of transitions or discontinuities in the circular presentation of texture patterns in a small region, thus it suits to our dataset. When such transitions are found to follow a rhyth- mic pattern, they are recorded as uniform patterns and are as- signed with unique labels. All other non-uniform patterns are grouped under single category. It assigns a GLTP label to every pixel in an image depending on uniformity of pattern around the pixel. This labeled image is represented using a one dimensional histogram with abscissa indicating the GLTP label and ordinate representing its frequency. The following rotational and gray- scale invariant GLTP operator [14] is used for describing a local image texture.

Texture Representation

In this section, we present an interval valued type symbolic representation scheme for representing reference Character samples in the knowledgebase. Let P be the *n* number of plant species each being a class. Let CL be the *m* number of clusters obtained for a class P_i (i = 1, 2, 3, . . . , n)

$$F_1^{(ij)} = \left\{f_{11}^{(ij)}, f_{12}^{(ij)}, f_{13}^{(ij)}, \dots, f_{1u}^{(ij)}\right\}$$
$$F_2^{(ij)} = \left\{f_{21}^{(ij)}, f_{22}^{(ij)}, f_{23}^{(ij)}, \dots, f_{2u}^{(ij)}\right\}$$
$$F_3^{(ij)} = \left\{f_{31}^{(ij)}, f_{32}^{(ij)}, f_{33}^{(ij)}, \dots, f_{3u}^{(ij)}\right\}$$
$$\vdots$$
$$F_r^{(ij)} = \left\{f_{r1}^{(ij)}, f_{r2}^{(ij)}, f_{r3}^{(ij)}, \dots, f_{ru}^{(ij)}\right\}$$

after clustering. Let L be the *r* number of leaf samples in a cluster *CLij* (j = 1, 2, 3, . . . , m). Let Fl $^{(i,j)}$ be the u number of features describing the shape or texture of l^{th} leaf sample belonging to *j* cluster of *I* class. Thus, we have the shape or texture feature vectors describing all the *r* samples belonging to *j* cluster of *i* class as follows

The variation among the th feature is captured by consolidating the feature values to an interval as follows

$$\mu F_k^{(ij)} = \frac{1}{r}\sum_{l=1}^{r} f_{lk}^{(ij)}$$ and

$$\sigma F_k^{(ij)} = \left[\frac{1}{r}\sum_{l=1}^{r}\left(f_{lk}^{(ij)} - \mu f_k^{(ij)}\right)^2\right]^{1/2}$$

be the mean and standard deviation of the *kth* (texture) feature values due to all the *r* leaf samples belonging to *jth* cluster of *ith* class respectively

Classification

In this section we use the symbolic classifier [22] for classifying the Kannada characters. In classification model, a test sample of an unknown flower is described by a set of *m* distances of type crisp and compares it with the corresponding interval type features of the respective symbolic reference samples *RFj*stored in the knowledgebase to ascertain the efficiency.

Let $F_i = [d_{t1}, d_{t2}, d_{t3}, d_{t4}, \dots, d_{tm}]$ be an *m* dimensional vector describing a test flower. Let *RFc*; *c = 1,2,3,…,N* be the representative symbolic feature vectors stored in knowledgebase. During classification process each k^{th} distance (feature) value of the test sample is compared with the respective intervals of all the representatives to examine if the feature value of the test image lies within them. The test sample is said to belong to the class with which it has a

Retrieval Number: F22290486S219/19©BEIESP
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

maximum acceptance count *Ac*.

4.1 Chi-square Symbolic Dissimilarity Measure

Let *TL* be the crisp texture feature vector of *u*-dimension repre- senting

the test sample to be classified and *RL* be the symbolic texture feature vector of *u*-dimension representing the reference sample stored in the knowledgebase as in Eqn.(4.5). The matching score between the reference and test samples is estimated as shown in Eqn.(4.10)

$$DS(TL,RL)=\frac{1}{u}\sum_{k=1}^{u}\begin{cases}0 & if\left(TF_k^{(ij)-}\le TF_k\le TF_k^{(ij)+}\right)\\ min\left[\frac{(TF_k^{(ij)+}-TF_k)^2}{(TF_k^{(ij)+}+TF_k)},\frac{(TF_k^{(ij)-}-TF_k)^2}{(TF_k^{(ij)-}+TF_k)}\right] & otherwise\end{cases}$$

If the crisp feature value of a test leaf lies between the interval valued feature of a reference then the dissimilarity with respect to this feature is considered as 0; otherwise the minimum dissimilari- ty value due to lower limit or due to upper limit of a symbolic feature with respect to crisp feature is considered. The average dissimilarity value due to all the features of vectors is considered as a matching score between test and a reference.

4.2 Symbolic Similarity Measure

Let *TL* be the crisp shape feature vector of *u*-dimension representing test and *RL* be the interval valued feature vector representing reference leaf as shown in Eqn.(4.9). The matching score between the reference and test leaf sample due to shape dependent or shape independent features is computed as shown in Eqn.(4.11).

$$DS(TL,RL)=\begin{cases}Similarity_{Score} & if\left(U_1^{(ij)}\le u\le U_r^{(ij)}\right)\\ 0 & otherwise\end{cases}$$

Where

$$Similarity_{score}=\frac{1}{u}\sum_{k=1}^{u}\begin{cases}0 & if\left(SF_k^{(ij)-}\le SF_k\le SF_k^{(ij)+}\right)\\ max\left[\frac{1}{1+abs(SF_k^{(ij)+}-SF_k)},\frac{1}{1+abs(SF_k^{(ij)-}-SF_k)}\right] & otherwise\end{cases}$$

From the Eqn.(4.9), it can be observed that in case of shape dependent representation, where the dimension of the test feature

vector is different from reference feature vector, if the dimension

u of the test feature vector lies between the interval *U*1 and *Ur*, then the similarity score is computed, otherwise the similarity score is considered as 0. But, in case of shape independent representation, where the dimension of both test and reference feature vectors are same; if the crisp feature value of test leaf lies between the interval valued feature of a reference leaf then the similarity with respect to this feature is considered as 1; otherwise, the maximum similarity value due to lower limit or due to upper limit of a symbolic feature with respect to crisp feature is considered. The overall similarity score between the test and reference leaf due to all the features is obtained by dividing the total similarity value by the dimension of the feature vector in case of shape independent representation. But, in case of shape dependent representation, the overall similarity is obtained by dividing the total similarity value by the dimension of the reference feature vector.

$$GLTP_{P,R}^{ru3}=\begin{cases}\sum_{p=1}^{P}s(g_p,g_c) & \text{if } U\le 3\\ P\times 9+1 & \text{otherwise}\end{cases}$$

$$\text{where, } s(g_p,g_c)=\begin{cases}0 & \text{if } g_p<(g_c-\Delta g)\\ 1 & \text{if } (g_c-\Delta g)\le g_p\le (g_c+\Delta g)\\ 9 & \text{if } g_p>(g_c+\Delta g)\end{cases}$$

$$p=1,2,\ldots,P$$

Weighted K-Means

Generally in K-means algorithms, every data point has equal importance in locating the centroid of the cluster and every member of the data points carry unit weight. This property no longer holds in the case of density-biased sample clustering, for which each
data point represents varied density in the original data. Therefore,
the clustering algorithm has to consider a weight associated with each data point in the computation of grouping similar data points.
In [21] introduced the weight function,*w(xi)*, in his algorithm to accelerate the recomputation of the new centroids in the next iteration. It represents the density of the original data points. He has
applied this technique for market-basket analysis

Algorithm 1: Enhanced weighted k-means algorithm

Input:*D = {d1, d2, d3, ... , di,..., dn }* // Set of n data points. *di= { x1, x2, x3 ,... , xi,..., xm}* // Set of m attributes of single data point. *Wi = {w1, w2,w3, ... ,wi,...,wm}* // Weight associated to attributes of a data point.
Ck= {c1, c2 ,ck} // Set of k unique initial
centroids k// Number of desired clusters.

Output:*k* distinct
clusters Steps:

1. Initialize the k initial cluster centroids.
2. Compute the distance between each weighted data point *w*ixi (1 <= *i<= n)* to all the initial centroids *cj (1 <= j <= k)*.
3. Repeat

31. For each weighted data point *w*ixifind the closest centroid *cj*and assign *w*ixito cluster *j*. 3.2. Set Cluster-Id[*i*]=*j*. // *j*:Id of the closest cluster.
3.3. Set NearestDist[*i*]=*d(w*ixi, *cj)*.
3.4. For each cluster *j (1 <= j <= k)*, recalculate the centroids.

3.4.1 For each weighted data point *w*ixi
3.4.1.1 Compute its distance from the centroid of the present nearest cluster.
3.4.1.2 If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster.

Else 3.4.1.3 For every centroid *cj (1<=j<=k)* compute the distance *d(w*ixi, *cj).*
End for;

Until the convergence criteria is met.

Next, for each weighted data point the distance is calculated from all the initial centroids. The next stage is an iterative process which makes use of

a heuristic approach to reduce the required computational time. The weighted data points are assigned to the clusters having the closest centroids is the next step. ClusterId of a data point denotes the cluster to which it belongs. NearestDist of a data point denotes the present nearest distance from closest cen- troid. Next, step is recalculating the centroids, for each cluster. In each cluster, taking the mean value for each attribute and that mean value is treated as a centroid for the next iteration. This pro- cess is repeated until there is no change in the cluster members in consecutive iterations. This is otherwise known as convergence criteria. The proposed algorithm selects the initial centroids based on the distance calculation. Hence, the proposed algorithm provides less complexity (by reducing the number of iteration) with great accuracy. It doesn't require any additional input like thresh- old value. By varying the k value to three or four, the proposed clustering algorithm becomes multiclass classification (k class classification).

VI. EXPERIMENTATION

In this work we have created our own database despite of exist- ence of other databases as these are less intra class variations or no change in view point. We collected Kannada character imag- es from World Wide Web in addition to taking up some photo- graphs of Characters that can be found in and around. The imag- es are taken to study the effect of the proposed method with large intra class variations. Fig. 4 shows a sample image of each different class. It is clearly understandable that there is a large intra class variation. The large intra-class variability and the small inter-class variability make this dataset very challenging.The proposed clustering based multiclass classification has been experimented with different sets of dataset and the results are reported in this section. To check the robustness of the pro- posed algorithm, analysis is made against existing K-means and weighted K-means algorithms. In K-means, random initial cen- troids are taken into account for clustering. In weighted K- means algorithm, unit weight is considered for all the attributes and the initial centroids are chosen based on the weighted rank- ing algorithm. The experiment is conducted for the proposed enhanced weighted K-means algorithm and it is found that the results obtained are consistent for all the runs. To prove the effectiveness of the proposed algorithm, the initial centroids se- lected by the existing K-means and the weighted K-means algo- rithms are compared and the results are incorporated in the same Table 1.The number of objects in the clustered group varies at every execution. There is a wide deviation in the number of objects in each cluster. Number of objects in each cluster varies from 40's 50's to 80's. The number of records correctly classi- fied with the existing and the proposed algorithms are presented in Figure 3. which clearly describes the accuracy of the proposed algorithm. The results obtained clearly indicates that the consistency in the initial centroids improve the accuracy of the algorithm. K-means algorithm has not provided consistent accu- racy because it fixes random initial centroids. Though weighted K-means select random initial centroids, it considers associated weight for all the attributes and improves the accuracy compared to the earlier two algorithms. But, accuracy is not assured due to its random initial centroids. The proposed enhanced weighted K-means algorithm makes use of weighted ranking algorithm for fixing unique initial centroids in addition to assigning weightage to the attributes. This in turn yields consistent accuracy and reduces the number of iterations to meet the convergence criteria. Ultimately the time complexity is also reduced in the proposed algorithm.

VII. CONCLUSION

In this paper, we presented a novel model based on texture fea- tures for classification of kannada characters. We made a suc- cessful attempt to explore the applicability of texture features and symbolic methods for effective classification of character classification. In order to investigate the effectiveness and ro- bustness of the proposed model, we conducted series of experi- ments on our own large dataset.

REFERENCES

1. Hanmandlu M, A V Nath, A.C Mishra and V.K Madasu, Fuzzy Model Based Recognition of Handwritten Hindi Numerals using Bacterial Forgaging , 6th IEEE/ACIS International Conference on Computer and Information Science(ICIS 2007), Computer Society, 2007.
2. S.V. Rajashekararadhya and Dr. P. Vanaja Ranjan, Efficient zone based feature extraction algorithm for handwritten numeral recognition of four popular south Indian scripts, Journal of Theoretical and Applied Information Technology, pp. 1171-1180, 2005-06.
3. U. Pal, T. Wakabayashi, N. Sharma and F. Kimura, Handwritten Numeral Recognition of Six Popular Indian Scripts. In Proc. 9th International Conference on Document Analysis and Recognition. Curitiba, Brazil, September 24-26, pp. 749-753, 2007.
4. Benne R.G.,Dhandra B.V and Mallikarjun Hangarge, Tri-scripts handwritten numeral recognition: a novel approach:, Advances in Computational Research, Volume 1, Issue 2, pp 47-51, 2009.
5. Dinesh Acharya U, N V Subbareddy and Krishnamoorthy, Multi-level Classifier in Recognition of Handwritten Kannada Numeral, Proceedings of World Academy of Science, Engineering and Technology, Vol. 32, pp 308- 313,2008.
6. G. G. Rajput and Mallikarjun Hangarge, Recognition of Isolated Kannada Numeral Based on Image Fusion Method. PReMI 2007, LNCS 4815, pp. 153– 160, 2007.
7. B.V.Dhandra, Mallikarjun Hangarge and Gururaj Mukarambi. Spatial Features for Handwritten Kannada and English Character Recognition. IJCA, Special Issue on RTIPPR (3), pp 146–151, 2010.
8. Ragha, L.R., Sasikumar, M.: Feature Analysis for Handwritten Kannada Kagunita Recognition. International Journal of Computer Theory and Engineering, IACSIT 3(1), 1793–8201, 2011.
9. Suresh Kumar D, Ajay Kumar B R, K Srinivas Kalyan, Kannada Character Recognition System using Neural Network, National Journal on Internet Computing, Vol-1, 33-35, 2007.
10. Zhang, Jun & Hu, Jinglu (2008). "Image segmentation based on 2D Otsu method with histogram analysis". Computer Science and Software Engineering, 2008 International Conference on. 6: 105–108.
11. Zhu, Ningbo and Wang, Gang and Yang, Gaobo and Dai, Weiming (2009). "A fast 2d otsu thresholding algorithm based on improved histogram". Pattern Recognition, 2009. CCPR 2009. Chinese Conference on: 1–5.
12. Ojala T., M. Pietikainen and T. Maenapaa, 2002. Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24. no. 7, pp. 971-987.
13. Guo Z., L. Zhang and D. Zhang, 2010. Rotation invariant texture classification using LBP Variance (LBPV) with global matching. Pattern Recognition, vol. 43, Pp.

Retrieval Number: F22290486S219/19©BEIESP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

706-719.
14. Surliandi A and K. Ramar, 2008. Local Texture Patterns - A univariate texture model for classification of images. Proceedings of the 16th International Conference on Advanced Computing and Communications (ADCOM08), Tamilnadu, India, pp. 32-39.
15. Jain A., K. Nandakumaran and A. Ross, 2005. Score normalization in multimodal biometric systems. Pattern Recognition, vol. 38, pp. 2270-2285.
16. Kira K and L. Rendel, 1992. The feature selection problem: traditional methods and a new algorithm. Proceedings of the Tenth Na- tional Conference on Artificial Intelligence, San Jose, USA, pp. 129-134.
17. Hall M., 2000. Correlation-based feature selection for discrete and numeric class machine learning. Proceedings of the Seventeenth In- ternational Conference on Machine Learning, Stanford, USA, pp. 359-366
18. Ververidis D and C. Kotropoulos, 2008. Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition. Signal Processing, vol. 88, no. 12, pp. 2956-2970.
19. K.P. Zaw and Z.M. Kyu, "Segmentation Method for Myanmar Character Recognition using Block based Pixel Count and Aspect Ratio", 27th International Conference on Computer Theory and Application (ICCTA), October 2017.
20. H. S. Mohana, R. Pradeepa, B. Rajithkumar BK and M. Shivakumar, "Printed and Handwritten Mixed Kannada Characters Recognition using Template Matching Method. International Journal of Electronic and Communication Technology, Vol. 6, Iss. 2, 2015.

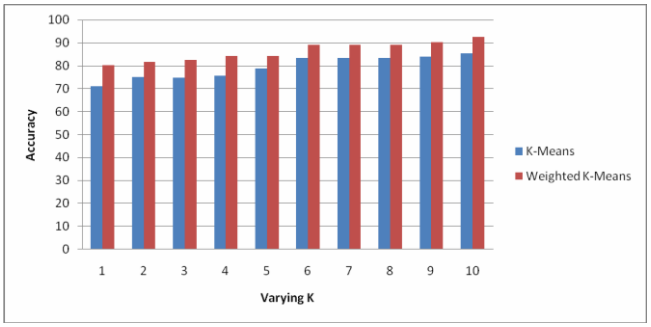

Figure 2: Illustration of preprocessing stages of character Ja. (a) Original Image (b) Binearization (c) Filtered and Morphological Image sampled

Figure 3. Classification accuracy of the proposed model for classification based on texture models

Figure 4: Sample Character images of different classes

Table 1. results of the proposed weighted K-means clustering algorithm

	K-Means		Weighted K-Means	
K	Initial Centroids (Object Number)	Number of Objects in Every Cluster	Initial Centroids (Object Number)	Number of Objects in Every Cluster
2	80,112	30,40	176,39	25,26
3	6,48,58	45, 56,42	220,92,292	61,39,35
4	5,16,71,142	88,118,103,86	213,52, 251,78	10,46,44,56
5	30,93, 8,33,45	112,39,45 ,20,21	81,34, 94,7,114	89,45,69,41,120
6	188,157, 164, 98,75,33	73,54,33, 46, 97,49,39, 21	21,35,66, 220,92,292	10,19, 32,39,78,84
7	190,196, 124,68, 61,202, 119	98,75,33, 175, 100,50,46	213,52, 251,78, 134,9,50	51,38, 143,64,120,78,96
8	56,187, 129,132, 90,63, 61,82	90,71,43, 171, 156,52,44 ,35	77,3,72,2, 175,5,18, 9	56,45, 158,89,110,36,78,26
9	106,49, 22,29, 243,142, 6, 33, 46	89,46,38, 33, 144,25,36 ,46,78	50,74,90, 16, 16,196,18 8,177,190	77,55,44,26,100, 39,86,136,157

10	199,176,54,58,177,131,89,249,38,85	87,48,28,35,164,177,102,103,58,79	75,62,24,35,3,170,113,164,243,142,	112,39,45,35,177,192,65,79,38,100

Retrieval Number: F22290486S219/19©BEIESP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication