Multimedia Tools and Applications (2021) 80:20671–20686
https://doi.org/10.1007/s11042-021-10775-6

DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition

Sonika Rani Narang [1] · Munish Kumar [2] · M. K. Jindal [3]

Received: 28 September 2020 / Revised: 20 December 2020 / Accepted: 25 February 2021 /
Published online: 8 March 2021

Abstract

Devanagari script is the most widely used script in India and other Asian countries. There is a rich collection of ancient Devanagari manuscripts, which is a wealth of knowledge. To make these manuscripts available to people, efforts are being done to digitize these documents. Optical Character Recognition (OCR) plays an important role in recognizing these documents. Convolutional Neural Network (CNN) is a powerful model that is giving very promising results in the field of character recognition, pattern recognition etc. CNN has never been used for the recognition of the Devanagari ancient manuscripts. Our aim in the proposed work is to use the power of CNN for extracting the wealth of knowledge from Devanagari handwritten ancient manuscripts. In addition, we aim is to experiment with various design options like number of layes, stride size, number of filters, kenel size and different functions in various layers and to select the best of these. In this paper, the authors have proposed to use deep learning model as a feature extractor as well as a classifier for the recognition of 33 classes of basic characters of Devanagari ancient manuscripts. A dataset containing 5484 characters has been used for the experimental work. Various experiments show that the accuracy achieved using CNN as a feature extractor is better than other state-of-the-art techniques. The recognition accuracy of 93.73% has been achieved by using the model proposed in this paper for Devanagari ancient character recognition.

Keywords Devanagari handwritten character dataset · Devanagari ancient · Deep learning · Deep convolutional neural network · Optical character recognition

✉ Munish Kumar
munishcse@gmail.com

Sonika Rani Narang
sonikanarang@davcollegeabohar.com

M. K. Jindal
manishphd@rediffmail.com

Extended author information available on the last page of the article

1 Introduction

India is a land of timeless and rich heritage, to which our ancient monuments and documents are the testimony. Great knowledge of Indian sages and other great persons is preserved in ancient documents. Devanagari script is the most widely used script in India and other Asian countries. There is a rich collection of ancient Devanagari manuscripts which is a wealth of knowledge. These documents are rare and degraded documents. Degradation of ancient documents may be due to age or writing style, ink stains, lightening of ink, uneven space between text lines, overlapping of text lines or characters, different layouts, broken characters or torn pages. To make these manuscripts available to people, efforts are being done to digitize these documents. This has motivated us to propose a system for the recognition of Devanagari ancient documents. Other state-of-the-art techniques have been previously used for the recognition of handwritten Devanagari documents but not much work has been done for the recognition of Devanagari ancient manuscripts. Convolutional Neural Network (CNN) is a powerful model that is giving very promising results in the field of character recognition. Deep Neural Networks do not require any feature to be explicitly defined, instead, they work on the raw pixel data and generating the best features and using them to classify the inputs into different classes [Lee et al. [23]]. In this paper, the authors have considered the deep learning model for the recognition of ancient Devanagari manuscripts. Our task is challenging because there is no standard database is available for Devanagari ancient manuscripts. So, the authors have created their own dataset along with the recognition task. The authors propose to use CNN as a feature extractor as well as a classifier for the recognition of 33 classes of basic characters of Devanagari ancient manuscripts. Though basic characters in Devanagari script are 44, total classes identified are 33 as some characters are converted to the same classes. For example, after segmentation अ◌ा is converted to अ and modifier ◌ा. Also, some characters like ञ are not found in ancient documents samples so far. A dataset containing 5484 characters has been used for the experimental work. Various experiments show that the accuracy achieved using CNN as a feature extractor is better than other state-of-the-art techniques. Using CNN, the authors have reported a recognition accuracy of 93.73%. As per the best of the author's knowledge, very less work has been done for the recognition of Devanagari handwritten ancient manuscripts. Deep learning has never been used for the recognition of such manuscripts. So the proposed work is the first ever use of deep learning techniques for the recognition of characters from Devanagari handwritten ancient manuscripts.

This paper is organized as follows: Prior work is given in Section 2. The main objectives of this work are depicted in Section 3. Section 4 presented a brief overview of the proposed system. The Devanagari script is introduced in Section 5. Section 6 describes the proposed methodology. Section 7 gives the experimental setup. Results and discussions are given in Section 8. Finally, concluding notes and future directions are presented in Section 9.

2 Literature review

In literature, many feature extraction techniques have been proposed for image processing. Many survey papers for feature extraction are available in the literature [Sharma et al. [32]; Narang et al. [25]] etc. Bag and Harit [7] endeavoured to provide a broad survey of OCR work on Devanagari and Bangla scripts published in the year 2000 and afterwards. Various machine learning techniques are being used in every field including OCR. The importance of machine learning is depicted in the work presented by Alzubi et al. [5]. Theeramunkong and Wongtapan [40] have presented a method called multi-directional island-based projection to extract global features from handwritten characters. They proposed two statistical approaches, namely, the *n*-gram model and the hidden Markov model. Quiles and Romero [30] used the Multilayer Perceptron (MLP). MLP has been a milestone in many recognition problems. But, the performance of MLP depends upon the selection of good features [Yang et al. [44]]. Deep neural networks do not require any explicitly defined features. Deep neural networks generate the best features and then classify the inputs [Lee et al. [23]. CNN is one class of deep neural networks that can work with a comparatively smaller set of parameters. By changing the number of layers and the trainable parameters in each layer, the recognition ability of CNN can be varied [Krizhevsky [20]]. At present, CNNs have proven to be providing better results as compared to traditional Artificial Neural Networks (ANNs). Krizhevsky et al. [20] proposed impressive research based on CNN. They used the deep convolutional neural network to classify the ImageNet dataset. Recent researches on CNN have been focused on computer vision problems such as image segmentation, image captioning and image classification [Farabet [12]; Vinyals [43]; Ciresan [9]]. In CNN, dropout is a simple way to prevent over fitting [Srivastava et al. [36]. CNN is being widely used in handwritten character recognition problems. Purkaystha et al. [29] presented a CNN based OCR for Bengali handwritten character recognition. Uysel and Gunal [41] examined the impact of pre-processing on text classification in two different languages, namely, Turkish and English.

Singh and Lehal [34] presented a comparative performance analysis of feature(s)-classifier combinations for Devanagari OCR. neighbours. Singh et al. [35] presented a system for the recognition of handwritten Devanagari characters. They used gradient features with radial basis function neural network (RBF) neural network. Shelke and Apte [33] presented a new approach for the classification of unconstrained handwritten Devanagari characters. They used two stages for classification, the first stage is based on the fuzzy inference system and the second stage is based on structural parameters.. Acharya et al. [1] published a new image dataset for the Devanagari script, namely, Devanagari Handwritten Character Dataset (DHCD). The dataset consists of 92,000 images of 46 unique classes of modern Devanagari characters segmented from handwritten files. Also, they proposed a deep learning architecture for the recognition of these characters. The proposed system scored an accuracy of 98.47% on DHCD dataset. Khanduja et al. [18] proposed a hybrid mechanism that combined the structural features of Devanagari character images and a mathematical model of curve fitting to get the better features. Kumar [21] used a multi-layer perceptron (MLP) network for handwritten Devanagari OCR.

Ghosh et al. [14] have proposed a novel approach for detecting OCR errors and improving retrieval performance from the erroneous corpus. They achieved significant improvements over the state-of-the-art baselines on most of the dataset. Alizadehashraf and Roohi [4] developed a system for Persian handwritten character recognition using a convolutional neural network. Agarwal et al. [2] have presented a deep network model for paraphrase detection in a short text message. They developed a new architecture, called DeepParaphrase, which enables to create an informative semantic representation of each sentence. Jangid and Srivastava [16] presented a handwritten Devanagari character recognition system using layer wise training of deep convolutional neural networks and adaptive gradient methods. With the introduction of unsupervised pre-training phase, CNNs have become very effective. The role of unsupervised training is investigated by Erhan et al. [11]. Kumar et al. [22] have presented a survey for character and numeral recognition of various non-Indic and Indic scripts. Verma and Singh [42] presented a Hindi handwritten character recognition system using CNN. Heyong and Ming [15] proposed a method termed "Hebb Rule-based Feature Selection (HRFS)". They indicated that HRFS is effective to achieve promising results rather that other state-of-the-art work.

During the past decade, work is going on for the recognition of ancient manuscripts in various countries. Shah and Badgujar [31] gave a review on Devanagari Handwritten Character Recognition (DHCR) for ancient documents. Kim et al. [19] presented a dedicated OCR system for Hanja historical documents. Korean name given to Chinese characters is Hanza. Sousa et al. [38] proposed a system based on Gabor filters and fuzzy logic for ancient printed text document recognition. This work gave a success rate of 88%. Diem and Sablatnig [10] presented a work to recognize degraded characters using local features. This work was done on an ancient manuscript where characters were washed out (partially visible) due to aging. Support Vector Machine was used to classify local descriptors and then identified by a voting scheme of neighbouring local descriptors. Phan et al. [28] presented a Nom historical document recognition system. In this work, recursive X–Y cut and Voronoi diagrams for segmentation was used by the authors. They extracted gradient features. Further, for more fine classification, the modified quadratic discriminant function (MQDF) was used. Garz et al. [13] proposed a robust method to separate text area from the decorative area in ancient documents using Scale Invariant Feature Transform (SIFT) and local descriptors. Ceccotti and Belaid [8] have presented a hybrid approach which was accompanied by some dedicated ICR for ancient documents. A model was further proposed which combined several OCRs, and some specific intelligent character recognition based on convolutional neural network.

Sumetphong and Tangwongsan [39] proposed a novel solution to recognize broken characters in Thai historical documents based on set-partitions. Kavitha et al. [17] proposed two approaches for historical document classification. The first approach is based on skewness for Indus document classification from English and South Indian scripts. Nearest Neighbour based Approach (NNA) was then applied to classify English from South Indian scripts. Soumya and Kumar [37] proposed the acknowledgment of text in the old Kannada script of the Ashoka and Hoysala era. Segmentation of characters was performed using the Nearest Neighbour clustering algorithm. After the segmentation process, statistical features such as Homogeneousness, Standard Deviation, Mean, Correlation, Variance, Kurtosis, Skewness, Contrast, Energy, and Coarseness were extracted. Mamdani Fuzzy Classifier was used in the classification of characters. In the final phase, the classified characters of early times were exhibited in the new Kannada form. The recognition rate for Brahmi script was found to be appreciable in comparison to Hoysala script. Narang et al.[27] used discrete cosine transform(DCT) and Histogram of oriented gradients (HOG) features with Naïve Bayes, decision tree and SVM classifiers. They reported a maximum accuracy of 90.70% accuracy with DCT features and SVM classifier on the Devanagari ancient manuscripts. Avadesh and Goyal [6] proposed a CNN based OCR for the recognition of printed ancient Sanskrit manuscripts. They calculated pixel intensities to identify letters in the image. They considered typical compound characters (half letter combinations) as separate classes. They achieved recognition accuracy of 93.32%. Narang et al. [24] used various statistical feature extraction techniques namely, intersection points, open endpoints, centroid, horizontal peak extent and vertical peak extent features. For classification, they used Convolutional Neural Network, Neural Network, Multilayer Perceptron, RBF-SVM and random forest techniques. They obtained a maximum accuracy of 88.95% using majority voting scheme. Narang et al. [26] used scale-invariant feature transform (SIFT) and Gabor filter feature extraction techniques. Support vector machine (SVM) classifier was used for the classification task in this work. They achieved a recognition accuracy of 91.39% using the ten fold cross-validation technique and poly-SVM classifier. Ahlawat et al. [3] designed a CNN to classify MNIST handwritten digits. They achieved accuracy better than that of ensemble architectures, along with reduced operational complexity and cost.

After a thorough literature review, it was found that not much work has been done for the recognition of Devanagari ancient manuscripts. The major reason for this is found to be the non availability of ancient manusripts for experiments and degraded documents. For this work, the authors have created their own database. In future, this database will be made available online so as to facilitate research in this field. Deep learning has never been used for the recognition of these manuscripts. In this work, the authors have used CNN as a feature extractor as well as a classifier and it was found to be better than the existing feature extraction and classification techniques for the recognition of Devanagari ancient manuscripts.

3 Research objectives

The goal of our research is to recognize the ancient characters of the Devanagari script using a CNN model of deep learning. The specific research objectives of the study are as follows: -

- To propose a methodology for character recognition of different scripts.
- To achieve promising results for the recognition of ancient characters of Devanagari script.
- To compare the achieved recognition results with the state-of-the-art work.

4 Data collection and Preprocesing

The flow diagram of the Devanagari ancient manuscripts recognition system is presented in Fig. 1. This diagram includes data collection, data preprocessing and data recognition. This work starts with the collection of ancient Devanagari manuscripts from various

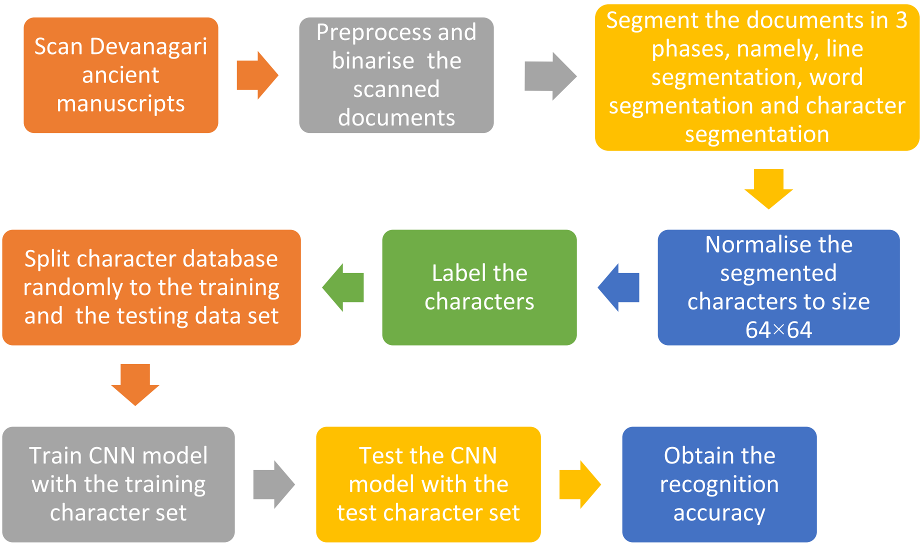

Fig. 1 Block diagram of the proposed system

libraries and museums in the country. The collected manuscripts are now processed in three phases – digitization, segmentation and recognition. The first step, namely, digitization involves the use of a camera, scanner etc. to convert manuscripts in the digital format generally (.jpg or .bmp). The author completed the task of digitization of more than 100 manuscripts, which were written between the 15th and 19th centuries and later converted those document images to binary scale format. The second step i.e. segmentation involves dividing the complete image into individual lines, individual words and then individual characters. All such types of characters were normalized to a size of 64 × 64 pixels character by using the nearest neighbourhood interpolation method for further processing and recognition.

The handwritten characters are of variable sizes, thickness and shapes in different documents due to different handwriting, nib sizes and writing styles of individuals. Therefore, the final step of character recognition becomes very tedious for handwritten ancient manuscripts. These characters are labelled. Dataset of normalized characters is randomly divided into two subsets: a training set and a testing set. The training dataset is used to train the CNN model and the test data is used to predict the correct labels and thereby obtaining recognition accuracy.

5 Characteristics of Devanagari script

Devanagari is a very popular script in India. Devanagari is used to write Hindi, Sanskrit, Nepali and Marathi languages. Devanagari has 11 vowels, 33 consonants, called basic characters. Vowels can be written as independent characters or by using diacritical marks. These diacritical marks are known as modifiers or *matras*. Characters formed by using modifiers are called conjuncts. Characters, formed by combining two or more consonants, are called compound characters. A sample of the basic Devanagari character set is delineated in Tables 1, 2. All the characters in the Devanagari script have a horizontal line at the upper part, known as *Shirorekha* or *headline*. The headline of one character joins with the headline of the next character of the same word.

In this work, authors have identified 33 basic characters that are found in Devanagari ancient manuscripts. Though there are 44 basic characters in Devanagari script, but for our work, a total of 33 classes are identified, because a few samples like अ◌ा gets converted to अ and matra in the segmentation process. Also, few samples like ञ are not found in ancient documents samples taken so far. These 33 classes are presented in Table 3. Table 3 shows 33 identified classes along with the corresponding characters from Table 1 and Table 2.

Table 1 Vowels and corresponding modifiers

Vowels	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
Corresponding Modifiers	---	◌ा	◌ि	◌ी	◌ु	◌ू	◌ृ	◌े	◌ै	◌ो	◌ौ

Table 2 Consonants

क	ख	ग	घ	ङ
च	छ	ज	झ	ञ
ट	ठ	ड	ढ	ण
त	थ	द	ध	न
प	फ	ब	भ	म
य	र	ल	व	
श	ष	स	ह	

6 Proposed methodology

Authors propose to use CNN as a feature extractor as well as a classifier for the recognition of 33 classes of basic characters of Devanagari ancient manuscripts. A dataset containing 5484 characters arranged in 33 classes has been used for the experimental work. Various experiments show that the accuracy achieved using CNN as a feature extractor is better than other state-of-the-art techniques.

Table 3 Basic characters identified in Devanagari ancient manuscripts and their corresponding characters from Table 1 and Table 2

(अ)	(इ)	(उ)	(ए)	(क)	(ख)	(ग)
(घ)	(च)	(छ)	(ज)	(ट)	(ड)	(त)
(थ)	(द)	(ध)	(न)	(प)	(फ)	(ब)
(भ)	(म)	(य)	(र)	(ल)	(व)	(श)
(ष)	(स)	(ह)	(ण)	(Half from of श)		

6.1 Convolutional neural network (CNN)

CNN has brought about a boom in computer vision and pattern recognition. It was introduced by Yann LeCun in 1998, but it gained popularity in 2012 when it was used in ImageNet competition. In this competition, CNN improved the accuracy by more than 15% as compared to the winner of 2011. Since then, CNN is used in many pattern recognition problems, especially in OCR. In many cases, CNN has shown better accuracy. CNN can be used as a feature extractor as well as a classifier. In the present work, the authors have considered the CNN for both tasks (features extraction as well as classification). Character images are passed directly to the CNN model for training and testing. CNN automatically extracts features and then classifies characters. CNN is a neural network that contains various layers like convolutional layer, pooling layer, activation layer, fully connected layer and a classification layer. These layers are briefly discussed in the following sub-sections.

6.1.1 Convolutional Layer

A convolution is an operation that converts the function into something else so that we may get more information. In this layer, mostly, a square kernel is used that is moved over the image. Convolutional layers multiply kernel value with the values in the image that is currently covered by the kernel and add these values to get a convoluted value. The distance moved by the window at a time is known as stride. In this work, we experimented with various kernel sizes and stride values and then finally selected a kernel of size 3×3 with stride 1.

We do not have to specify kernel values. The value of the kernel is learnt during training. Filters are used to learn some abstract concepts. Multiple features are used in the convolutional layer to learn multiple features. If one can use multiple convolutional layers on top of each other, then we may obtain information in each layer. In this work, we have used 3 convolutional layers. There are 32 filters in the first convolutional layer, 64 filters in the second convolutional layer and 128 filters in the third convolutional layer. Activation function ReLU (Rectified Linear Unit) is used after each convolution. In Fig. 2, we can see how different filters in each CNN layer interpret the character च(ch).

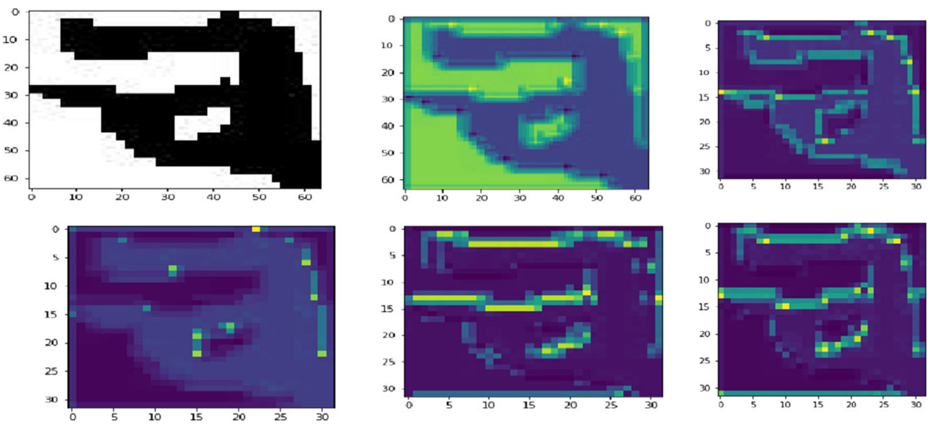

Fig. 2 Character च (ch) after various filters

6.1.2 Activation Layer

The next layer is the activation layer. In this layer, value is passed through a function that changes the value in a range. There are many activation functions like identity, binary step, arctan, rectified linear unit (ReLU) etc. For this work, we have used the ReLU activation function. This function is given as:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

6.1.3 Pooling Layer

Pooling layers describe a window of an image using a single value which is the max or the average of that window. The pooling layer is used mainly for two reasons:

- First reason is down sampling. It reduces the amount of computation to be done.
- The second reason is to send only the important data to the next layers.

The pooling layer also uses a kernel that moves over the image. It is like a convolutional layer; the only difference is the function applied on the kernel. Maxpooling and average pooling are the two mostly used for pooling functions. We have used maxpooling in this work with window size 2×2. Max pooling takes the largest value from the window of the image currently covered by the kernel. After each convolutional layer, there is a pooling layer in the present work.

6.1.4 Fully Connected Layer

We have a fully connected layer as the last layer. This layer uses the soft-max function. The purpose of this layer is classification. This layer computes the classification score of a character.

7 Experimental setup

The authors have considered the following phases for the experimental setup of the proposed work.

- The authors have used a dataset containing 5484 pre-segmented characters.
- Characters are divided into 33 classes.
- Normalized the size of each character into 64×64. So, our input to CNN is of size $[64 \times 64 \times 1]$. Here 1 indicates channels. In a color RGB image, we normally have 3 channels: red, green and blue. In a color CMYK image, we will have four channels (each colour being one channel). Filters will be applied to each channel. In this work, we have used a binarized image, so the channel is 1.
- The authors have used 3 convolutional layers having 32, 64 and 128 filters, respectively. Each convolutional layer used filter size 3×3 and stride 1. The first convolutional layer will produce an output of size $(64 \times 64 \times 32)$.

- To apply RELU function in the activation layer. It keeps the result unchanged.
- To use filters of size 2×2 in pooling layer. Maxpooling is used in the pooling layer for down sampling. The volume of data in the first pooling layer is down sampled to size [32, 32, 32].
- The fully connected layer calculates the class score of a character using the Soft-max function.

The CNN architecture used in this work is prescribed in Fig. 3.

8 Experimental results and discussions

For experimental work, authors have considered 5484 samples of pre-segmented characters taken from Devanagari ancient manuscripts composing a dataset of 33 classes. For the recognition of characters in ancient Devanagari manuscripts, CNN is used as a feature extractor as well as a classifier. The proposed CNN features are tested on an image database consisting of about 5484 images.

The effectiveness of the proposed method is evaluated by using CNN classifier. While implementing CNN, authors have experimented with different train-test sizes. Each of the train-test sizes are selected, and further tested for the different number of epochs. Experimental results with train-test sizes 80%–20% and 75%–25% are depicted in Table 4.

On train-test size of 80%–20%, maximum accuracy of 92.89% is achieved with 20 epochs. The train-test size was then changed to 75%–25% and an improvement in accuracy was observed, increasing it to 93.73%, the number of epochs being 30. Other train-test sizes were also tested, which depicted a decrease in accuracy. Figure 4 depicts the training and validation accuracy of the proposed model. Training accuracy shows how our model is progressing in terms of it's training whereas validation accuracy helps us to measure the quality of our model. It shows how well our model is able to make predictions on new data. Figure 4. shows that our model has achieved training accuracy of 100% after 30 epochs. For unseen data, validation accuracy is 93.73%.

CNN calculates the probabilities of a character belonging to a class and then classifies the character based on the highest probability. For example, a character ल (l) of Fig. 5, with 20 epochs, its probability of belongingness to a class is given in the following Table 5.

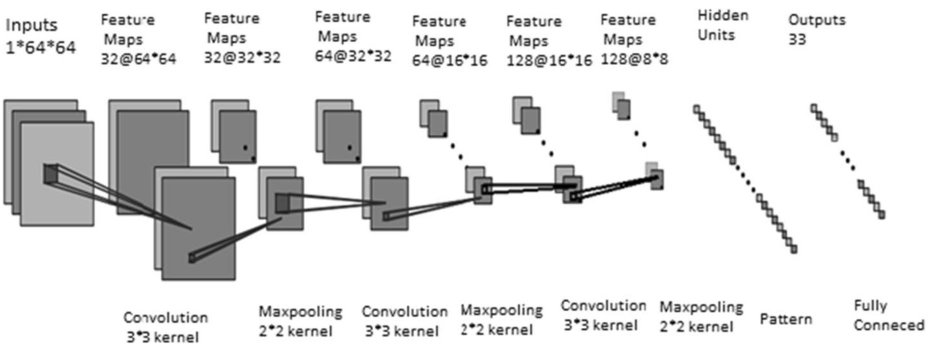

Fig. 3 CNN architecture for the present methodology

Table 4 Experimental results of the proposed system for ancient Devanagari character recognition.

Train/test split (in % age)	Number of epochs	Accuracy
80/20	20	92.89%
80/20	25	92.43%
80/20	30	92.80%
75/25	20	91.54%
75/25	25	93.65%
75/25	30	**93.73%**

8.1 Limitations of the CNN model for Devanagari ancient character recognition

Devanagari ancient characters have many types of characteristics and deformities which result in incorrect recognition. In this work, the authors have discussed some of these cases.

8.1.1 Mis-classified characters because of similarity in the shape of characters

Devanagari script has many characters that look similar. Sometimes, it is difficult to differentiate these characters even manually. A list of similar looking characters is given in Table 6.

Such characters are misclassified sometimes. For example, the following character in Fig. 6 is labelled as class प(p) but it is predicted as class य(y) because of similar structures. Some of these characters are hard to predict even manually because they are so similar. The probability that it belongs to class य(y) is 0.5888 and the probability that it belongs to class प(p) is 0.4109.

8.1.2 Mis-classified characters because of faded characters

There are some characters that are broken due to faded ink. Such characters may result in wrong identification. One such example of character is shown below in the Fig. 7. This character is labelled as class श(sh) but it is predicted as class ज(j). Some other misclassification occurs due to touching characters or different writing styles of writers.

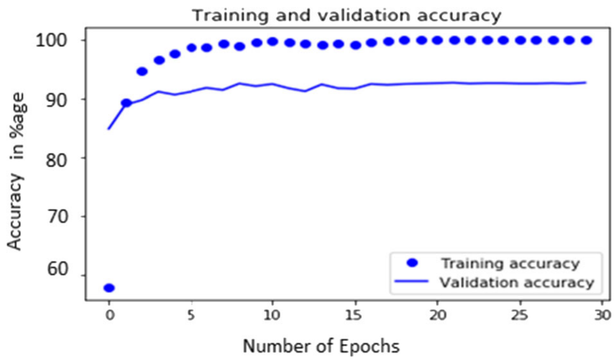

Fig. 4 Recognition accuracy for training and testing dataset

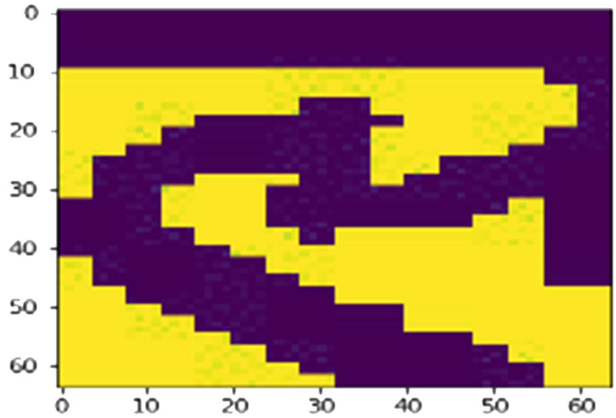

Fig. 5 Character identified as ल(l)

Table 5 Probability of belongingness of a character to a class

Class	Probability
ल	0.9832
ह	0.0128
व	0.0017
द	0.0015
क	0.0007
स	0.0001

8.2 Comparison with the existing work

To validate our proposed technique, we compared it with the work proposed by Narang et al. [27], Avadesh and Goyal[6] and Narang et al. [25] All three of the above mentioned works are done on Devanagari ancient manuscripts. Our proposed work shows considerable improvement over their work. The proposed work achieves 93.73% accuracy. Table 7 shows the comparative analysis of the proposed method with the method proposed by Narang et al. [27], Avadesh and Goyal [6] and Narang et al. [26].

Table 6 Similar shaped characters

न त	प य ष	ड इ ह
ट द ढ	भ म	य थ
क फ	घ ध	ब व

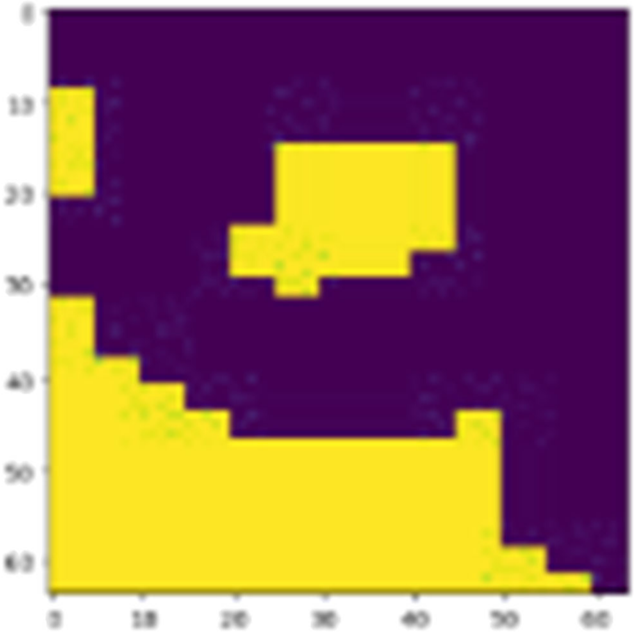

Fig. 6 Misclassified character due to similarity in the shape of charaters

For character recognition, the major implication is to understand that recognition results retrieved by the optical character recognition system are promising. In literature review, it was observed that Acharya et al.[1] achieved an accuracy of 98.47% on Devanagari handwritten modern characters by using deep learning techniques. But, in ancient documents, most of the

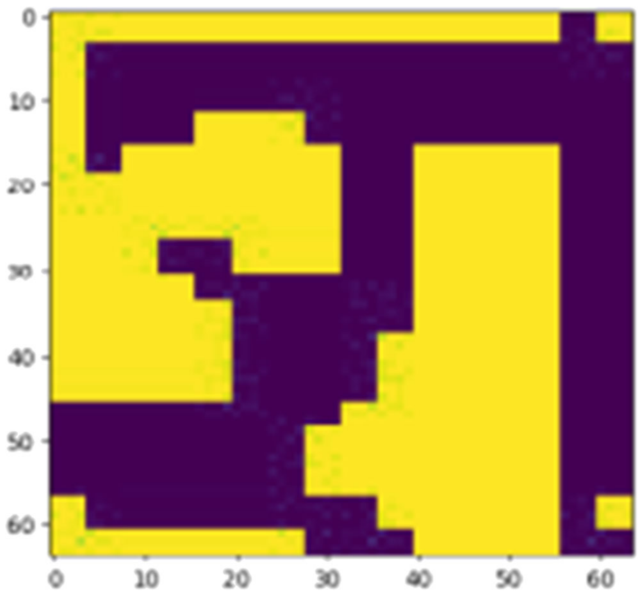

Fig. 7 Misclassified character due to fading of ink

Table 7 Comparative analysis of the proposed work with existing work

References	Type of Document	Feature	Classifier	Accuracy
Narang et al. [27]	Devanagari handwritten ancient Documents	HOG	Naïve Bayes	74.11%
			SVM	88.73%
			Decision Tree	59.00%
		DCT	Naïve Bayes	76.84%
			SVM	90.70%
			Decision Tree	58.15%
Avadesh and Goyal [6]	Sanskrit printed ancient Documents	CNN	CNN	**93.32%**
Narang et al. [25]	Devanagari handwritten ancient Documents	SIFT	SVM	**65.74%**
		Gabor	SVM	**91.39%**
Proposed work	Devanagari handwritten ancient Documents	CNN	CNN	**93.73%**

characters are touching or fading due to age or writing style, ink stains, lightening of ink, uneven space between characters, overlapping of characters or broken characters. So, a decline in the accuracy was observed in the present work as compared to the work done by Acharya et al.[1]. But, for Devanagari ancient documents, our proposed technique is giving the best results as compared to any other existing work. Our proposed work uses deep learning for the recognition of Devanagari handwritten anient manuscripts for the first time ever.

9 Conclusion and future scope

The authors have presented a character recognition system for the basic characters of ancient Devanagari documents. The authors have identified 33 classes and used 5484 characters for the experiments. Characters are segmented from the ancient Devanagari manuscripts. For the recognition of these characters, deep learning with the CNN model is used. Deep learning has never been used for the recognition of the handwritten Devanagari ancient manuscripts. This work is the first of its kind. Maximum accuracy of 93.73% was achieved with 75% train data and 25% test data after 30 epochs. In the present work, Devanagari handwritten character dataset is confined only to the 33 basic characters. In the future, we will extend our database to include other characters also. The number of classes will be increased. The generated database in this article can also be used for further research purposes. In future, we plan to publish our database so that it can be used by the research community.

Declarations

Conflict of interest The authors declared that they have no conflict of interest in this work.

References

1. Acharya S, Pant AK, and Gyawali PK (2015). Deep learning based large scale handwritten Devanagari character recognition. Proceedings of the 9th international conference on software, knowledge, information management and applications (SKIMA), 1–6.
2. Agarwal B, Ramampiaro H, Langseth H, Ruocco M (2018) A deep network model for paraphrase detection in short text messages. Inf Process Manag 54:922–937

3. Ahlawat S, Choudhary A, Nayyar A, Singh S, Yoon B (2020) Improved handwritten digit recognition using convolutional neural networks (CNN). Sensors 20(12):3344
4. Alizadehashraf B and Roohi S (2017). Persian handwritten character recognition using convolutional neural network. *Proceedings of the* 10*th Iranian Conference on Machine Vision and Image Processing*, 247-251.
5. Alzubi J, Nayyar A, Kumar A (2018) Machine learning from theory to algorithms: an overview. J Phys Conf Ser 1142(1):012012
6. Avadesh M and Goyal N (2018). Optical character recognition for Sanskrit using convolution neural networks. *Proceedings of the 13th IAPR International Workshop on Document Analysis Systems (DAS)*, Vienna, 447-452.
7. Bag S, Harit G (2013) A survey on optical character recognition for Bangla and Devanagari scripts. Sadhana 38(1):133–168
8. Cecotti H, Belaid A (2005) Hybrid OCR combination approach complemented by a specialized ICR applied on ancient documents. Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, pp 1–5
9. Ciresan DC, Meier U, Masci J, Gambardella LM and Schmidhuber J (2011) Flexible high performance convolutional neural networks for image classification. *Proceedings of the* 22*nd* *International Joint conference on Artificial Intelligence*, 1237-1242.
10. Diem M, Sablatnig R (2010). Recognizing characters of ancient manuscripts. *In the Proceedings of the International Conference on Computer Image Analysis in the Study of Art*: 753106-753106.
11. Erhan D, Bengio Y, Courville A, Manzagol P, Vincent P, Bengio S (2010) Why does unsupervised pre-training help deep learning? J Mach Learn Res 11:625–660
12. Farabet C, Couprie C, Najman L, Lecun Y (2013) Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell 35:1915–1929
13. Garz A, Diem M, and Sablatnig R (2010) Local descriptors for document layout analysis. Proceedings of the International Symposium on Visual Computing 29–38
14. Ghosh K, Chakraborty A, Parui SK, Majumder P (2016) Improving information retrieval performance on OCRed text in the absence of clean text ground truth. Inf Process Manag 52:873–884
15. Heyong W, Ming H (2019) Supervised Hebb rule- based feature selection for text classification. Inf Process Manag 56:167–191
16. Jangid M, Srivastava S (2018) Handwritten Devanagari character recognition using layer wise training of deep convolutional neural networks and adaptive gradient methods. J Imag 4(2):41
17. Kavitha AS, Shivakumara P, Hemantha G (2013) Skewness and nearest neighbour based approach for historical document classification. *Proceedings of the International Conference on Communication Systems and Network Technologies*, 602-606.
18. Khanduja D, Nain N, Panwar S (2016) A hybrid feature extraction algorithm for Devanagari script. ACM Trans Asian Low-Resource Language Inf Process 15(1):2
19. Kim MS, Jang MD, Choi HL, Rhee TH, Kim JH and Kwag HK (2004) Digitalizing scheme of handwritten Hanja historical documents. *Proceedings of the First International Workshop on Document Image Analysis for Libraries*: 321-327.
20. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Proces Syst 25:1–9
21. Kumar S (2016). A study for handwritten Devanagari word recognition. Proceedings of the international conference on communication and signal processing (ICCSP), 1009–1014.
22. Kumar M, Sharma RK, Jindal MK, Jindal SR (2018) Character and numeral recognition for non-Indic and Indic scripts: a survey. Artif Intell Rev 52:2235–2261
23. Lee H, Grosse R, Ranganath R and Ng AY (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the* 26*th Annual International Conference on Machine Learning,* 609–616.
24. Narang SR, Jindal MK, Kumar M (2019) Devanagari ancient documents recognition using statistical feature extraction techniques. Sādhanā 44(6):0141
25. Narang SR, Jindal MK, Ahuja S, Kumar M (2020a) On the recognition of Devanagari ancient handwritten characters using SIFT and Gabor features. Soft Comput 24:17279–17289
26. Narang SR, Jindal MK, Kumar M (2020b) Ancient text recognition: a review. Artif Intell Rev 53:5517–5558
27. Narang SR, Jindal MK and Sharma P (2018) Devanagari ancient character recognition using HOG and DCT features. Proceedings of the 5*th IEEE International Conference on Parallel, Distributed and Grid Computing* (*PDGC-2018*), Solan, India.
28. Phan TV, Nguyen KC, Nakagawa M (2016) A nom historical document recognition system for digital archiving. Int J Document Anal Recognit 19(1):49–64

29. Purkaystha B, Datta T, Islam MS (2017) Bengali handwritten character recognition using deep convolutional neural network. *Proceedings of the* 20*th International Conference of Computer and Information Technology* (*ICCIT*), 1-5.
30. Quiles MG, Romero R (2006) A computer vision system based on multi-layer Perceptrons for controlling Mobile robots. ABCM Symp Ser Mechatron 2:661–668
31. Shah KR, Badgujar DD (2013) Devanagari handwritten character recognition (DHCR) for ancient documents: a review. *Proceedings of IEEE Conference on Information and Communication Technology*, 656-660.
32. Sharma N, Patnaik T, Kumar B (2013) Recognition for handwritten English letters: a review. Int J Eng Innov Technol 2(7):318–321
33. Shelke S, Apte S (2015) A fuzzy-based classification scheme for unconstrained handwritten Devanagari character recognition. In International Conference on Communication, Information & Computing Technology (ICCICT), 1–6.
34. Singh J, Lehal GS (2014) Comparative performance analysis of feature (S)-classifier combination for Devanagari optical character recognition system. Int J Adv Comput Sci Appl 5(6):37–42
35. Singh D, Saini JP, Chauhan DS (2015). Hindi character recognition using RBF neural network and directional group feature extraction technique. In international conference on cognitive computing and information processing (CCIP), 1–4
36. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from Overfitting. J Mach Learn Res 15:1929–1958
37. Soumya A, Kumar HG (2015) Feature extraction and recognition of ancient Kannada epigraphs. Smart Innov Syst Technol 33:469–478
38. Sousa JMC, Pinto JRC, Ribeiro CS, Gil JM (2005) Ancient document recognition using fuzzy methods. *Proceedings of IEEE International Conference on Fuzzy Systems*, 833-836.
39. Sumetphong C, Tangwongsan S (2012) An optimal approach towards recognizing broken Thai characters in OCR systems. *International Conference on Digital Image Computing Techniques and Applications (DICTA)*,1-5.
40. Theeramunkong T, Wongtapan C (2005) Off-line isolated handwritten Thai OCR using island-based projection with n-gram model and hidden Markov model. Inf Process Manag 41:139–160
41. Uysel AK, Gunal S (2014) The impact of preprocessing on text classification. Inf Process Manag 50:104–112
42. Verma K, Singh M (2018) Hindi handwritten character recognition using convolutional neural network. International Journal of Computer Sciences and Engineering 6(6):909–914
43. Vinyals O, Toshev A, Bengio S and Erhan D (2015) Show and tell: a neural image caption generator. *Proceeding of the IEEE conference on Computer Vision and Pattern Recognition*, 3156-3164.
44. Yang J, Shen K, Ong C, Li X (2009) Feature selection for MLP neural network: the use of random permutation of probabilistic outputs. IEEE Trans Neural Netw 20(12):1911–1922

Affiliations

Sonika Rani Narang[1] · **Munish Kumar**[2] · **M. K. Jindal**[3]

[1] Department of Computer Science, DAV College, Abohar, Punjab, India

[2] Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda, India

[3] Department of Computer Science & Applications, Panjab University Regional Centre, Muktsar, Punjab, India