Multi-font and Multi-size Kannada Character Recognition based on the Curvelets and Standard Deviation

Mamatha H.R
Asst Prof, Department of ISE, PES Institute of Technology, Bangalore, India

Sucharitha S
Department of CSE, PES School of Engineering, Bangalore, India

Srikanta Murthy K
Professor, Department of CSE, PES School of Engineering, Bangalore, India

ABSTRACT

Good features are those whose values are similar for objects belonging to the same category and distinct for objects in different categories .In this paper, an attempt is made to develop an algorithm for recognition of machine printed isolated Kannada vowels of different font, size and style using fast discrete curvelet transform. A standard deviation is applied to the coefficients obtained and the result of this is used as the feature vector. In addition, the features are obtained by applying the curvelet transform with different scales. A k-NN classifier is adopted for classification. The proposed algorithm is experimented on 5850 samples of vowels. The recognition is independent of the font and size of the printed characters and the system is seen to deliver reasonable recognition accuracies for different scales with 90.17% being the highest for a combination of scales 1, 2, 3.

General Terms

Pattern Recognition, Optical Character Recognition.

Keywords

Curvelets, Standard deviation, Kannada character recognition, k-NN classifier

1. INTRODUCTION

The idea and medium are the two main constituents of any document. The ideas are authored in different languages. Every text based document, in its printed or hand written form involves a language and the corresponding character set. The recognition of the character that constitutes the document makes character recognition as the basic step towards document recognition [1].

A character can be considered as the general representation of the information communicated by a language using human voice. The concept of character which contains abstract information is converted to the physical form by using a number of writing tools, so that the information it contains can be transmitted to the reader. As human's function of character recognition has been being studied in advanced form, the technology of character recognition, which is comparable to human's abilities, has been realized. Hence, character recognition has to be developed to the technology satisfying the requirement of automation of insertion and human-computer interface.

Over the last few decades, a lot of research has been done on Optical Character Recognition (OCR) and many papers have been published in this area. At present, there are numerous commercial OCR systems that are available in the market. But most of these systems work for Roman, Chinese, Japanese and Arabic characters. There are no sufficient numbers of works on Indian language character recognition.

In fact, there are few works reported in the literature on character recognition of Indian languages. Most of the pieces of existing work are concerned about Devanagari and Bangla script characters, the two most popular languages in India. Some studies are reported on the recognition of other languages like Tamil, Telugu, Oriya, Kannada, Punjabi, Guajarati, etc. At present, several organizations have started work on Indian languages OCR. Ministry of Information Technology, Government of India, has initiated a Technology Development on Indian Languages (TDIL) project under which OCR system development for most of the important Indian language scripts have been taken up by different labs and academic institutions [2].

From the literature survey it is observed that there is a lot of demand on Indian scripts character recognition and an excellent review has been done on the OCR for Indian languages [3]. A Detailed Study and Analysis of OCR Research on South Indian Scripts is presented in [4].

It can be seen that there are a number of reasonably good OCR systems available in the market for local as well as universal languages and scripts. But the recognition rate drops down drastically when the input document consists of characters which belong to different font styles and font sizes. This can be particularly seen in documents such as newspapers which contain headings or subheadings, pamphlets with artistic characters, text books for kids etc. As a consequence the methodology for Kannada OCR should be font and size independent. It must also be scalable for including variety of fonts for training with little effort. Hence a large amount of research is going on for the development of an efficient and robust OCR system for different languages and scripts containing diverse font styles and sizes. In order to overcome the above mentioned complexities, many methods have been proposed for OCR of Indian scripts like Bangla, Devanagiri, Telugu and Tamil.

Ashwin et al [5] have formed the three basic Zones for the underlying character image. Each Zone is divided into a number of circular tracks and sectors. ON pixels in each angular region is used as a feature. Support vector machine was employed for the classification of characters and achieved an accuracy of 86.11%.A modified region decomposition method and optimal depth decision tree for the recognition of Kannada characters was used by Nagabhushan et al [6]. Sanjeev Kunte et al [7] have developed an OCR system for the recognition of basic characters of printed Kannada text, which works for different font size and font style. Each image was characterized by using Hu's invariant and Zernike moments. They have achieved the recognition accuracy as 96.8% with Neural Network classifier. An OCR system for complete set of printed Kannada characters employing wavelet descriptors is presented in [8].This survey of literature on Kannada OCR reveals that the research for Kannada character recognition is still in infant stage.

The most important factor that contributes to the performance of a character recognition system is the selection of an apt feature extraction method for a particular type of input documents. In [9] a survey on the feature extraction methods for character recognition is reviewed.

A lot of different methods have been used to extract features in an OCR system. Over the years, Multiresolution schemes have been used efficiently to perform feature extraction. In this paper we have put forward a Multiresolution feature extraction technique based on the Digital Curvelet Transform of Candes and Donoho[10]. People who promoted curvelets demonstrated that although wavelets are good at representing point discontinuities, they are not very good at representing edges. But this drawback has been overcome by curvelets and it has been used in a lot of applications in the fields of astrophysics, seismic imaging and image de-noising. However, we have not come across curvelets being used as a feature extraction method for Kannada OCR.

The rest of the paper is organized as follows. Section 2 describes the Kannada script, section 3 discusses about the proposed methodology, section 4 briefly discusses the experimental setup and the results obtained are discussed respectively. Finally in Sections 6 and 7, comparative study and conclusions are made.

2. DESCRIPTION OF THE KANNADA SCRIPT

Kannada also called as Canarese, is the official language of the state of Karnataka which is present in the southern part of India. Described as 'sirigannada', it is one of the earliest languages evidenced epigraphically in India. The language is spoken by about 50 million people spread over the states of Karnataka, Tamil Nadu, Andhra Pradesh and Maharashtra. The visual form of the Kannada language is the Kannada script. The Kannada script originated from the southern Bramhi Lipi during the period of Ashoka. It underwent a lot of changes from time to time during the reign of Sathavahanas, Kadambas, Gangas, Rastrakutas and Hoysalas[11]. The modern Kannada script emerged in the thirteenth Century. It is also used to write Tulu, Konkani and Kodava languages.

The Kannada script has a large number of structural features which are common with other Indian language scripts. The writing system of the Kannada script includes the principles that govern the phonetics, syllabic writing systems and phonemic writing systems of the language. The effective unit of writing Kannada is the orthographic syllable consisting of a consonant (C) and vowel (CV) core and optionally, one or more preceding consonants and hence has the canonical structure of ((C) C) CV.

Kannada has 49 basic characters which are classified into three categories: swaras(vowels), vyanjans(consonants) and yogavaahas (part vowel, part consonants). The scripts also include 10 different Kannada numerals of the decimal number system.

Kannada lacks a standard test bed of character images for OCR performance evaluation. As a consequence the methodology for Kannada OCR should be font and size independent. It must also be scalable for including variety of fonts for training with little effort.

Kannada is a non-cursive script where most of the characters are circular in shape. A sequence of characters is written horizontally from left to right to form a word and then a sentence. The characters present within a word are isolated. The vowels present in the Kannada script are as shown in Figure 1.

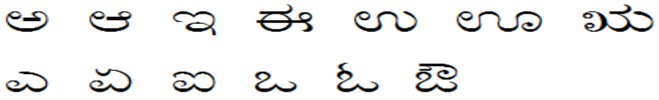


Fig 1: Kannada Vowels of style Nudi Akshar 01 with regular font size 48

One of the challenges faced during the development of an OCR system for Kannada Characters is the distinction of several similar shaped components in the script. Some characters have very similar variation between them and this leads to recognition complexity and reduces the accuracy rate of the recognition system. Some of the sets of similar characters are as in Figure 2.

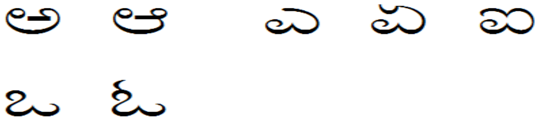


Fig 2: Some similar vowels

3. PROPOSED METHODOLOGY

3.1 Data Set and Preprocessing

Kannada lacks a standard test bed of character images for OCR performance evaluation. Hence, we have created a database of the 13 vowels using Nudi 4.0 software. 45 Different font styles like Nudi Akshara-01, Nudi Akshara-02, Nudi Akshara-03 etc were used. Each font was used to create characters of 10 different sizes with the size varying between 12 and 72 points. These are scanned through a flat bed HP scanner at 300 dpi. Isolated characters are obtained by manual cropping. Thus 450 different samples of each vowel were created with the total of 5850 samples.

Initially the gray scale images are converted to binary using global threshold method, object representing 1 and background of the object as 0. On the binary image, thinning is applied. Thinning is an image preprocessing operation performed to make the image crisper by reducing the binary-valued image regions to lines that approximate the skeletons of the region. Region labeling is performed on the thinned binary image of the character and a minimum rectangle bounding box is inserted over the character. The bounding box image would be of variable size due to different style and size of characters. Hence this image is resized to a 128*128 image.

3.2 The Curvelet Transform

Curvelet Transform is a unique member of the emerging family of multiscale geometric transforms. It was seen that the traditional multiscale approaches such as wavelets had a lot of inherent limitations. In order to overcome these limitations the curvelet transform was developed in the recent years. The curvelet transform can be theoretically described as a multiscale pyramid which has many directions and positions at each length scale. Its elements are needle shaped at fine scales [12].

Curvelets were introduced for the first time in [10] and they have been around for a little over five years by now. As soon as they were introduced, numerical algorithms were developed for their implementation by researchers [13, 14]. However, in the past few years, curvelets have undergone a lot of revisions in order to make them easier to understand and use. Two new curvelet algorithms were introduced in [15]. The first one is the Unequispaced FFT Transform. Here, the curvelet coefficients are found by the irregular sampling of the fourier coefficients of an image. The second algorithm that was developed is the Wrapping Transform. This uses a series of translations and a wraparound technique. Both these algorithms produce the same output, but the Wrapping Transform is a more intuitive algorithm and the computation time for this algorithm is also lesser [16]. Hence, in this paper we exclusively focus on the Discrete Curvelet Tranform with the Wrapping Technique.

The input given to the Curvelet Transform based on wrapping of Fourier samples is a 2-D image in the form of a Cartesian array represented as $f\ [m,\ n]$ where $0 \le m < M, 0 \le n < N$. The algorithm generates the output as number of curvelet coefficients which indexed by a scale j, an orientation l and two spatial location parameters ($k1$, $k2$). In order to obtain the curvelet texture descriptor, a number of statistical operations are applied to the coefficients generated as the output. The coefficients of the Discrete Curvelet Transform can be defined by the equation (1) [15]:

$$C^D(j,l,k_1,k_2) = \sum_{\substack{0\le m<M \\ 0\le n<N}} f[m,n]\emptyset^D_{j,l,k_1,k_2}[m,n] \quad (1)$$

Here, each $\emptyset^D_{j,l,k_1,k_2}[m,n]$ is a digital curvelet waveform. If the frequency responses of curvelets at different scales and orientations are combined, a rectangular frequency tiling that covers the whole image in the spectral domain (Figure 3) is obtained. The curvelet spectra cover the entire frequency plane of the image. Thus there is no loss of information like in the case of Gabor filters.

When the curvelet transform is applied on a C^2 curve, most of the curvelet coefficients will have negligible magnitude values. Hence it was declared that optimal sparseness is offered by curvelet transform when the input images are "curve-punctuated smooth" i.e. where the images are smooth with the exception of discontinuities along C^2 curves [17]. Good results can be obtained if the Curvelet Algorithm is used with images which contain a large number of C^2 curves (i.e. an image which has a large number of long edges) [16].

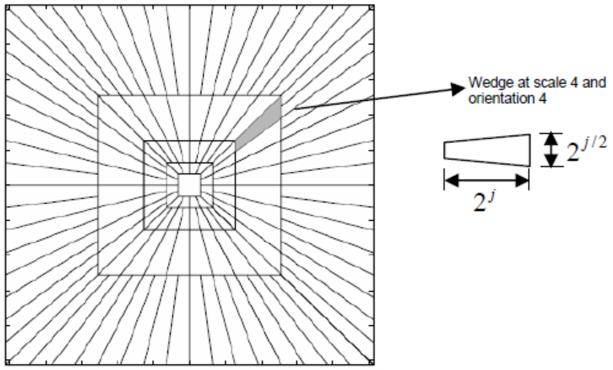


Fig 3: Rectangular frequency tiling of an image with 5 level curvelets.

3.3 Feature Extraction

Wrapping based discrete curvelet transform using Curvelab-2.1.2 is applied to find the coefficients and to create feature vectors for every 128×128 image in the database. In this experiment we have used the default orientation and 4 levels of discrete curvelet decomposition. Hence for an image of size 128*128, curvelet coefficients in four different scales are obtained.

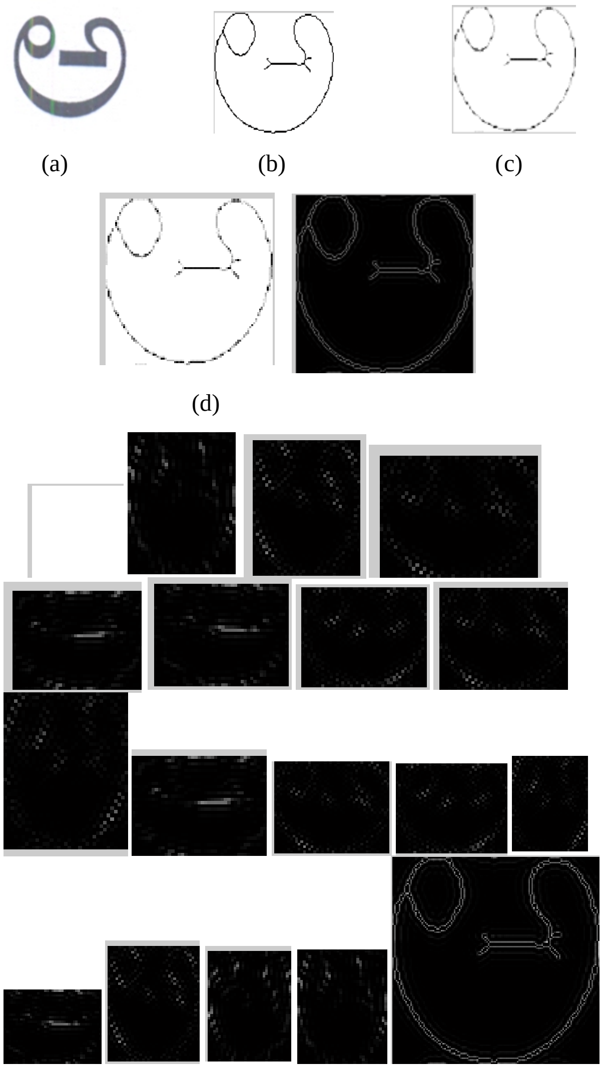
The obtained curvelet coefficients for each sample are numeric. In this implementation, we have chosen wavelet in the finest level of curvelet transform. This is because use of wavelet reduces the redundancy factor [18]. After the application of curvelet transform on the input, one subband at the coarsest and one subband at the finest level of curvelet decomposition are obtained. Different numbers of subbands are obtained at each level for the other levels of curvelet decomposition. All the coefficients obtained cannot be used in the feature vector as it will increase the size of the feature vector drastically and also the time taken for feature vector formation. Hence for extracting the best features and also decreasing the size of feature vector for each sample, we use standard deviation as the dimension reduction technique. First, the standard deviation of the coarsest and the finest levels are calculated using equation (2). Then, we calculate the standard deviation of the first half of the total subbands at each of the remaining scales. The reason for considering only the first half of the total subbands at a resolution level for feature calculation is because; the curvelet at angle θ produces the same coefficients as the curvelet at angle (θ+π) in the frequency domain i.e. these subbands are symmetric in nature. Hence, considering half of the total number of subbands at each scale reduces the total computation time for the feature vector formation without the loss of information of the image. For the finest and the coarsest subbands the standard deviation calculated is used directly in the feature vector but for the other subbands the sum of the standard deviation is calculated and stored in the feature vector. This helps us to reduce 7225 features obtained in scale 1 to 85, 23609 features obtained in scale 2 to 213, 29769 features obtained in scale 3 to 179 and 31257 features obtained in scale 4 to 173.

$$s = \left(\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{1/2} \qquad (2)$$

Where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and n is the number of elements in the sample.



(a) (b) (c)

(d)

(e)

Fig 4: An example of Digital Curvelet transform for the vowel

(a) Original Image (b) Resized and Thinned Image (c) Curvelet coefficients in scale 1 (d) Curvelet coefficients in scale 2 (e) Curvelet coefficients in scale 3

3.4 Classification

The classifier used in the proposed method is the k nearest neighbor classifier [19]. The Nearest Neighbor Classifier is an efficient technique to use when the classification problem has pattern classes that display a reasonably limited degree of variability. It considers each input pattern given to it and classifies it to a certain class by calculating the distance between the input pattern and the training patterns. It takes into account only k nearest prototypes to the input pattern during classification. The decision is generally based on the majority of class values of the k nearest neighbors.

The k value specifies the number of neighbors used for classification. In this paper, k=1, k=4, k=7, k=10 and k=13 are chosen and the results are compared to find out the optimum value of k. From the experimental results it is clear that the recognition rate has been independent of the change in the k value of the k-NN classifier. In the k-Nearest neighbor classification, we compute the distance between features of the test sample and the feature of every training sample. Here, Cosine measure is used as the distance and nearest as the rule. The cosine distance measure can be represented by equation (3),

$$similarity = cos(\theta) = \frac{A.B}{\|A\|\|B\|}$$

$$similarity \;=\; \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \qquad (3)$$

Where *A* and *B* are two vectors of attributes, and θ, the cosine similarity [20].

3.5 Algorithm

Training:

Begin

Input: a set of sample images

Output: a database i.e., a feature matrix of numerals

Method:

- Acquire the sample.
- Convert the image into binary form using global threshold method. Then invert the image such that the background is black and object is white. Perform thinning of the image and invert it back.
- Perform the region labeling for the preprocessed image and fit a minimum rectangle-bounding box on it.
- Resize the image to 128*128 pixels.
- Apply Wrapping based discrete Curvelet transform.
- Store the obtained feature vector in the database.

End

Testing:

Begin

Input: a set of test image, a database i.e., a feature matrix of vowels

Output: class of test image

Method:

- Acquire the sample
- Convert the image into binary form using global threshold method. Then invert the image such that the background is black and object is white. Perform thinning of the image and invert it back.
- Perform the region labeling for the preprocessed image and fit a minimum rectangle-bounding box on it.
- Resize the image to 128*128 pixels.
- Apply Wrapping based discrete Curvelet transform
- Obtain the feature vector of the test sample
- Classify using k-NN classifier.

End

4. EXPERIMENTAL RESULTS

The experiments were carried out in Matlab 7.5.0, on a 64-BIT 2.67 GHz INTEL i5 processor, with 4 GB RAM. The curvelet transformation was done using the Curvelet 2.1.2 toolbox, available from http://www.curvelet.org. The morphological operations were performed using Matlab's Image Processing Toolbox.

Forty five Kannada fonts available in Nudi 4.0 software were used for experiments. For each of the fonts 10 different font sizes viz., 12, 14, 18, 20, 22, 24, 28, 36, 48 and 72 points were taken. The entire data set consisted of 450 samples for each vowel comprising of 45 different fonts and ten font sizes. The total number of samples used was 5850, of which 80% of the samples i.e. 4680 samples were used to train the classifier and the remaining 1170 samples were used for testing. Eight different font sizes of all the 45 fonts were used as the prototype for the k-NN classifier. The remaining two sizes were used for testing. For each of the fonts, the two different sizes that were to be tested were selected randomly, i.e. say for the Nudi 01 e font the testing set comprised of sizes 12 and 72 while for Nudi 02 e the font sizes for testing were randomly selected to be 14 and 48 points. For each of the fonts, the rest of the ten font sizes served as the prototype for the k-NN. Testing was also done for 780 samples selected randomly from the trained set of samples. The classification is done using Cosine distance measure as the distance and nearest as the rule. Five different classifiers were used with the K values as 1,4,7,10,13. k value specifies the number of neighbors used for classification.

The obtained results for curvelets and standard deviation using different scales with the default orientation and their combinations are shown in Table 1.

From the results obtained, we can conclude that it is not always necessary to use all the coefficients obtained by applying the curvelet transform. Instead we can use the coefficients with larger values to form the feature vector. It can also be noted that the results have been independent of the change in the k value of the KNN classifier.

According to the Table 1, for the testing samples when we use the information of scale 3 for achieving feature vector, our recognition rate got better. So, the information obtained by using scale 3 is said to be the best for classification for the samples considered.

Another important observation is that when the combination of scales are used for classification, the recognition rates appear better and is the best (90.17%) with 0% rejection rate when the scales 1, 2, 3 are used together. One of the reasons for this result might be the use of the image's information in different sizes, partitions and various scales. For the training samples used for testing, the recognition rate was found to be 100%.

From the Table 2 we can conclude that the possibility of misclassification is higher among the sets of similar vowels mentioned above in Figure 2. For example for the combination of scales 1, 2, 3 the vowel ಏ is recognized as ಎ 6 times and as ಐ 13 times.

Table 1: The average recognition rates for different scales and their combinations using Cosine distance measure.

Scale	Test data		Trained data	
	Total number of recognized samples	Average recognition rate (%)	Total number of recognized samples	Average recognition rate (%)
1	1018	87.00	780	100
2	992	84.78	780	100
3	1026	87.69	780	100
4	964	82.39	780	100
1,2	991	84.70	780	100
1,3	1045	89.31	780	100
1,4	1007	86.06	780	100
2,3	1040	88.88	780	100
2,4	1009	86.23	780	100
3,4	997	85.21	780	100
1,2,3	1055	90.17	780	100
2,3,4	1029	87.94	780	100
1,2,3,4	1046	89.40	780	100

As seen from Table 3, the time used for classification of features is less than a second for cosine distance and this can be attributed to the fact that the entire coefficient set obtained is reduced using standard deviation which results in dimensionality reduction of the feature vector and hence reduces the time taken for recognition.

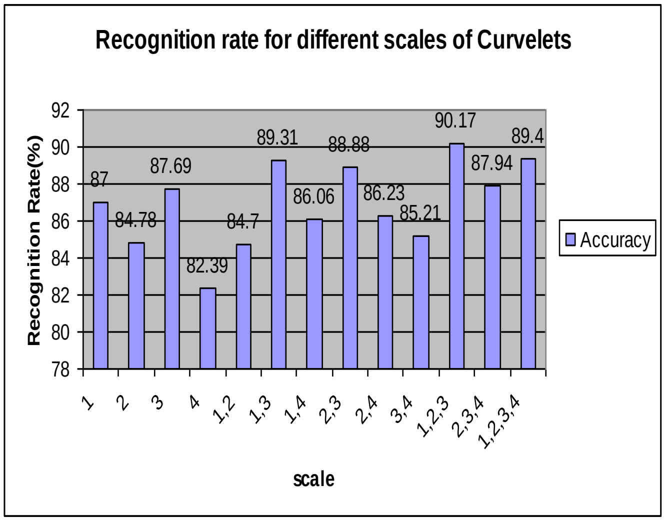


Fig 5: Graphical representation of recognition rate vs. scale

Table 3: The average recognition times for the training and testing samples

Scale	K value	Average time taken in seconds for testing data(seconds)	Average time taken in seconds for training data(seconds)
1	1	0.2507	0.2978
	4	0.2399	0.2352
	7	0.2397	0.2316
	10	0.2383	0.2310
	13	0.2417	0.2373
2	1	0.2709	0.2586
	4	0.2722	0.2540
	7	0.2687	0.2563
	10	0.2685	0.2542
	13	0.2751	0.2611
3	1	0.2609	0.2454
	4	0.2615	0.2483
	7	0.2598	0.2538
	10	0.2647	0.2476
	13	0.2626	0.2489
4	1	0.2730	0.2461
	4	0.2614	0.2455
	7	0.2648	0.2526
	10	0.2684	0.2453
	13	0.2596	0.2514
1,2	1	0.3171	0.2769
	4	0.3112	0.2805
	7	0.3240	0.2821
	10	0.3115	0.2754
	13	0.3149	0.2761
1,3	1	0.2917	0.2676
	4	0.2891	0.2669
	7	0.2911	0.2722
	10	0.2881	0.2658
	13	0.2919	0.2687
1,4	1	0.2898	0.2665
	4	0.2881	0.2649
	7	0.2901	0.2777
	10	0.2977	0.2660
	13	0.2891	0.2666
2,3	1	0.3333	0.2998
	4	0.3465	0.2990
	7	0.3330	0.3000
	10	0.3375	0.2964
	13	0.3395	0.2991
2,4	1	0.3311	0.2987
	4	0.3405	0.2967
	7	0.3342	0.3014
	10	0.3308	0.2965
	13	0.3314	0.3005
3,4	1	0.3187	0.2893
	4	0.3229	0.2863
	7	0.3180	0.2907
	10	0.3235	0.2877
	13	0.3203	0.2912
1,2,3	1	0.3563	0.3358
	4	0.3730	0.3316
	7	0.3566	0.3337
	10	0.3559	0.3312
	13	0.3622	0.3355
2,3,4	1	0.3780	0.3532
	4	0.3782	0.3466
	7	0.3782	0.3547
	10	0.3793	0.3518
	13	0.3824	0.3587
1,2,3,4	1	0.3977	0.3657
	4	0.3974	0.3665
	7	0.4086	0.3638
	10	0.3971	0.3620
	13	0.3961	0.3704

5. COMPARITIVE STUDY

The Table 4 shows the comparison results of existing methods with proposed method. From the comparative study it is seen recognition is done using few number of fonts. But in the proposed method 45 fonts with different sizes has been used.

6. CONCLUSION

In this paper, a new algorithm for recognition of vowels with different fonts and sizes in Kannada language using curvelet transform is proposed. In this approach, curvelet coefficients are used to create the feature vector used for recognition. Standard deviation is performed on the coefficients obtained in order to reduce the size of the feature vector. From the results obtained we can conclude that the curvelet transform along with standard deviation for dimensionality reduction can be used efficiently for OCR and all the coefficients obtained need not be used in the construction of the feature vector.

Table 2: The confusion matrix which shows the division of recognition of each vowel for the K values (K=1,4,7,10,13) and Scale-1,2,3 with Cosine distance

Vowels	ಅ	ಆ	ಇ	ಈ	ಉ	ಊ	ಋ	ಎ	ಏ	ಐ	ಒ	ಓ	ಔ
ಅ	73	5	0	1	0	0	0	1	0	6	0	0	4
ಆ	7	80	2	0	0	0	0	0	0	0	0	0	1
ಇ	0	0	87	0	0	0	0	1	1	0	0	0	1
ಈ	0	0	0	90	0	0	0	0	0	0	0	0	0
ಉ	1	0	0	0	88	0	1	0	0	0	0	0	0
ಊ	0	0	0	0	1	89	0	0	0	0	0	0	0
ಋ	0	0	0	0	1	0	89	0	0	0	0	0	0
ಎ	1	2	4	0	0	0	0	64	12	7	0	0	0
ಏ	0	1	3	0	0	0	0	6	67	13	0	0	0
ಐ	1	1	1	0	0	0	0	4	10	73	0	0	0
ಒ	0	0	0	0	0	0	0	1	1	2	84	2	0
ಓ	0	0	0	0	0	0	0	0	1	1	3	83	2
ಔ	0	1	0	0	0	0	0	0	1	0	0	0	88

7. REFERENCES

[1] Nagabhushan P, Angadi S.A, Anami B.S 2003 A Fuzzy Statistical Approach to Kannada Vowel Recognition based on Invariant Moments, proceedings of NCDAR -2003, PESCE, Mandya , pp 275-285.

[2] B.B. Chaudhuri, U. Pal 1998 A Complete Printed Bangla OCR System. Pattern Recognition, 31 (1998), pp 531–549.

[3] U. Pal, B.B. Chaudhuri 2004 Indian Script Character Recognition: a Survey. Pattern Recognition, 37 (2004),pp 1887 – 1899.

[4] M. Abdul Rahiman , M. S. Rajasree 2009 A Detailed Study and Analysis of OCR Research in South Indian Scripts.International Conference on Advances in Recent Technologies in Communication and Computing, pp.31-38.

[5] Ashwin T V, Sastry P S 2002 A fonts and size-independent OCR system for printed Kannada documents using support vector machines. Sadhana 27:35-58

[6] Nagbhushan P, Pai Radhika M 1999 Modified region decomposition method and optimal depth decomposition tree in the recognition of non –uniform sized characters – An experimentation with Kannada characters. Pattern Recognition .Letter. 20:1467-1475

[7] Kunte Sanjeev R, Sudhaker Samuel (2006), Hu's invariant moments & Zernike moments approach for the recognition of basic symbols in printed Kannada text. Sadhana vol .32, part 5, October 2007, pp. 521-533.

[8] R Sanjeev Kunte, R D Sudhaker Samuel 2007 An OCR system for printed Kannada text using Two-stage Multi-network classification approach employing Wavelet Features, International Conference on Computational Intelligence and Multimedia Applications 2007, pp 349-355.

[9] O.D. Trier, A.K. Jain, T. Taxt. 1996 Feature Extraction Methods for Character Recognition—a Survey. Pattern Recognition, 29(1996), pp 641–662.

[10] E. J. Cand`es and D. L. Donoho.2000 Curvelets – a surprisingly effective nonadaptive representation for objects with edges. In C. Rabut A. Cohen and L. L.

Schumaker, editors, Curves and Surfaces, pages 105–120, Vanderbilt University Press, 2000. Nashville, TN.

[11] Narasimha Murthy. A.V. 1975 Kannada Lipiya Ugama Mattu Vikasa. Institute of Kuvempu Kannada Studies Publication University of Mysore.

[12] Ishrat Jahan Sumana. 2008 Image Retrieval Using Discrete Curvelet Transform. Master Thesis. Gippsland School of Information Technology .Monash University, Australia.

[13] J. L. Starck, E. J. Cand`es, and D. L. Donoho 2002. The curvelet transform for image denoising.IEEE Trans. Im. Proc., 11-6 (2002), 670–684.

[14] H. Douma and M. V. de Hoop. 2004 Wave-character preserving prestack map migration using curvelets. Presentation at the Society of Exploration Geophysicists, Denver, CO, 2004.

[15] Emmanuel Cand`es, Laurent Demanet, David Donoho and Lexing Ying, “Fast Discrete Curvelet Transforms”, Technical report July 2005, revised March 2006.

[16] Farhad Mohamad Kazemi,Jalaleddin Izadian,Reihane Moravejian,Ehsan Mohamad Kazemi. 2008. Numeral Recognition Using Curvelet Transform. ACS International Conference on Computer Systems and Applications.pp 606-612.

[17] E. J. Candes and D. L. Donoho. 1999. Ridgelets: the key to higher-dimensional intermittency. Phil.Trans. R. Soc. Lond. A. 357 (1999), 2495–2509.

[18] M. J. Fadili and J.-L. Starck, 2007.Curvelets and Ridgelets," Encyclopedia of Complexity and System Science , in press.

[19] S. Theodoridis, K. Koutroumbas.1999 .Pattern Recognition, Academic Press, New York.

[20] The Wikipedia website.[Online].Available: http://www.wikipedia.org/

[21] Mallikarjun Hangarge,Shashikala Patil,B.V.Dhandra 2010. Multi-font/size Kannada Vowels and Numerals Recognition Based on Modified Invariant Moments IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition RTIPPR, 2010,pp126-130

[22] Dinesh Acharya U, N V Subba Reddy and Krishnamoorthi 2008 .Hierarchical Recognition System for Machine Printed Kannada Characters.IJCSNS International Journal of Computer Science and Network Security,VOL.8 No.11, pp.44-53, November-2008.

[23] B.V. Dhandra, Mallikarjun Hangarge, Gururaj Mukarambi.2011. Spatial Features for Multi-Font/Multi-Size Kannada Numerals and Vowels Recognition. "International Conference on Communication, Computation, Control and Nanotechnology (2010).

[24] Karthik Sheshadri, Pavan Kumar T Ambekar, Deeksha Padma Prasad and Dr. Ramakanth P Kumar.2010. An OCR System for Printed Kannada using K-Means Clustering.2010 IEEE International Conference on Industrial Technology (ICIT),pp.183-187.

Table 4 Comparative results for printed Kannada characters with other existing methods

Authors	Characters considered	No. of samples in data set	Feature extraction method	Classifier	Accuracy (%)
Mallikarjun Hangarge et al[21]	**Vowels 7 fonts**	**1800**	**Modified invariant moment**	**Nearest neighbor**	**97.7**
T V Ashwin et al [5]	**All**	**3000**	**Modified structural**	**H-SVM**	**86.11**
			Zernike Moments	**H-SVM**	**86.09**
Dinesh Achraya et al[22]	**All**	**8400**	**Stroke and topological**	**Decision tree and K-NN**	**92.68**
	Vowels	**-**			**92.32**
R.Sanjeev Kunte et al[8]	**All**	**15,000**	**Wavelets**	**Two-stage neural**	**91**
B.V Dhandra et al[23]	**Vowels 7 fonts**	**1400**	**Spatial**	**SVM**	**90.64**
R.Sanjeev Kunte et al[7]	**Basic symbols**	**4500**	**Hu’s moments**	**RBF Neural**	**82**
Karthik Sheshadri et al[24]	**All 4 fonts**	**-**	**No.of black pixels**	**K-Means**	**95**
Proposed	**Vowels 45 fonts**	**5850**	**Curvelets with Std**	**K-NN**	**90.17**