INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Two Stage Multi Modal Deep Learning Kannada Character Recognition Model Adaptive to Discriminative Patterns of Kannada Characters

S R Siddanna[1,2]*, **Y C Kiran**[2]

1 Department of Information Science and Engg, SJB Institute of Technology, Bangalore, 560060, Karnataka, India
2 Department of Information Science and Engg, Global Academy of Technology, , Bangalore, 560098, Karnataka, India

OPEN ACCESS

Received: 21-09-2022
Accepted: 14-12-2022
Published: 20-01-2023

Citation: Siddanna SR, Kiran YC (2023) Two Stage Multi Modal Deep Learning Kannada Character Recognition Model Adaptive to Discriminative Patterns of Kannada Characters. Indian Journal of Science and Technology 16(3): 155-166. https://doi.org/10.17485/IJST/v16i3.1904

* **Corresponding author**.

srsiddanna@sjbit.edu.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: Designing optical character recognition systems for Kannada character is challenging due to higher self-similarity in characters and higher number of character classes. This work addresses the two major problems of reduced accuracy and higher false positives due to higher self-similarity in characters. **Methods**: This work proposes a two stage multi modal deep learning technique to handle the complexity in Kannada character recognition. The characters are first grouped based on morphological and structural similarity. A novel morphological/structural difference maximization convolution kernel based deep learning modal is trained for each of character group to recognize characters in that group. This divide and conquer strategy reduce the complexity of deep learning model in learning discriminative features for Kannada character recognition. **Findings**: The proposed two stage multi modal deep learning provides 89% recognition accuracy which is at least 6% higher compared to existing works. The false positives in proposed solution are at least10% lower compared to existing works. **Novelty**: A novel sector intensity distribution feature specific to curve structure of Kannada characters for deciding the number of groups of characters. Classifier is designed for each group of characters. Classification is done using convolutional neural network with a novel morphological/structural difference maximization convolution kernel to solve the structural similarity problem in Kannada character recognition.

Keywords: Handwritten character Recognition; Deep Learning; Machine Learning; Pattern Recognition; Fuzzy Gaussian

1 Introduction

Optical character recognition systems (OCR) are gaining rapid popularity in document digitization applications. Recently, India is facing a rapid digitization drive for local languages. Government is biggest consumer for these services. Huge volumes of handwritten documents to converted to digitized form. OCR systems with higher accuracy are needed for Indian local languages. This has created huge research interest. Motivated by it, this work explores the OCR for Kannada character recognition. Kannada is official language of Karnataka state of India. It uses Brahmi script. The language has 49 phonemic letters. The letters are in two groups: 15 vowels, 34 consonants. Modifier glyphs (half letters) from vowels are combined with consonants (15*34=510) to from a total of (510+34) 544 characters. The vowels and consonants in Kannada language are given in Figure 1. In addition to it, with extra modifier for each of the consonant called as consonant conjuncts, there are totally 18511 distinct characters in this language (544*34+ 15= 18511). Some of the consonant conjuncts are given in Figure 3.

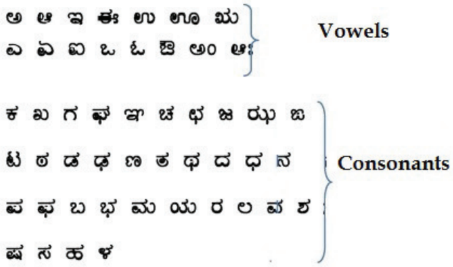


Fig 1. Vowels and Consonants in Kannada

ಕ ಕಾ ಕಿ ಕೀ ಕು ಕೂ ಕೃ ಕೄ ಕೆ ಕೇ ಕೈ ಕೊ ಕೋ ಕೌ ಕಂ ಕಃ

Fig 2. Sample Vowel consonant combination

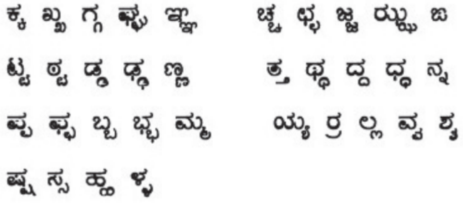

ಕ್ಕ ಖ್ಖ ಗ್ಗ ಘ್ಘ ಙ್ಙ ಚ್ಚ ಛ್ಛ ಜ್ಜ ಝ್ಝ ಞ
ಟ್ಟ ಠ್ಠ ಡ್ಡ ಢ್ಢ ಣ್ಣ ತ್ತ ಥ್ಥ ದ್ದ ಧ್ಧ ನ್ನ
ಪ್ಪ ಫ್ಫ ಬ್ಬ ಭ್ಭ ಮ್ಮ ಯ್ಯ ರ್ರ ಲ್ಲ ವ್ವ ಶ್ಶ
ಷ್ಷ ಸ್ಸ ಹ್ಹ ಳ್ಳ

Fig 3. Sample consonant conjuncts

Developing highly accurate OCR systems for Kannada language is difficult due to higher volume of characters (18511) and its structural complexity. The characters are mostly curves than straight or slant lines. Some shapes are wider and some are longer. Many characters have higher similarity. A sample of character grouped on similarity is listed in Figure 4.

Skipping the consonant conjuncts, OCR for 544 characters (510 vowel consonant + 34 consonant) is also challenging due to the problems of : structural complexity, similarity and curves with variation in regional density. Developing a single classifier model to recognize all the 544 characters with higher accuracy is challenging.

In[1], deep learning was used for Kannada characters. Author solved the problem of self similarity in characters using transfer learning. But the approach classified only vowels, consonants and numerals in Kannada. In[2], convolutional neural network features extracted from Kannada characters were used for classification. Author increased the training volume through augmentation, so as to increase the accuracy. But the work did not address self similarity in characters. VGGNet19 was used in[3] for Kannada character recognition. But without consideration for self similarity in characters, the method was able to achieve only about 70% classification accuracy. Convolutional neural network (CNN) features were used in[4] for Kannada character recognition, but it was tested only for limited classes of vowels. CNN was used in[5] for Kannada character recognition but accuracy was less than 73%. CNN used in[6] did not address the self similarity in Kannada characters in recognition stage. In[7], optimized CNN was used for Kannada numeral recognition. Though CNN features were found to be better in character recognition compared to hand crafter features, they had two important problems. Most existing OCR systems could

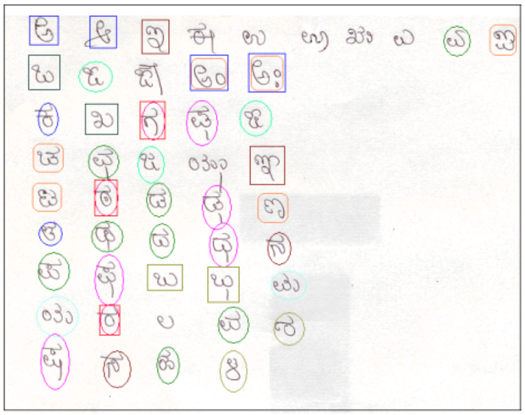


Fig 4. Similar characters are grouped.

not differentiate the Kannada characters with higher structural similarity. Also the classifier complexity is higher for recognition of 544 character classes with most of them having high structural similarity. Due to the complexity, accuracy falls and false positives increases. Convolutional neural network (CNN) was used to recognize handwritten Hindi characters in [8]. Four layer CNN followed by three layer recognition was used in this work. Hindi character images are taken as input and features are learnt at convolutional layers. But the method is not suitable complex character patterns in Kannada language. In [9] experimentation was done with deep learning models of: AlexNet, Densenet121, Vgg11, Vgg16, Vgg19 and Inception V3 for recognition of Devanagari characters. Inception V3 found to perform better compared to other models. Higher layers increased the accuracy. But considering the complexity of Kannada characters, a single model needs higher number of layers and this can create learning bias resulting in large difference in accuracy for the classes. Work in [10] extracted features of chain coding, edge detection using gradient features and direction features from Devanagari handwritten image. The features are reduced using Linear Discriminant Analysis (LDA) features. The reduced features are used to recognize the characters using SVM classifier. But these features fail for curved character patterns like Kannada characters. Work in [11] experimented with three different deep learning models of: CNN, Modified Lenet CNN and AlexNet CNN. Non linearity is introduced to handle non-linearity of Devanagari characters, in all the three models using rectified linear units. Though all three models achieved more than 90% accuracy, they were tested with limited classes and their suitability in handling complex character patterns were not tested. Hu's seven variations and Zernike moments were extracted from the handwritten Kannada characters and KNN classifier is used for character recognition in [12]. The method has higher false positives. Work in [13] extracted gradients are represented as histogram of gradients and classified to Brahmi characters using SVM classifier. The method was tested only for limited class of characters. In [14] histogram features were used for character recognition, but the false positive is very high in this approach. CNN was used in combination with transfer learning to recognize Kannada characters in [15]. Transfer learning was used to solve the problem of similarity in characters. But with same CNN model, transfer learning was not able to solve the feature generalization problem. Due to this accuracy reduced. Work in [16] used hidden markov model (HMM) to recognize handwritten bangla characters using stroke features. Point based and curvature based stoke features are extracted, and they are classified using HMM. This work was extended in [16] replacing HMM with LSTM and Bi-directional LSTM. But both approaches failed to model the stokes for cursive character patterns which is more prominent in Kannada character set. Work in [17] addressed the problem of discriminating region selection is classifying similar characters. But the method was tested only for Assamese and English letters. Work in [18] analyzed the impact of pre-processing methods for Kannada character recognition. The study inferred that moving average filter with window span of 3 improved the recognition accuracy of Kannada characters. In [19] stoke features were extracted using self controlled Ramer-Douglas-Peucker (RDP) algorithm and stroke features classified to characters using one dimensional CNN. The method was tested for Gurmukhi scripts, whose complexity is less compared to Kannada scripts. Convolutional auto encoder with generative adversarial network was used in [20] for recognition of Devanagari characters. The character recognition problem was solved like face recognition with improvised adversarial learning. Intricate features were not learnt for fine level classification. Without intricate feature learning, it becomes difficult to differentiate character with higher self similarity like in Kannada language.

From the survey, most character recognition approaches for Indian characters sets used single classifier model through they attempted different features like strokes, gradient histogram, pattern intersection features. For complex and high structural similarity, more discriminative features cannot be learnt and classified by single classifier model. Also existing works on

Kannada character recognition has two important problems (i) reducing the classifier complexity for large character classes and increase the accuracy (ii) reduce the false positives in presence of self similar characters.

Addressing the above two problems, this work proposes a divide & conquer strategy with two stage process. In the first stage, the 544 characters are split into many groups based on higher structural similarity. In the second stage, a deep learning convolutional neural network with convolution kernel customized to learn more discriminative features specific to particular group of characters is trained to recognize the characters in that group. By using a divide and conquer strategy, the proposed solution solves the self similarity in character and higher false positive in a better way compared to existing works. Following are the novel contributions of this paper work.

1. A novel sector intensity distribution feature specific to curve structure of Kannada characters for deciding the number of groups of characters.
2. A novel fuzzy Gaussian membership built on center of mass feature to group similar characters
3. A novel morphological/structural difference maximization convolution kernel based convolutional neural network trained for each group to recognize the characters in the group

The rest of the paper is organized as follows. Section II presents the survey on approaches for character recognitions specific to Indian languages. Section III presents the proposed multi-modal deep learning model for Kannada character recognition. Section IV presents the proposed solution results and comparison to state of art existing works. Section V presents the conclusion and scope for future research.

2 Methodology

2.1 Two stage multi-modal deep learning character recognition

The architecture of the two stage multi modal deep learning character recognition is given in Figure 6. The solution has two stages. In the first stage, the characters are grouped based on the structural similarity into k group of characters. A Fuzzy Gaussian membership function is designed to associate the unknown characters to the established k group of characters. In the second stage, CNN with discriminative feature learning and goal of structural difference maximization is designed for each of the k groups. With this two stage multi-modal deep learning technique, the complexities in Kannada character recognition is solved as divide and conquer strategy. With multiple CNN models adapted for structural difference maximization, the CNN feature learning is more directed to identifying more discriminative features on fine differences in the similar group of characters. Each of the stages of the proposed solution is detailed below.

2.1.1 Grouping Characters

Based on the observation of curved structures in most of the Kannada characters, a novel sector intensity distribution feature is proposed in this work. A structure of concentric circles cover the character and each of the concentric circle split to sector is placed over the character image. The structure is shown in Figure 4.

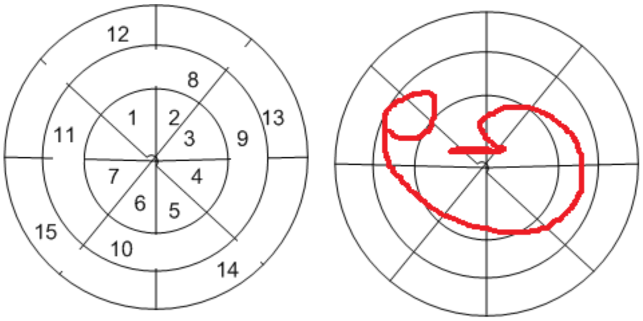


Fig 5. (a) Concentric structure, (b) Letter over structure

There are 15 sectors. The total number of pixels in each of the 15 sector is divided by the total number of pixel. The normalized feature vector corresponding to character is given as

$$F = < f_1, f_2, \ldots f_{15} > \quad (1)$$

Where

$$f_x = \frac{num\,of\ pixel\ in\ sector\ x}{Total\ number\ of\ pixels\ in\ image} \tag{2}$$

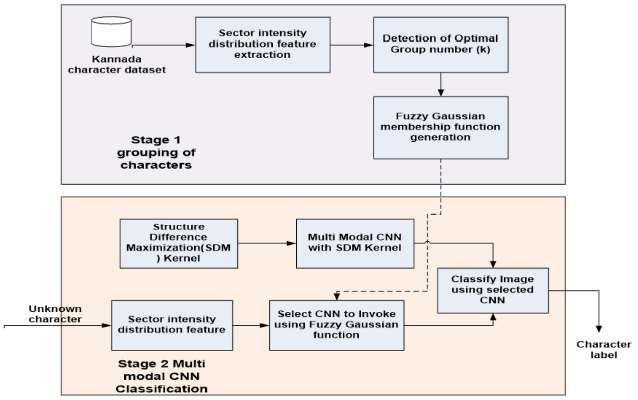

Kannada
character dataset
Sector intensity distribution feature extraction
Detection of Optimal Group number (k)
Fuzzy Gaussian membership function generation
Stage 1 grouping of characters
Structure Difference Maximization(SDM) Kernel
Multi Modal CNN with SDM Kernel
Classify Image using selected CNN
Unknown character
Sector intensity distribution feature
Select CNN to Invoke using Fuzzy Gaussian function
Character label
Stage 2 Multi modal CNN Classification

Fig 6. Architecture of two stage multimodal character recognition

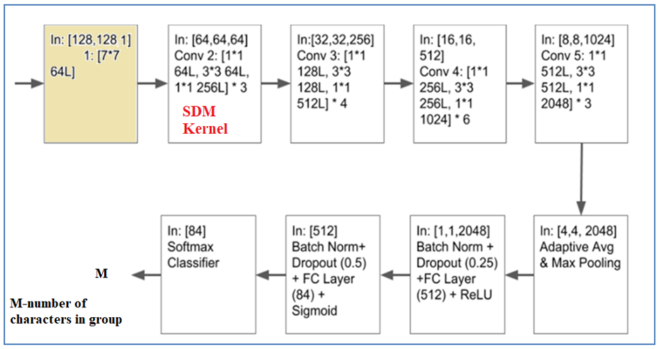

In: [128,128 1] 1: [7*7 64L]
In: [64,64,64] Conv 2: [1*1 64L, 3*3 64L, 1*1 256L] * 3
SDM Kernel
In:[32,32,256] Conv 3: [1*1 128L, 3*3 128L, 1*1 512L] * 4
In: [16,16, 512] Conv 4: [1*1 256L, 3*3 256L, 1*1 1024] * 6
In: [8,8,1024] Conv 5: 1*1 512L, 3*3 512L, 1*1 2048] * 3
In: [4,4, 2048] Adaptive Avg & Max Pooling
In: [1,1,2048] Batch Norm + Dropout (0.25) +FC Layer (512) + ReLU
In: [512] Batch Norm+ Dropout (0.5) + FC Layer (84) + Sigmoid
In: [84] Softmax Classifier
M
M-number of characters in group

Fig 7. Architecture of Multi modal CNN

The features are extracted for all the 544 characters and the features must be grouped to *k* groups. Elbow method is used for finding the *k* value. A graph is plotted by calculating the cost of clustering in terms of average distance between cluster points to its centroids for different values of *k*. As the *k* value increases, the cost drops. At certain k value, a elbow (similar to human hand) appears and this *k*value is selected as the optimal number of clusters. An sample elbow point formation is shown in Figure 8.

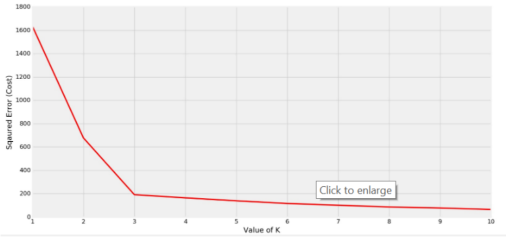


Fig 8. Elbow method

In the Figure 7, the elbow point is formed at $k = 3$ and thus the optimal number of cluster is 3.

The dataset of Kannada character images is passed to the feature extraction stage and sector intensity distribution feature vector is extracted for each character image. The dataset is clustered using Fuzzy C Means clustering with number of cluster as found using Elbow method. The centroid of cluster after clustering is given as

$D = \{ D_{e,q} \ , \ e = 1,2\ldots k \ and \ q = 1,2,3,..nooffeature\}$

$D_{e,q}$ is the coordinate of the cluster. Gaussian function [28] is used to find the closeness of the rth of the data in terms of its qth coordinate.

$$G\left(f_{r,q}, D_{e,q}, \sigma_{e,q}\right) = e^{\frac{\left(f_{r,q} - D_{e,q}\right)^{2}}{\sigma_{e,p}{}^{2}}} \tag{3}$$

Where

$$\sigma_{e,q} = \frac{1}{N_e}\sum_{r=1}^{N_e}\left(f_{r,q} - D_{e,q}\right)^{2} \tag{4}$$

The closeness in terms of all coordinates is defined in terms of closeness of qth coordinates as

$$\Psi_{\mathrm{r,e}} = \prod_{\mathrm{q=1}}^{\mathrm{P}} \mathrm{G}\left(\mathrm{f_{r,q}}, \mathrm{D_{e,q}}, \sigma_{\mathrm{e,q}}\right) \tag{5}$$

The label for the eth cluster is formed as linear regression of input features as

$$\Phi_{\mathrm{r,e}} = \mathrm{W_{e,0}} + \sum_{\mathrm{q=1}}^{\mathrm{p}} \mathrm{W_{e,q,f_{r,q}}} \tag{6}$$

Where W is the regression coefficients for each of the feature coordinate. The data point r has membership to all cluster with varying degree. Thus final label for the rth data point is found by weighted function over membership of each cluster as

$$\bar{N}\left(r\right) = \ \sum_{e=1}^{P} \Psi_{r,e}\Phi_{r,e} \tag{7}$$

The error between the $\bar{N}\left(r\right)$ and the $N\left(r\right)$ which is found using training is calculated as

$$E = \sum_{r=1}^{N} | \left(\bar{N}\left(r\right) - N\left(r\right) \right| |^{2} \tag{8}$$

To reduce the error, the optimum values for Gaussian parameters $(D_{e,q}, \sigma_{e,q})$ and the regression coefficients $(W_{e,p})$ is found using gradient descent method.

$$D_{e,q}\left(t+1\right) = D_{e,q}\left(t\right) + \ \eta_{C}\frac{\partial E}{\partial D_{e,q}} \tag{9}$$

$$\sigma_{e,q}\left(t+1\right) = \sigma_{e,q}\left(t\right) + \ \eta_{\sigma}\frac{\partial E}{\partial \sigma_{e,q}} \tag{10}$$

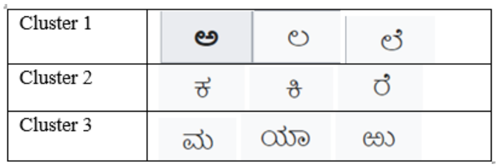
$$W_{e,q}\left(t+1\right) = W_{e,q}\left(t\right) + \ \eta_{W}\frac{\partial E}{\partial W_{e,q}} \tag{11}$$

In the equations (7), (8) and (9), the iteration count is given as t and learning parameters for Gaussian parameters and regression coefficients are η_C, η_σ, η_W. The iteration stops when desired error threshold is reached. The Fuzzy Gaussian membership function for each of the cluster in terms of features of the data point is thus found for k cluster. For any unknown character, its sector intensity distribution feature is extracted and fuzzy membership function $\Phi_{r,e}$ is calculated for each of the k class and the character is associated to the group for which $\Phi_{r,e}$ is maximum.

Some of the characters in each group are illustrated in Figure 9

Grouping is achieved by the Gaussian membership function constructed on features extracted as in equation 1. The Gaussian membership function on features gives a value which is almost close by when the features are similar. By this way, similar characters in terms of features given in equation 1, falls into same group. The range of values for the Gaussian membership function for each cluster when tested for 657 different Kannada characters is given in Table 1.

From Table 1, it can be seen that there is diverse range difference between the clusters in terms of their Gaussian function values indicating a good separation between the clusters.



Cluster 1	ಅ	ಲ	ಲೆ
Cluster 2	ಕ	ಕಿ	ರೆ
Cluster 3	ಮ	ಯಾ	ಋ

Fig 9. Character grouping result

Table 1. Gaussian function values

Cluster	Min	Max	Average	Std deviation
1	310	340	324	3.45
2	200	260	235	5.78
3	530	590	570	7.89

2.2 Multi modal deep learning character recognition

Deep learning CNN model is adapted to learn more intricate features to differentiate the subtle differences in similar group of characters. This CNN is trained for each of k class. Thus complexity in char recognition using single classifier model is solved as multi-modal CNN classifier with one CNN for each group of characters. The architecture of the multi-modal CNN is shown in Figure 7. CNN have the capability to learn more intricate features due to use of its convolutional kernels. But the structural similarity cannot be learnt in detail with default convolutional kernel. To overcome this problem, this work proposes a CNN model with novel convolution kernel called structural difference maximization (SDM) kernel. This kernel magnifies the significant areas for patterns, so that the features in those significant layers become amplified in subsequent feature learning. By this way, the classification accuracy is improved. A binary mask is constructed for each group of characters. This mask is constructed by finding the sectors of higher similarity in the character group and setting 1 for that sector and 0 for rest of the sectors. The SDM kernel convolution is applied by taking the image of size 64×64 and computing 8 local binary pattern LBP for it. Each of LBP result and binary mask are joined with logical AND operation. Each of the 8 result after AND is then convolved with 7×7 kernel and summed up to get the output feature map.

$$C(q) = \sum_{m=1}^{M} \sum_{n=1}^{N} AND\left(LBP(q), mask(m)\right).K(j) \tag{12}$$

Where $M$the number of times of masking is, N is the number of LBP and $mask(m)$ is the binary mask applied to m^{th} LBP pattern.

The CNN model given in Figure 7 is trained for each group of characters to recognize each of M different character in the group. Due to structural difference maximization (SDM) kernel in the first convolutional layer, the structural differences are amplified and this becomes further amplified in the successive convolutional layers. The multi modal CNN model is trained with character images and their corresponding labels for each group. The trained multi modal CNN model is then used for character recognition.

For an unknown character to be recognized, its sector intensity distribution feature is extracted first. Applying the fuzzy Gaussian membership function, the group and the corresponding CNN model to be invoked for the group is found. The character image is then passed as input to the corresponding CNN model to get the character label as the output.

For a unknown character to be recognized, its sector intensity distribution feature is extracted first. Applying the fuzzy Gaussian membership function, the group and the corresponding CNN model to be invoked for the group is found. The character image is then passed as input to the corresponding CNN model to get the character label as the output.

The overall flow of algorithm for detection of unknown character is given in pseudo code below

```
Algorithm:  Recognize Characters
Input: Image of character
Output: character name
1. Extract sector wise feature vectors of input image as in equation 1
2. Invoke equation 6 for three clusters and find the cluster with maximum value for $\Phi_{r,e}$
3. Result ←Classify the image using the CNN corresponding to $\Phi_{r,e}$
4. Return result
```

3 Results and Discussion

The performance of the proposed solution is experimented against Kannada handwritten characters dataset from Kaggle[21]. The dataset has 16,425 images with 657 classes. Each class has 25 images. The performance is measured for three categories of Vowels, Consonants and Vowel-consonant combination. The performance is measured in terms of: recognition accuracy and false positives. The performance is compared against Modified Lenet proposed by Prashanth et al.[11], image gradient with SVM proposed by Kaur et al[13] and RDP approach proposed in Singh et al[19].

The recognition accuracy is measured for vowels and the result is given in Table 2.

Table 2. Recognition accuracy for vowels

Accuracy	Proposed	Prashanth et al.[11]	Kaur et al.[13]	Singh et al.[19]
Maximum	0.9558	0.8654	0.8571	0.934
Minimum	0.8385	0.7701	0.7208	0.832
Average	0.9427	0.8549	0.8436	0.904
Std. Deviation	0.0168	0.0169	0.0204	0.017

The false positives are measured for vowels and the result is given in Table 3.

Table 3. False positives for vowels

False positives	Proposed	Prashanth et al.[11]	Kaur et al.[13]	Singh et al.[19]
Maximum	0.26	0.34	0.35	0.29
Minimum	0.15	0.27	0.29	0.25
Average	0.18	0.30	0.32	0.26
Std. Deviation	0.015	0.031	0.03	0.01

The average recognition accuracy for Kannada vowels in proposed solution is at least 9% higher compared to Prashanth et al., 10% higher compared to Kaur et al. and 4% higher compared to Singh et al. The false positives for Kannada vowel in proposed solution is almost 12% lower compared to Prashanth et al, 14% lower compared to Kaur et al and 8% lower compared to Singh et al.

The recognition accuracy is measured for consonants and the result is given in Table 4.

Table 4. Recognition accuracy for consonants

Accuracy	Proposed	Prashanth et al.[11]	Kaur et al. .[13]	Singh et al.[19]
Maximum	0.9418	0.8454	0.80	0.90
Minimum	0.821	0.7301	0.71	0.80
Average	0.931	0.8149	0.77	0.87
Std. Deviation	0.017	0.015	0.03	0.03

The false positives are measured for consonants and the result is given in Table 5.

Table 5. False positives for consonants

False positives	Proposed	Prashanth et al.[11]	Kaur et al.[13]	Singh et al.[19]
Maximum	0.29	0.35	0.37	0.32
Minimum	0.18	0.28	0.31	0.28
Average	0.20	0.32	0.34	0.30
Std. Deviation	0.02	0.03	0.03	0.01

The average recognition accuracy for Kannada consonants in proposed solution is at least 12% higher compared to Prashanth et al., 16% Kaur et al. compared to image gradient and 6% higher compared to Singh et al. The false positives for Kannada vowel in proposed solution is almost 12% lower compared to Prashanth et al., 14% Kaur et al. compared to image gradient and 10% lower compared to Singh et al. Compared to Vowels, the recognition accuracy in proposed solution for consonants has reduced by 1%. The recognition accuracy is measured for Vowel consonants combination and the result is given in Table 6.

Table 6. Recognition accuracy for vowel consonants

Accuracy	Proposed	Prashanth et al.[11]	Kaur et al.[13]	Singh et al.[19]
Maximum	0.9218	0.8254	0.76	0.86
Minimum	0.831	0.7101	0.71	0.79
Average	0.89	0.791	0.73	0.83
Std. Deviation	0.02	0.02	0.01	0.02

Table 7. False positives for vowel consonants

False positives	Proposed	Prashanth et al.[11]	Kaur et al[13]	Singh et al[19]
Maximum	0.32	0.38	0.40	0.35
Minimum	0.19	0.32	0.36	0.31
Average	0.23	0.34	0.39	0.33
Std. Deviation	0.03	0.03	0.02	0.02

The false positives are measured for Vowel consonant combination and the result is given in Table 7.

The average recognition accuracy for Kannada vowel consonant combination in proposed solution is at least 10% higher compared to Prashanth et al, 16% higher compared to Kaur et al. and 6% higher compared to Singh et al. The false positives for Kannada vowel consonant combination in proposed solution is almost 11% lower compared to Prashanth et al.[11], 16% lower compared to Kaur et al[13] and 10% lower compared to Singh et al[19]. Compared to Vowels, the recognition accuracy in proposed solution for consonants has reduced by 1%. The average accuracy over three categories of vowels, consonants and vowel consonant combination is shown in Figure 10.

The average accuracy has increased in proposed solution compared to others due to use of two stage processing and reducing the learning bias at CNN. Though Prashanth et al. using CNN, it was a single model for all classifiers and this prevented it from learning discriminating features for similar characters. Kaur et al. could not perform well curve difference in characters. Singh et al though performed well compared to other approaches, it could not solve the single classifier bias problem and thus the accuracy in it is lower than the proposed solution.

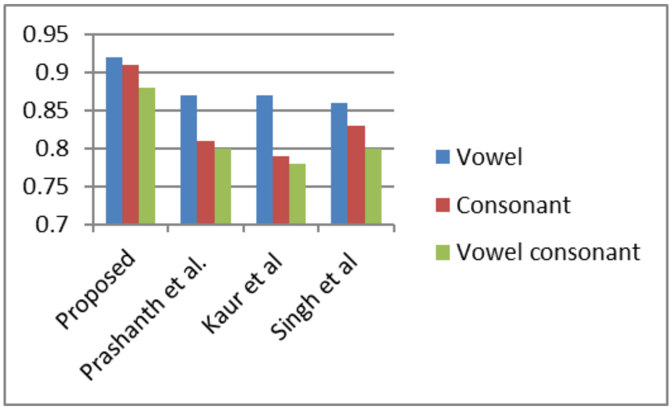


Fig 10. Comparison of accuracy

The maximum accuracy is achieved for Vowels followed by consonant, vowel consonant combination in all the solutions. This is inline with character complexity. Character complexity and structural similarity is higher in vowel combination letters. But even in that case, the proposed solution achieved 89% accuracy which is atleast 6% higher compared to existing works. The false positives are higher in Vowel consonant followed by consonant and vowels. But the proposed solution has false positives at 23%, which is atleast 10% lower compared to existing works. The accuracy gain due to SDM kernel used in proposed solution is compared against CNN with default kernel and the result is given in Table 8.

The SDM kernel has increased the average accuracy in proposed solution by 13% compared to default kernel. It has increased accuracy for vowels by 11%, consonants by 14% and vowel consonants by 13%.

The CNN configuration used for classifying the characters in each group is given in Table 9.

Table 8. Comparison of accuracy across kernels

Accuracy	SDM Kernel	Default Kernel
Vowel	0.94	0.83
Consonant	0.92	0.79
Vowel consonant	0.89	0.76

Table 9. CNN Configuration for classification

Layer	Configuration
Convolutional 1D	10*128, ReLU, Stride=2
MaxPool 1D	Size=2,Stride=2
Convolutional 1D	10*128, ReLU, Stride=2
MaxPool 1D	Size=2,Stride=2
Convolutional 1D	8*128, ReLU, Stride=2
Convolutional 1D	8*128, ReLU, Stride=2
Flatten	-
Dense	1*512,ReLU
Output Softmax layer	(1 neuron for each character in the group)

The average training accuracy and loss of CNN is given in Figures 11 and 12.

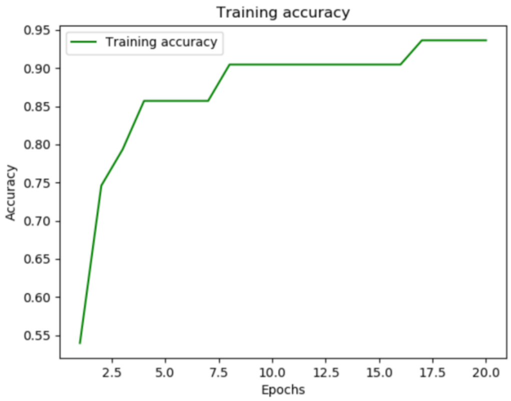


Fig 11. Training accuracy

At the epoch of 20, maximum accuracy of 94% is achieved by the CNN. The lowest loss of 0.22 is achieved for epoch of 20.

4 Conclusion

This work proposed a novel two stage multi-modal deep learning character recognition for Kannada characters. The complexities in Kannada character regions in terms of structural complexity; structural similarity and difficulty in learning discriminating features are solved applying divide and conquer strategy in two stages. In the first stage, the characters are grouped based on a novel feature extraction algorithm. In the second stage, for each group of character, s convolutional neural network is trained. The convolutional neural network was trained with a structural difference maximization kernel to solve the problem of structural similarity in recognition of characters. These novelties helped the proposed solution to perform better compared to existing works. The method can provide 89% recognition accuracy which is at least 6% higher compared to existing works. The false positives in proposed solution are at least10% lower compared to existing works. Extending the work, by improving the character grouping strategy is in scope of future work.

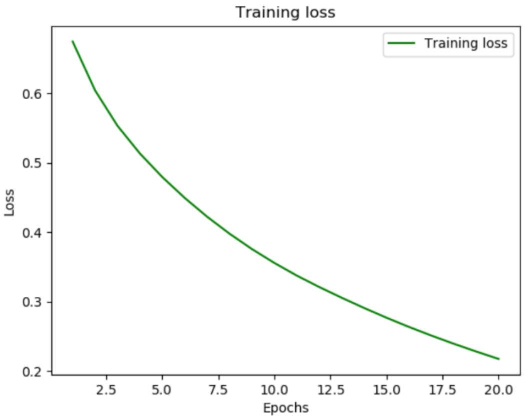


Fig 12. Training loss

References

1) Hegde P, Rajath S, Shwetha D, Sindhu C, Ravi P. Handwritten Character Recognition of Kannada Language Using Convolutional Neural Networks and Transfer Learning. 2021. Available from: https://doi.org/10.1088/1757-899X/1110/1/012003.
2) Rao AS, S S, K A, Arpitha, Nayak C, Meghana, et al. Exploring Deep Learning Techniques for Kannada Handwritten Character Recognition: A Boon for Digitization. 2020. Available from: http://sersc.org/journals/index.php/IJAST/article/view/25189.
3) Rani NS, Subramani AC, P AK, Pushpa BR. Deep Learning Network Architecture based Kannada Handwritten Character Recognition. *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2020. Available from: https://doi.org/10.1109/ICIRCA48905.2020.9183160.
4) Ravikumar M, Sampathkumar S. Recognition of Kannada Handwritten Words from Answer Scripts Using Machine Learning Approaches.Information and Communication Technology for Competitive Strategies (ICTCS 2020). *Lecture Notes in Networks and Systems*. 2022;191. Available from: https://doi.org/10.1007/978-981-16-0739-4_99.
5) Ramesh G, Sharma GN, Balaji JM, Champa HN. Offline Kannada Handwritten Character Recognition Using Convolutional Neural Networks. *2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*. 2019;p. 1–5. Available from: https://doi.org/10.1109/WIECON-ECE48653.2019.9019914.
6) Gandhana MH, Lakshman D, Naik. Online Kannada Handwritten Characters and Numerical Recognition using CNN Classifier. *International Journal Of Engineering Research & Technology* . 2021;10(07). Available from: https://www.ijert.org/online-kannada-handwritten-characters-and-numerical-recognition-using-cnn-classifier.
7) Upadhye GD, Kulkarni UV, Mane DT. Improved Model Configuration Strategies for Kannada Handwritten Numeral Recognition. *Image Analysis & Stereology*. 2021;40(3):181–191. Available from: https://doi.org/10.5566/ias.2586.
8) Baranidharana B, Kandpal A, Chakravorty A. Hindi Handwritten Character Recognition using CNN. *International Journal of Advanced Science and Technology*. 2020;29(06):58–66. Available from: http://sersc.org/journals/index.php/IJAST/article/view/11294.
9) Aneja N, Aneja S. Transfer Learning using CNN for Handwritten Devanagari Character Recognition. *1st International Conference on Advances in Information Technology (ICAIT)*. 2019;p. 293–296. Available from: https://doi.org/10.1109/ICAIT47043.2019.8987286.
10) Shitole S, Jadhav S. Recognition of handwritten devanagari characters using linear discriminant analysis. In: 2nd International Conference on Inventive Systems and Control (ICISC). IEEE. 2018;p. 100–103. Available from: https://doi.org/10.1109/ICISC.2018.8398991.
11) Prashanth DS, Mehta RVK, Ramana K, Bhaskar V. Handwritten Devanagari Character Recognition Using Modified Lenet and Alexnet Convolution Neural Networks. *Wireless Personal Communications*. 2022;122(1):349–378. Available from: https://doi.org/10.1007/s11277-021-08903-4.
12) Prashanth DS, Panini CN. KNN classification of Kannada Characters using Hu's Seven Variants and Zernike Moment. 2017. Available from: https://www.researchgate.net/publication/312495850.
13) Kaur S, Sagar BB. Brahmi character recognition based on SVM (support vector machine) classifier using image gradient features. *Journal of Discrete Mathematical Sciences and Cryptography*. 2019;22(8):1365–1381. Available from: https://doi.org/10.1080/09720529.2019.1692445.
14) Parekh KA, Goswami MM, Mitra SK. Handwritten Numeral Recognition Using Polar Histogram of Low-Level Stroke Features. *Proceedings of 3rd International Conference on Computer Vision and Image Processing*. 2020;p. 169–181. Available from: https://doi.org/10.1007/978-981-32-9088-4_15.
15) Parikshith H, Rajath SMN, Shwetha D, Sindhu CM, Ravi P. Handwritten Character Recognition of Kannada Language Using Convolutional Neural Networks and Transfer Learning. *IOP Conference Series: Materials Science and Engineering*. 2021;1110(1):012003. Available from: https://doi.org/10.1088/1757-899X/1110/1/012003.
16) Bhattacharya N, Roy PP, Pal U, Setua SK. Online Bangla handwritten word recognition. *Malaysian Journal of Computer Science*. 2018;31(4):300–310. Available from: https://doi.org/10.22452/mjcs.vol31no4.4.
17) Mandal S, Prasanna SRM, Sundaram S. An improved discriminative region selection methodology for online handwriting recognition. *International Journal on Document Analysis and Recognition (IJDAR)*. 2019;22(1):1–14.
18) Ramya S, Shama K. The Effect of Pre-processing and Testing Methods on Online Kannada Handwriting Recognition: Studies Using Signal Processing and Statistical Techniques. *Pertanika J Sci & Technol*. 2018;26(2):671–690.
19) Singh S, Chauhan VK, Smith EHB. A self controlled RDP approach for feature extraction in online handwriting recognition using deep learning. *Applied Intelligence*. 2020;50(7):2093–2104. Available from: https://doi.org/10.1007/s10489-020-01632-4.

20) Ghosh R, Vamshi C, Kumar P. RNN based online handwritten word recognition in Devanagari and Bengali scripts using horizontal zoning. *Pattern Recognition*. 2019;92:203–218. Available from: https://doi.org/10.1016/j.patcog.2019.03.030.

21) . . Available from: https://www.kaggle.com/general/221455.