

Recognition of Off-line Kannada Handwritten Characters by Deep Learning using Capsule Network

Ramesh. G, J. Manoj Balaji, Ganesh. N. Sharma, Champa H.N

Abstract—Handwritten character recognition is an important subfield of Computer Vision which has the potential to bridge the gap between humans and machines. Machine learning and Deep learning approaches to the problem have yielded acceptable results throughout, yet there is still room for improvement. off-line Kannada handwritten character recognition is another problem statement in which many authors have shown interest, but the obtained results being acceptable. The initial efforts have used Gabor wavelets and moments functions for the characters. With the introduction of Machine Learning, SVMs and feature vectors have been tried to obtain acceptable accuracies. Deep Belief Networks, ANNs have also been used claiming a considerable increase in results. Further advanced techniques such as CNN have been reported to be used to recognize Kannada numerals only. In this work, we budge towards solving the problem statement with Capsule Networks which is now the state of the art technology in the field of Computer Vision. We also carefully consider the drawbacks of CNN and its impact on the problem statement, which are solved with the usage of Capsule Networks. Excellent results have been obtained in terms of accuracies. We take a step further to evaluate the technique in terms of specificity, precision and f1-score. The approach has performed extremely well in terms of these measures also.

Keywords-Capsule Network, Character recognition, Computer Vision, Deep Learning, Kannada Characters

I. INTRODUCTION

Computer Vision is a fast-growing field of computer science that is making its way into all other domains. An important challenge in the field of Computer vision is to develop efficient and generic handwriting recognition technology in the interest of a wide range of application. Translation of scripts, reading sign boards, banks, offices, legislative bodies, literature, archaeological applications, assistance to blind, are some the important areas of application of handwriting recognition. Digitization of huge amount of documents can be easily carried out with the help of handwriting recognition technology. Various advancements and efforts have been made in the field ranging from scene text detection and recognition [32], writer identification [16] that helps in specifically recognize handwritten scripts in international languages.

Some of the elegant techniques such as, various feature extraction methods have been used to recognize English scripts, which have yielded impressive results as presented by Mori *et al.*, [17]. Work towards the classification of handwritten characters of various languages has also been done using neural networks. The scope of these techniques has been limited to international languages such as Latin, Chinese, Japanese, Arabic as quoted by Amin *et al.*, [4] and Nagy G [19]. Even though considerable efforts have also been made to recognize Indian language scripts, most of them emphasise towards Gurumukhi, Devanagari or Bangla languages [1], [12], [13] [20] [30].

Very fewer efforts are seen with the introduction of 'state-of-the-art' technology in the field of handwriting recognition to regional languages. Majority of the above works use Support Vector Machines for classification of handwritten characters. Some of the latest works towards recognizing South Indian language characters include usage of feature vectors with deep belief networks or machine learning for classification. Further, the technological improvements in the field of Computer Vision have scaled up. The artificial neural networks which are commonly used with various feature extraction techniques have been succeeded by Convolutional neural networks or CNNs. The Convolutional neural networks consist of multiple neuron layers, which carry forward the strong features as scalars. This feature of CNN is backed by max-pooling to increase the area of interest for any particular feature but, the orientation of the features is lost in the process due to the scalar representation of features. Capsule networks have been introduced to rectify the above problem, by considering the features as vectors, with information about both the probability of the feature and a set of instantiation parameters such as pose, which determine its orientation. Using dynamic-routing it is ensured that the output of a layer is directed to the appropriate parent layer. This paper aims in classifying off-line kannada handwritten letter with the help of effective classifier techniques. The work is unfolded over the pages in 8 sections. Section I provides the introduction and Section II provides the insight about the previous works on the problem statement. Section III is about the motivation behind this work and Section IV describes clearly the problem statement we have approached to solve.

Revised Manuscript Received on August 28, 2019.

Ramesh. G, Department of Computer Science and Engineering University Visvesvaraya College of Engineering Bengaluru - 560001, India

J. Manoj Balaji, Department of Computer Science and Engineering University Visvesvaraya College of Engineering Bengaluru - 560001, India

Ganesh. N. Sharma, Department of Computer Science and Engineering University Visvesvaraya College of Engineering Bengaluru - 560001, India

Champa H.N, Department of Computer Science and Engineering University Visvesvaraya College of Engineering Bengaluru - 560001, India

Section V narrates the system architecture and Section VI provides a detailed description of the proposed system. Section VII is the results section and Section VIII concludes the work, while Section IX addresses the acknowledgement.

II. RELATED WORK

Handwriting recognition is a part of computer vision and revolves around identification and classification of handwritten character or texts of various different languages. The source of input is from various documents, photographs and other surface devices which enable characters to be written on them. Effort towards advances in captured scene text detection and recognition [32] have been made by Zhu *et al.*, to identify state-of-the-art algorithms and predicted possible research directions.

Major challenges in the process of recognition of characters of Indian languages range from the variety in the set of characters to the similarity between the characters. The solutions proposed are custom designed and no definite generic solution can be provided. Also, the origin of Indian language script being the Brahmi script, the similarity between the characters of two different languages is a challenge open to be solved. Further in Kannada language script, apart from vowels and consonants, the compound characters are represented as a combination of one or more consonant or vowel, with a symbolic representation of the vowel/consonant like a sub/superscript to the character. These problems are confined to the classification of Indian languages and scripts, unlike English. Also, the different styles of writing and varying sizes cause the results to be skewed in some methods of classification. Pertaining to the handwritten character recognition of Kannada Kagunita characters, Kagunita characters' moments feature extraction from Gabor wavelets has been emphasised by Ragha *et al.*, [24]. Work towards the provision of benchmark Kannada handwritten document dataset and its segmentation is done by Alaei *et al.*, [2] [3] to solve the unavailability of a standard data set and provide an insight by ground truths for researchers working on the Kannada language.

Obaidullah *et al.*, [23] have also contributed to the dataset of Kannada handwritten scripts recently, with a page-level document image dataset of 11 Indic scripts, of which, Kannada is also part of. The complexity and versatility of Kannada script, it being a Dravidian language is quoted by Ramakrishna *et al.*, [25] in their attempt to solve the challenges in the segmentation of on-line handwritten, isolated Kannada words. An attempt to classify totally unconstrained handwritten Kannada characters using Fourier transform based Principle Component Analysis and Probabilistic Neural Networks is made obtaining an accuracy of 88.64% by Manjunath *et al.*, [14] while Niranjana *et al.*, [22] had approached the same problem of classification of unconstrained handwritten Kannada character using FLD. A newer approach to the same problem has been the usage of ridgelet transforms, by representing monodimensional

singularities in bidimensional space, is used by Naveena

et al., [21]. Contributions towards the sub-field of on-line handwritten Kannada character recognition is also seen. A method to choose classifiers for Hierarchical Recognition of Kannada characters in online mode is proposed by Murthy *et al.*, [18]. Further, combination of online and off-line complementary systems is also introduced to recognize handwritten Kannada characters yielding an increase of online recognizer accuracy by 11%. Application of Kannada handwriting recognition techniques has been used to identify writer's text. Dhandra *et al.*, [6] have worked towards writer identification by using texture analysis and a solution towards this problem rectification is done using a feature vector consisting of directional multi-resolution spatial features based on Radon transforms, Discrete Cosine transforms and structural features such as aspect ratio and on-pixel ratios.

Some of the recent advancements in the field have concentrated towards better results by usage of various techniques such as SVMs and Neural Networks. Ramya *et al.*, [26] has used SVM to classify the images of online Kannada handwriting based on various features such as x,y coordinates, pressure and strokes and has achieved an average of accuracy 94.35%. Other techniques to dynamically recognize online handwriting for multiple languages was made by Keyzers *et al.*, [11] using techniques of generation and scoring of character hypothesis and best path searching. The authors also suggest language-specific adaptations to international languages such as Latin, Chinese, Japanese and Indian languages such as Gurumukhi, Kannada, Telugu, Malayalam and others. Word retrieval using Gabor Wavelets from Kannada documents has been worked upon by Hangarge *et al.*, [8] using directional energy feature of the word images. The preprocessing method of the data set, as an important factor in the process of classifying the Kannada characters, is studied by using signal processing and statistical techniques by Ramya *et al.*, [27]. This clearly shows how the performance of the classifiers such as SVM are affected by the preprocessing and testing methods. In off-line Kannada handwritten character recognition, Hallur *et al.*, [7] have proposed a holistic approach based system to recognize the off-line handwritten Kannada numerals. This system is tested on a database of 147 x 10 Kannada digits contributed by 147 writers. In spite of improper shape in handwritten numerals our proposed system archives accuracy of 95.98% in efficiently recognising them.

In [5], the authors have made efforts towards the restoration of degraded Kannada handwritten paper manuscripts or hastapratish using both special local and global binarization techniques, by elimination of uneven background illumination.

MSE and PNSR techniques have been used to measure performance, with the benchmark results obtained by Epigraphists. Comparison of these techniques with standard techniques, such as sauvola and niblack, which demonstrate the efficacy of the proposed method is also

characters. Karthik *et al.*, [10] present a method based on Histogram of Oriented Gradients for the recognition of handwritten Kannada numeral. HOG descriptors, which are proven to be invariant to geometric transformation, are one among the best feature descriptors for character recognition. They have used multi-class SVMs for the classification. The method is experimented by the authors on 4,000 images of isolated handwritten Kannada numerals and an average accuracy of 95% is obtained. Further, the use of HOG with ANN(Artificial Neural Networks) and SVM is done by Yadav *et al.*, [31] in their work towards the field. They describe an approach to performing classification on Kannada characters of handwritten origin or those present in natural images, using the HOG descriptors, for feature extraction from the images of the handwritten kannada characters, and employing a machine learning model (neural networks or support vector machines) for final classification. This effort shows the comparison of classification accuracies between the two classifiers.

Mirabdollah *et al.*, [15] have proposed the alternative feature vector of DAG, wherein the image is divided into blocks and windows of standard sizes, and the average of the values of pixels in each window is appended to the descriptor to obtain the feature vector. The authors Karthik *et al.*, in [28] have grouped the vowels and consonants separately and used 400 images per character to train the deep belief network. They have claimed an average accuracy of 97.04%. Even though some of these efforts have yielded considerable accuracies, the introduction of state-of-the-art, robust methods to the field is a clear requirement, in the view of the challenges issues with these existing methods.

III. MOTIVATION

Kannada being one of the derivatives from the Bramhi language, is a refined and elegant language. The vocabulary of the language is vast and apt. Karnataka state in the country of India has identified itself with Kannada being its official language. Kannada is also the transactional language in Karnataka. Among the official languages of India, Kannada is also one of the group, due to its widespread usage. The advancements in various fields have come from the western world and local languages have become barriers rather than assets, in the path of knowledge for the localities. Recognition, followed by translation of Kannada characters is the required technology to bridge this gap.

Apart from the philanthropy, the existing technological approaches to recognize handwritten Kannada characters are prone to limitations as elaborated in the previous sections. Also, some of the latest technologies such as Convolutional Neural networks also, are not the best-suited approaches to

done. Some of the gradient feature extraction techniques such as Histogram of Oriented Gradients, Distributed Average of Gradients for classification have been tried to classify handwritten Kannada

the problem due to reasons stated in the sections to follow. Thus the importance of the solution to this problem statement and the absence of a robust, reliable, apt and updated technique to solve serves as the motivation for us. This work is an attempt to take a step towards stability from the above mentioned, existing situations.

IV. PROBLEM STATEMENT

This work aims to solve the "recognition of handwritten Kannada characters", by using a robust, apt, state of the art technique, which provides best in the class accuracies and reliable results. Kannada alphabet has 13 vowels and 34 consonants. The effort is focused on the identification of these characters from various sources of literature. Hand in hand, the work also analyses the limitations in various techniques that are in use as the solution for the problem statement in context.

V. SYSTEM ARCHITECTURE

The work flow of the system is represented as shown in the figure:

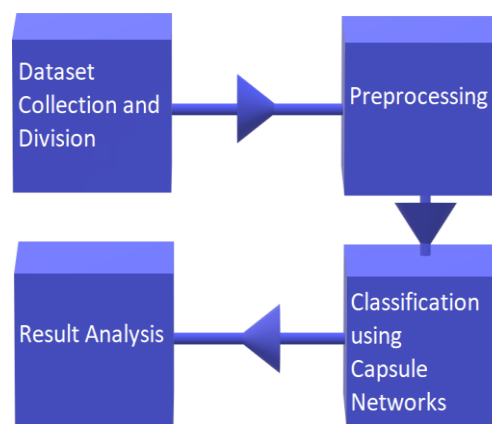


Fig. 1. System Architecture

The stages in the method used are as follows:

- 1) **Dataset Collection and Division:** The collected images set, containing images of each Kannada character are separated manually into vowels and consonants. The dataset is further divided in 80:20 ratios for training and validation respectively.

- 2) **Dataset Customisation:** The system involves minimal preprocessing, where in the images are converted to gray scale and resized to 28x28.
- 3) **Classification using Capsule Networks:** The customised images are classified using the Capsule Network model. The training and testing dataset thus

VI. PROPOSED SYSTEM

A. Kannada Language

The Kannada Alphabet or the "Varnamale", shown in Fig 2 consists of 47 characters with 13 vowels called "Swaras" and 34 consonants called "Vyanjanas". Kannada as a language has a very old history and is one of the wide-spread languages in the country. It is one of the 22 official Indian languages. Unlike the character in English, the Kannada characters have very similar orientation and minor differences are seen between some characters. The strokes of the Kannada characters also vary from English ones, as in the latter, most of the letters are written with strokes above and below a horizontal line, while in the former most the letters are written with a vertical line of reference. Considering the origin of most of the Indian scripts from the Brahmi script, scripts for Telugu and Kannada languages are similar in many ways. The language consists of the formation of complex characters from the combination of ones in the Varnamale. The language displays some minor changes from its early form which is called "Halegannada" to the current version in use across the world.

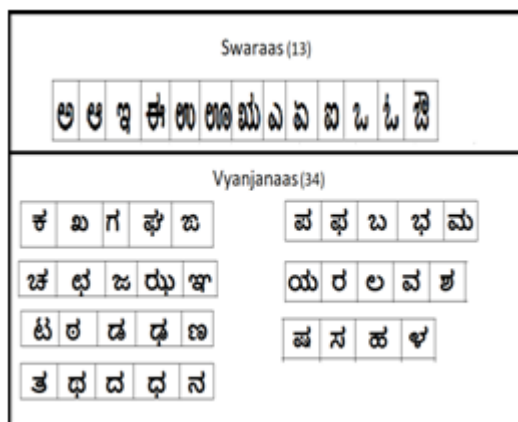


Fig. 2. Alphabet of Kannada language script(Varnamale)

B. Capsule Networks

1) **Capsules:** Capsule Networks imitate the human visual system by considering a parse tree structure construction. Each view processed by the visual cortex is converted into a parse tree structure and the nodes of the tree represent the neurons. A group of neurons on a particular layer of the multi-layered structure is termed as a capsule. The activation of a capsule is dependent on the constituent neurons and produces output vector with both the probability of an entity being present in the image and a set of instantiation parameters which determine the pose, lighting, deformations and other visual aspects which are dependent on the canonical form of the entity in the image. The probability factor of the vector remains translational invariant, but the

obtained in the previous steps are used to evaluate the model.

- 4) **Result Analysis:** The output of the classifier is analysed and the accuracy, sensitivity, f1 scores and precision are determined, thus giving a detailed analysis over the model performance.

other parameters change with a change in the visual notation of the entity in the image. This method is inspired by the 3D rendering techniques which form the visual entity using information

about it. The reverse of the process is hypothesised to be the mechanism of visual information storage in the human brain. Hence, capsules also function as "Transforming auto encoders" [9]

The authors ensure the validity of the probability factor to remain in the range of 0 and 1 with a non-linearity function which does not affect the orientation, but the magnitude is scaled down. Also, the ensurance of the output of one capsule reaching proper parent layer capsule is done by the dynamic routing algorithm. The last layer of the network is convolutional and the spatial information is place-coded for low-level capsules and rate-coded for higher level capsules in the hierarchy. This is with the background of an increase in the degrees of freedom with the increase in the level of the capsules in the hierarchy [29], The Structure of capsule network as shown in Fig.3.

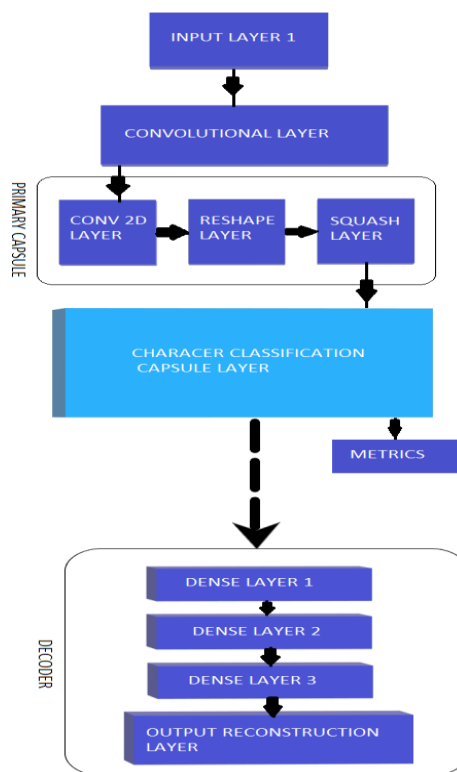


Fig. 3. Architecture diagram of the Capsule Networks

2) **Functioning of Capsule Networks:** The traditional neural network approach to the down-scaling of the output using the non-linearity function started with the step function, to sigmoid and CNN using ReLu for the activation of neurons in the next layers. These approaches handle the validity of the output of the neurons, as they are considered to be the probabilities of the presence of the visual entity. Thus the output is scaled down to a value between 0 and 1. The methods work when the output of the neurons is a scalar. Capsule Networks are composed of units whose output is a vector with both the probability and the orientational details. Thus the non-linearity function used is called "squash" which reduces the weighted sum of the prediction vectors to a probabilistic value.

$$v_x = \frac{\|s_x\|^2}{1 + \|s_x\|^2} \frac{s_x}{\|s_x\|}$$

the view of complex characters("ottaksharas") in Kannada alphabet, which consists of representations of two or more consonants, while the vowels, in combination with every consonant give a wide array of characters which make the "gunitaaksharas". The quality of the images in the data set varied over colours and pressure from writers as shown in Fig 4.

where v_x is the output vector from capsule x and s_x is the sum of inputs to the particular capsule.

The sum of inputs to any capsule x is dependent on the prediction vectors and the coupling coefficient values, in turn, the prediction vector is calculated as the product of the weight matrix and the output vector of a capsule in the previous layer.

$$s_x = \sum_i a_{ix} \hat{U}_{i|x}$$

$$\hat{U}_{i|x} = W_{ix} u_i$$

The coupling coefficient is determined by a softmax function variant for the biases as follows.

$$a_{ix} = \frac{e^{b_{ix}}}{\sum_y e^{b_{iy}}}$$

The routing algorithm is responsible for the calculation of the vector outputs from the capsules and the flow of this output. The training of the network is posed as a problem of minimization of loss function which is formulated as:

$$L_z = T_z \max(0, m^+ - \|v_z\|)^2 + \lambda (1 - T_z) \max(0, \|v_z\| - m^-)^2$$

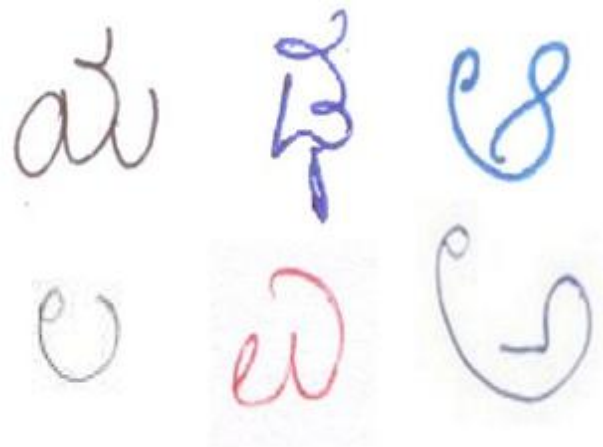


Fig. 4. Data Set sample images

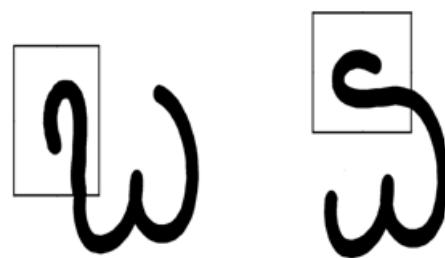


Fig. 5. Characters "Ba" and "Yee" differ only by the position of the curve element

VII. RESULT

A. Simulation Data Set

The data set consists of 47 users data where each user's 500 off-line Kannada handwritten images are collected. To capture them a camera of 16 pixels was used and each image of 230x230px was collected. Thus a total of 23500 images were collected, out of which 18800 images have been used for training and the remaining 4700 for testing the network. The collected dataset is divided into vowels and consonants, to experiment separately. This separation has been opted in

B. Model Description

The approach used in the work is to separately tackle the recognition of Kannada Vowels(Swaras) and the Consonants(Vyanjanas). The input layer 1 feeds the images of size 28 X 28 to the convolutional layer 1. The convolutional layer which traditionally is used with a kernel size of 9 is split into two with kernel sizes of 3 and 7 yielding an increase in validation accuracy of 4 percent for Consonants and 7 percent for vowels. The images go through a series of three convolutional layers, of which one is the primary capsule layer.

TABLE I MODEL DESCRIPTION FOR VOWELS

Layer Type	Output Shape	Parameter Count	Next Layer
Input Layer 1	(None, 28, 28, 1)	0	Convolutional Layer 1
Convolutional Layer 1	(None, 26, 26, 256)	2560	Convolutional Layer 2
Convolutional Layer 2	(None, 20, 20, 256)	3211520	Primary Capsule Convolutional Layer
Primary Capsule Convolutional Layer	(None, 6, 6, 256)	5308672	Primary Capsule Reshape Layer
Primary Capsule Reshape Layer	I (None, 1152, 8)	0	Primary Capsule Squash Layer
Primary Capsule Squash Layer	(None, 1152, 8)	0	Character Capsule Layer
Character Capsule Layer	(None, 13, 16)	1916928	Mask Layer 1, Capsule Network
Input Layer 2	(None, 13)	0	Mask Layer 1
Mask Layer 1	(None, 208)	0	Decoder
CapsNet	(None, 13)	0	Metrics, Input Layer 2
Decoder	(None, 28, 28, 1)	1435920	Mask Layer 1
Total params: 11,875,600			
Trainable params: 11,875,600			
Non-trainable params: 0			

DECODER DESCRIPTION FOR VOWELS

TABLE II

Layer Type	Output Shape	Parameter Count
Dense Layer 1	(None, 512)	107008
Dense Layer 2	(None, 1024)	525312
Dense Layer 3	(None, 784)	803600
Output Reconstruction Layer	(None, 28, 28, 1)	0
Total params: 1,435,920		
Trainable params: 1,435,920		
Non-trainable params: 0		

These layers perform kernel convolutions over the image with different kernel sizes of 3, 7 and 9 respectively. The objective of the convolutional layers is to detect the basic features and their combinations, in the 2D image provided as input. The output of the primary capsule convolutional layer is reshaped and squashed to feed forward it to the character capsule layer, which performs the expansion of the input scalar from the previous layer into a vector. The models used for the two sub problems of classification of vowels and consonants differ in the output shapes of the character capsule layer. This layer is where the routing algorithm works. The output shape of the character capsule layer is defined by the number of classes of input and the dimensionality of the output vector produced by the capsules. The input layer 2 accepts the input from character capsule layer and feeds it to the decoder network. The mask layer 1 which is a part of decoder network and maximises the capsule output with highest vector length and feeds this input to the decoder model. The dense layers of the decoder network are fully connected neurons and every output from the layer preceding is weighted and directed to the next neuron. The layers of the decoder network collectively decode the input vector and perform the classification. The layers of the model handle the sections of the images, filtered, as trainable parameters, and the count of the parameters has been mentioned under the respective models in tables I, II, III, IV and V is Comparison of the Proposed work with the Existing works.

C. Performance Analysis

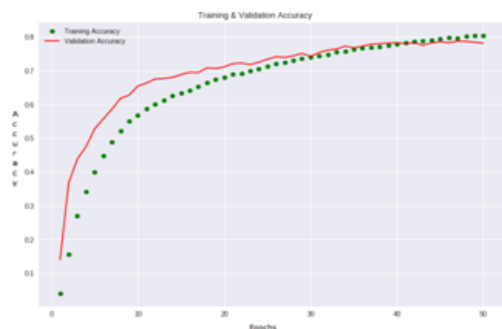


Fig. 6. Training vs Validation accuracy in Convolutional Neural Networks for Consonants

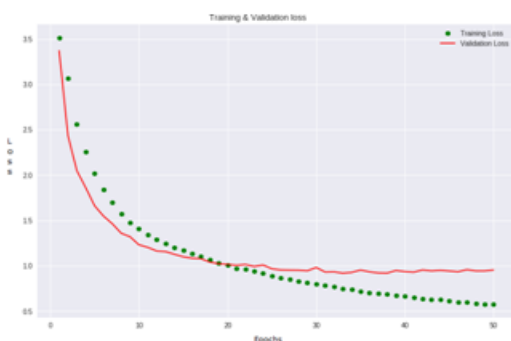


Fig. 7. Training vs Validation loss in Convolutional Neural Networks for Consonants

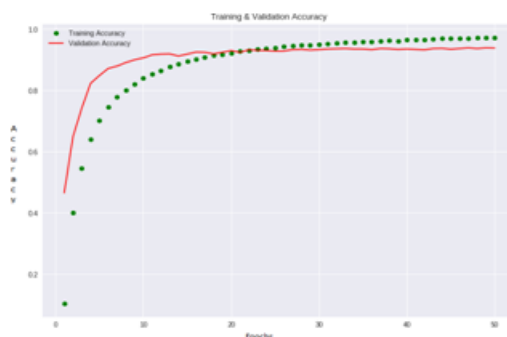


Fig. 8. Training vs Validation accuracy in Convolutional Neural Networks for Vowels

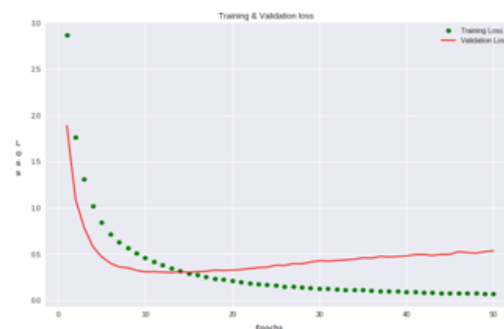


Fig. 9. Training vs Validation loss in Convolutional Neural Networks for Vowels

1) Convolutional Neural Network Model:

Convolutional Neural Networks Are Translation Invariant and use the scalar approach to propagate features of the image, which leads to loss of orientation information of elements. The arcs and lines which form the Kannada characters are treated to be the strong features, but the positional attributes such are disregarded. Examining the characters in Kannada alphabet, one can notice that, not only the probabilistic presence of the elements such as circles, arcs, lines that is important but also the position at which these occur. Fig 9 shows two different letters, the position of the highlighted curve element is the only difference. Considering the curve element to be a strong feature and if the letter is broken by pressure difference by the writer or any other common aspect of noise, the CNN regardless of its position will classify both the characters to be the same.

Capsule Network Model: Acknowledging the fact that, instantiation parameters of the elements of the images are important to achieve proper classification, capsule networks fit best, for the requirement, supported by the architecture and ability to take the information about the pose of the element into consideration. The results obtained were as follows: Figures 6-13 show the variation of the accuracy and loss percentage over 50 epochs of training and validation. The technique presents state - of - the - art performance with average training accuracies of 98.56 percent for consonants and 97.05 percent for vowels and validation/test accuracies of

TABLE III
MODEL DESCRIPTION FOR CONSONANTS

Layer Type	Output Shape	Parameter Count	Next Layer
Input Layer 1	(None, 28, 28, 1)	0	Convolutional Layer 1
Convolutional Layer 1	(None, 26, 26, 256)	2560	Convolutional Layer 2
Convolutional Layer 2	(None, 20, 20, 256)	3211520	Primary Capsule Convolutional Layer
Primary Capsule Convolutional Layer	(None, 6, 6, 256)	5308672	Primary Capsule Re-shape Layer
Primary Capsule Reshape Layer	I (None, 1152, 8)	0	Primary Capsule Squash Layer
Primary Capsule Squash Layer	(None, 1152, 8)	0	Character Capsule Layer
Character Capsule Layer	(None, 34, 16)	1916928	Mask Layer 1, Capsule Network
Input Layer 2	(None, 34)	0	Mask Layer 1
Mask Layer 1	(None, 544)	0	Decoder
CapsNet	(None, 34)	0	Metrics, Input Layer 2
Decoder	(None, 28, 28, 1)	1607952	Mask Layer 1
Total params: 15,144,208			
Trainable params: 15,144,208			
Non-trainable params: 0			

TABLE IV
DECODER DESCRIPTION FOR CONSONANTS

Layer Type	Output Shape	Parameter Count
Dense Layer 1	(None, 512)	279040
Dense Layer 2	(None, 1024)	525312
Dense Layer 3	(None, 784)	803600
Output Reconstruction Layer	(None, 28, 28, 1)	0
Total params: 1,607,952		
Trainable params: 1,607,952		
Non-trainable params: 0		

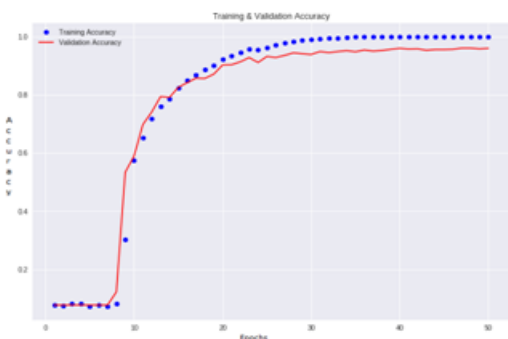


Fig. 10. Training vs Validation accuracy in Capsule Networks for Vowels

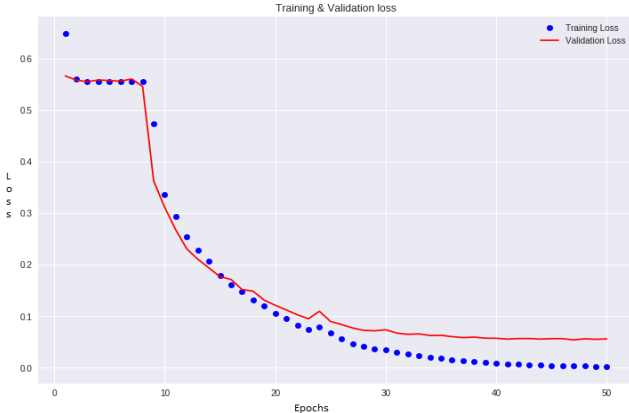


Fig. 11. Training vs Validation loss in Capsule Networks for Vowels

Fig. 11. Training vs Validation loss in Capsule Networks for Vowels

83.94 percent for consonants and 95.92 percent for vowels. The difference in accuracies are explained by the structural similarities between the consonants.

The measures of performance such as f1-score, recall and precision have also been calculated for the capsule network as shown in Fig 14 and 15, for the given data set. These measures give an insight about the model's performance with classifying the characters. The model's accuracy defines the number of correct predictions made by the model out of all the predictions. But this measure alone can sometimes not be sufficient to judge the performance of the model, as, in a hypothetical example, if the model predicts 98 of the 100 the characters of class "A" as not characters of class "B", the model's accuracy would still be calculated to be 98%, as accuracy is given by,

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN}$$

, where TP, TN, FP, FN denote the number of True Positives, True Negatives, False Positives, False Negatives respectively. The other measures of performances, thus used to evaluate the

model are:

$$Recall = \frac{TP}{TP + FN}$$

and

$$F1_Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

The measures of precision and recall of the model for multiple characters,

would be defined over the confusion matrix M with i rows and j columns, where ith row indicates the distribution of predictions for ith character over all other characters and jth column indicates the distribution of jth character being predicted, as:

$$Precision = \frac{\sum_i M_{ii}}{\sum_j M_{ji}}$$

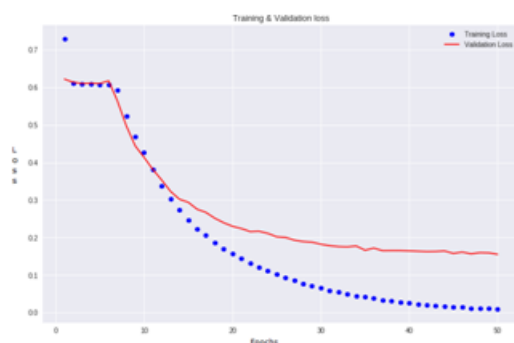


Fig. 12. Training vs Validation accuracy in Capsule Networks for Consonants

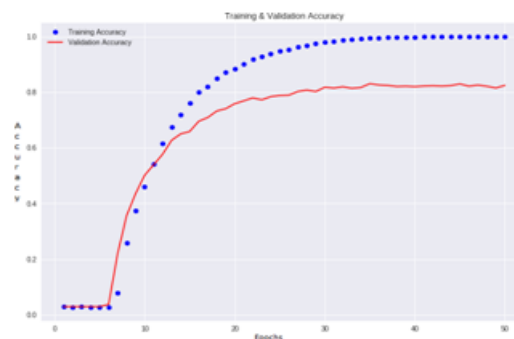


Fig. 13. Training vs Validation loss in Capsule Networks for Consonants

VIII. CONCLUSION

In this paper capsule network and CNN is used for classification of off-line Kannada handwritten letters. Capsule Networks outperform out of both these techniques and

Table V
Comparison of the Proposed Work with the Existing Work

Authors	Method	Accuracy Obtained
Manjunath <i>et al.</i> , [14] 2010	PCA+ANN	88.64%
Karthik <i>et al.</i> , [10] 2015	SVM+HOG	96.41%
Karthik <i>et al.</i> , [28] 2018	Deep Belief Networks	97.04%
Proposed	Capsule Networks	98.7%

given its advantages with the ability to handle raw images, saving preprocessing time, to preserve positional information of entities, which helps in the recognition of not only letters but words, owing to the complex letter formation style in the language.

ACKNOWLEDGEMENT

The authors of the paper would thank S. Karthik for making the provision of dataset required to train and test the models. The first second and third authors have made equal contribution to this work.

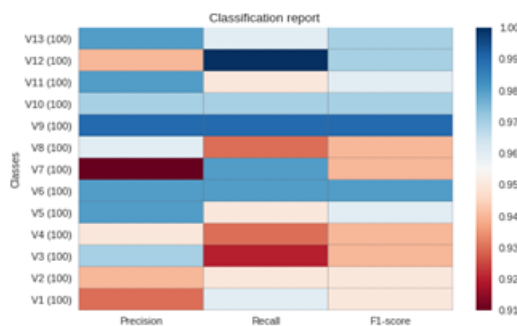


Fig. 14. Classification report for Vowels

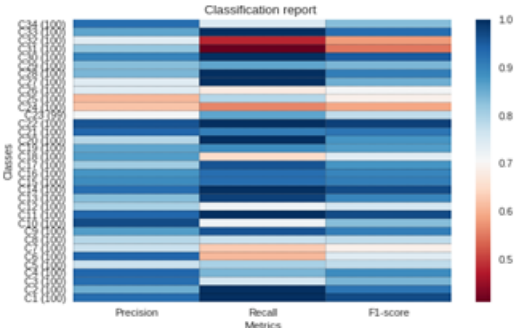


Fig. 15. Classification report for Consonants

the training accuracies for both peaked at 99.5, while the validation or testing accuracies were 83.94% and 95.92% respectively for consonants and vowels, while the accuracies for CNN have been noted to be 81 and 93%. With these impressive results, the real-life requirements that motivated us to work on the problem statement, are reachable with more reliability. The presented method's performance measures such as recall, precision and f1-scores have also

been up to the mark, with 96% for vowels and 84% for consonants respectively. Some of the improvements required lie in the computational times and performance with the classification of consonants. These improvements can make the system even more robust,

REFERENCES

1. A. Majumdar and B. Chaudhuri, "A mlp classifier for both printed and handwritten bangla numeral recognition," pp. 796–804, 2006.
2. A. Alaei, P. Nagabhushan, and U. Pal, "A benchmark kannada handwritten document dataset and its segmentation," pp. 141–145, 2011.
3. A. Alaei, U. Pal, and P. Nagabhushan, "Dataset and ground truth for handwritten text in four different scripts," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 04, p. 1253001, 2012.
4. A. Amin, "Off-line arabic character recognition: the state of the art," *Pattern recognition*, vol. 31, no. 5, pp. 517–530, 1998.
5. P. Bannigidad and C. Gudada, "Restoration of degraded kannada handwritten paper inscriptions (hastaprat) using image enhancement techniques," pp. 1–6, 2017.
6. B. Dhandra, M. Vijayalaxmi, G. MukarambiHu, and M. HanGarge, "Writer identification by texture analysis based on kannada handwriting," *Int. J. Comm. Netw. Secur.*, vol. 1, no. 4, pp. 80–85, 2012.
7. V. C. Hallur and R. Hegadi, "Offline kannada handwritten numeral recognition: Holistic approach," pp. 632–637, 2014.
8. M. Hangarge, C. Veershetty, R. Pardeshi, and B. Dhandra, "Gabor wavelets based word retrieval from kannada documents," *Procedia Computer Science*, vol. 79, pp. 441–448, 2016.
9. G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," pp. 44–51, 2011.
10. S. Karthik and K. S. Murthy, "Handwritten kannada numerals recognition using histogram of oriented gradient descriptors and support vector machines," pp. 51–57, 2015.
11. D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang, and V. Carbone, "Multi-language online handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1180–1194, 2017.
12. G. S. Lehal and C. Singh, "A post-processor for gurmukhi ocr," *Sadhana*, vol. 27, no. 1, pp. 99–111, 2002.
13. A. Majumdar and B. Chaudhuri, "Curvelet-based multi svm recognizer for offline handwritten bangla: a major indian script," vol. 1, pp. 491–495, 2007.
14. V. Manjunath Aradhya, S. Niranjana, and G. Hemantha Kumar, "Probabilistic neural network based approach for handwritten character recognition," *International Journal of Computer & Communication Technology*, vol. 1, no. 2, 3, 4, pp. 9–13, 2010.
15. M. H. Mirabdollah, M. A. Mohamed, and B. Mertsching, "Distributed averages of gradients (dag): A fast alternative for histogram of oriented gradients," pp. 97–108, 2016.
16. M. Mohammadi, M. E. Moghaddam, and S. Saadat, "A multi-language writer identification method based on image mining and genetic algorithm techniques," *Soft Computing*, pp. 1–15, 2018.
17. S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of ocr research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992.

19. V. N. Murthy and A. G. Ramakrishnan, "Choice of classifiers in hierarchical recognition of online handwritten kannada and tamil aksharas." *J. UCS*, vol. 17, no. 1, pp. 94–106, 2011.
20. G. Nagy, "Chinese character recognition: a twenty-five-year retrospective," pp. 163–167, 1988.
21. M. Naser, A. Mahmud, T. Arefin, G. Sarowar, and M. N. Ali, "Comparative analysis of radon and fan-beam based feature extraction techniques for bangla character recognition," *IJCSNS*, vol. 9, no. 9, p. 287, 2009.
22. Naveena and V. M. Aradhya, "An impact of ridgelet transform in handwritten recognition: A study on very large dataset of kannada script," pp. 618–621, 2011.
23. S. Niranjana, V. Kumar, and H. Kumar, "Fid based unconstrained handwritten kannada character recognition," vol. 3, pp. 7–10, 2008.
24. S. M. Obaidullah, C. Halder, K. Santosh, N. Das, and K. Roy, "Phdindic_11: page-level handwritten document image dataset of 11 official indic scripts for script identification," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 1643–1678, 2018.
25. L. R. Ragha and M. Sasikumar, "Feature analysis for handwritten kannada kagunita recognition," *International Journal of Computer Theory and Engineering*, vol. 3, no. 1, p. 94, 2011.
- A. Ramakrishnan and J. Shashidhar, "Development of ohwr system for kannada," *VishwaBharat@ idil*, vol. 39, p. 40, 2013.
26. Ramya, S and Shama, Kumara , "Comparison of svm kernel effect on online handwriting recognition: A case study with kannada script," pp. 75–82, 2018.
27. Ramya, S and Shama, Kumara, "The effect of pre-processing and testing methods on online kannada handwriting recognition: Studies using signal processing and statistical techniques," *Pertanika Journal of Science and Technology*, vol. 26, no. 2, pp. 671–690, 2018.
28. S. Karthik and K. S. Murthy , "Deep belief network based approach to recognize handwritten kannada characters using distributed average of gradients," *Cluster Computing*, pp. 1–9, 2018.
29. S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," pp. 3856–3866, 2017.
30. S. K. Shrivastava and S. S. Gharde, "Support vector machine for handwritten devanagari numeral recognition," *International journal of computer applications*, vol. 7, no. 11, pp. 9–14, 2010.
31. P. Yadav and M. Kumar, "Kannada character recognition in images using histogram of oriented gradients and machine learning," pp. 265– 277, 2018.
32. Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.