ISSN: 0974-5823 Vol. 7 No. 2 February, 2022

*International Journal of Mechanical Engineering*

# An Accurate Deep-Learning Model for Handwritten Devanagari Character Recognition

**Anuj Bhardwaj**

Research scholar, Dept. of CS & IT, M.J.P. Rohilkhand University, Bareilly (U.P.), India

## Abstract

Handwritten character recognition is increasingly important in a variety of automation areas, for example, authentication of bank signatures, identification of ZIP codes on letter addresses, and forensic evidence, etc. Handwritten character recognition is the process where a machine detects and recognizes characters from a text image and converts that processed data into a code that is understood by the machine. This is a fundamental but challenging task in the field of pattern recognition. In this paper, we have used a new public image dataset for the Devanagari script character: the Devanagari character dataset (DCD). This considered dataset contains 92 thousand images of 46 different classes of characters in the Devanagari script, fragmented from handwritten documents. The paper also explores the challenges faced in the identification of Devanagari characters. This paper proposes a deep learning based convolutional neural network (CNN) architecture for the recognition of those handwritten characters in an unrestricted environment, with datasets. Deep convolutional neural networks have shown better results than traditional shallow networks in many recognition tasks. While keeping distance from the routine approach of character recognition by Deep CNN, we focus on the use of dropout and dataset augmentation approaches to improve test accuracy. By applying these techniques to Deep CNN, we were able to increase the test accuracy to about 0.98 percent. The proposed architecture achieved the highest test accuracy of 98.13% on the considered dataset. The results indicate that the proposed model may be a strong candidate for handwritten character recognition and automatic handwritten Devanagari script character recognition applications.

## Keywords

Devnagari script character, Handwritten Devanagari Character, Deep Learning, Convolutional Neural Network (CNN) model, Handwritten character recognition, Deep convolutional neural networks, Image processing, Computer vision, Devanagari handwritten character dataset

## Introduction

Handwritten character recognition is considered to be one of the most challenging and appealing research areas in the field of pattern recognition and computer vision. It is the natural way of interacting with the computer. (Acharya et al., 2015) Due to the critical factors of differences in writing patterns and cursive text, and the similarity of various characters in Devanagari script characters, recognition research is time-consuming and challenging. It has been a field of great interest for researchers and scientists. Character recognition is the process where the machine detects and recognizes the characters from a text image and converts that processed data into a code which is understood by the machine. It is a fundamental yet challenging task in the field of pattern recognition. Recognition of handwritten Devanagari script characters may be performed online or offline. Online character identification is relatively simple due to the temporal-based character properties such as form, number of strokes, distance, and direction of writing. Offline character recognition implementation is complex due to variations of writers and fonts. Recognition is termed as optical character recognition (OCR) as they deal with characters which are optically processed and not magnetically processed.(Sinha, 2009) The literature shows a high accuracy rate for recognition of characters and isolated words in optical character recognition (OCR) or printed text; however, there is a need for a competent handwritten character recognition system capable of generating a high degree of accuracy in handwritten text recognition.(Fujisawa, 2008)(Plamondon & Srihari, 2000)(Steinherz et al., 1999)(Arica & Yarman-Vural, 2001)

Concept of character recognition was introduced by Carey in 1870 when the researcher developed the retina scanner.(Acharya et al., 2015) Then in 1890, the sequential scanner was introduced which brought in the revolution for reading machines. The advancement of technology in terms of both scanning devices and the computation power has also led to the development in the field of character recognition. Today, a number of OCR's are present in the market which can perform recognition task with minimal error rate. Some of these are also available for handheld and portable devices like cellular phones.

Lately, several such systems have been developed for more complex languages such as Chinese, Korean, Arabic, etc. but the accuracy and consistency are not at par as compared to those for the English language. One of the reasons for this is that English Script is not as complex and complicated as that of Chinese or Devanagari Script. Pertaining to this reason, the existing character recognition faces difficulties in understanding the not so popular languages like Sanskrit, Hindi, and Tamil, and hence are not useful for people whose native language is not English. Another shortcoming of the majority of the OCR's is that they fail to understand the context of the text.

Character classification is an important part of handwritten character recognition that plays an essential role in many computer vision problems like OCR, license Plate recognition, etc. Development of a recognition system is an emerging need for digitizing handwritten hindi documents that use Devnagari characters. Optical Character Recognition systems are least explored for Devnagari characters. (Pant et al., 2012)(Dongre & Mankar, 2011) present a few approaches for segmentation and recognition of

Devnagari characters. The major challenging task while creating a recognizer and classifier for the Devnagari scripts is that they have a large number of symbols as compared to languages like English. English contains mainly 52 characters while a script like Devanagari consists of more than 200 symbols. Almost all the Indic languages are similar to each other. Some of them have a characteristic of allowing characters to be combined together to form another character, generally referred to as 'sayuktakshar' where Sayukt stands for combined and 'akshar' means word. Another challenge is the identification of vowel modifiers or 'matras'. Vowel modifiers change the UNICODE value of that character, and hence, in order to identify the correct character and its Unicode value, we need to write script specific rules. These modifiers may be present at the left, right, bottom or top of the character. In some of the cases, the modifier may be present at two positions on the same character. Hence, it is required to identify these modifiers correctly to reduce the error in classification. So this paper considering a publicly available Devnagari Character Dataset, of 92 thousand images of 46 Devnagari characters. Then, this paper proposing a Deep learning Convolutional Neural Network model to classify the characters in DCD. Introduction of multilayer perceptron network has been a milestone in many classification tasks in computer vision.(Quiles & Romero, 2005) But, performance of such a network has always been greatly dependent on the selection of good representing features.(Ruck et al., 1990)(Yang et al., 2009) Deep Neural Networks on the other hand do not require any feature to be explicitly defined, instead they work on the raw pixel data generating the best features and using it to classify the inputs into different classes.(Lee et al., 2009) Deep Neural networks consist of multiple nonlinear hidden layers and so the number of connections and trainable parameters are very large. Besides being very hard to train, such networks also require a very large set of examples to prevent overfitting. One class of DNN with comparatively smaller set of parameters and easier to train is Convolutional Neural Network.(LeCun et al., 1998) The ability of CNN to correctly model the input dataset can be varied by changing the number of hidden layers and the trainable parameters in each layer and they also make correct assumption on the nature of images.(Krizhevsky et al., 2012) Like a standard feed forward network, they can model complex non-linear relationship between input and output. But CNN have very few trainable parameters than a fully connected feed-forward network of same depth. CNNs introduce the concept of local receptive field, weight replication and temporal subsampling(LeCun et al., 1990) which provide some degree of shift and distortion invariance. CNNs for image processing generally are formed of many convolution and sub-sampling layers between input and output layer. These layers are followed by fully connected layers thereby generating distinctive representation of the input data. Beside image recognition, CNNs have also been used for speech recognition.(Abdel-Hamid et al., 2012)(Sainath et al., 2013) Although deep convolutional neural networks have a small and inexpensive architecture compared to standard feed forward network of same depth, training a CNN still requires a lot of computation and a large labeled dataset. Training such a network was not so effective and did not produce any superior result to traditional shallow network, until recently. With the availability of large labeled dataset like IMAGENET, NIST, SVHN, development of state of the art GPUs and introduction of unsupervised pre-training phase, CNNs have at present proven to surpass traditional feed forward network in a number of classification tasks. In CNNs, initializing the weight randomly and applying gradient descent and backpropagation to update the weights seems to generate poorer solution for a deep network.(Larochelle et al., 2009) So, generally, greedy layer wise unsupervised pre training is applied prior to supervise training. Why such unsupervised training helps is investigated in (Erhan et al., 2010).

## Literature review

This section covers the related work on handwritten character recognition of English, Arabic, Meetei Mayek, Bangla, Sanskrit, Hindi, and Chinese etc. handwritten characters is described. These following research work perform character recognition with different algorithms and they also used different deep learning for better accuracy and recognition.

Many studies have been conducted on handwritten English, Arabic, Chinese and Urdu character recognition; however, Devanagari character recognition has not been explored to the same extent. English and Arabic handwritten character recognition can be classified into two approaches, conventional approaches and deep learning-based approaches.

Conventional approaches depend on manual feature extraction by experts whereas deep learning-based techniques automatically extract features from raw images. Conventional techniques cannot extract features from images in their raw form. Machine learning experts have struggled to design feature extractors that extract discriminative features from raw data into vectors as an input to classifiers for pattern recognition.(Abandah et al., 2008)

In a notable early work for isolated handwritten character recognition, Roy et. al., 2017 they developed Supervised Layer Wise training of a Deep Convolutional Neural Network (SL-DCNN) architecture. It had a set a new benchmark of 9.67% error rate on the rather difficult CMATERdb 3.1.3.3 handwritten Bangla isolated compound character dataset. The model represented a lowering of error rated by nearly 10% from the previously set benchmarks on the same and is an excellent result in the area of Bangla compound character recognition.(Roy et al., 2017) Alom et. al., 2017 they have used a combination of dropout and many filters on a dataset CMATERdb 3.1.1 and evaluate the performance of CNN and DBN with the integration of Deep Learning. They perform many experiments and conclude that CNN with Gabor feature and dropout get better accuracy for BANGLA digit recognition to compare to another technique.(Alom et al., 2017) El-sawy et. al., 2017 they had used convolution neural network (CNN) handwritten character recognition for ARABIC character and show the results were promising with a 94.9% classification accuracy rate on testing image scripts.(El-Sawy et al., 2017) Savitha Attigeri, 2018 presented a neural network based offline character recognition from the handwritten script using offline mode without using feature extraction from the scanned image. They had used layers and got accuracy for 90.19/5 from 100 neurons.(Attigeri, 2018) Adnan et. al., 2018 had analyzed different deep convolution neural networks (DCNNs) and concluded that DenseNet can generate more fruitful results in BANGLA character recognition. They also mention that the accuracy could achieve up to 98% for digit, alphabet and special character recognition.(Adnan et al., 2018) Jangid et. al., 2018 have used deep convolution neural network (DCNN) and adaptive gradient methods for DEVENAGRI character recognition. They had also used Network Architecture – 6 and RAMSProp optimizer method. They achieved about 98% recognition accuracy using ISIDCHAR and V2DMCHAR dataset.(Jangid & Srivastava, 2018) D.S. Joshi & Risodkar, 2018 focused on GUJARATI character recognition with the help of K-NN and Neural Network. They also used filtering, edge detection, morphological transformation for image processing and achieved 78.6% accuracy on a dataset.(Joshi & Risodkar, 2018) Das et. al., 2018 had presented a popular CNNs with kernel size, pooling methods and activation function for BANGLA handwritten character recognition. Results proved that CNNs architecture can improve the character recognition accuracy.(Sen et al., 2018) Yu Weng & Chunel Xia, 2019 developed a convolution neural network to handwritten character recognition of SHUI. After performing the different experiments and they concluded that characters can be classified easily using CNN models. They also made a comparison with their developed model and another available model.(Weng & Xia, 2019) Gan et. al., 2019 used 1-D CNN for CHINESE character recognition with the implementation of the sequential structure of handwritten character and also derived that this model of CNN is very compact and runes faster than other architecture like 2-D CNN and RNN.(Gan et al., 2019) Kavitha & Srimathi, 2019 have used CNNs for TAMIL handwritten character recognition. They achieved a testing accuracy of 97.7% using the HP Lab's dataset. They also proved that the CNNs are a more accurate paradigm for character recognition from the handwritten script.(Kavitha & Srimathi, 2019) Saha & Saha, 2018 introduced a deep convolution neural network with techniques like Divide and Merge Mapping and Optimal Path Finder for BANGLA handwritten character recognition. They achieved the results with the lowest complexity of the provided parameters on a small network as well as a large network of the dataset.(Saha & Saha, 2018) Uki et. al., 2019 used CNN's depth and wavelet transformation on the image. They had used different approached like max-pooling, probabilistic voting, feature extraction, feature correction, etc. with the help of all these methods they found that the accuracy of the character recognition is increased more than 4% with the previous techniques.(Ukil et al., 2019)

**Table 1: Detailed comparison of approaches uses by different author for character recognition**

| Sr. No. | Author(s) | Year Of Publication | Approach | Dataset | Pre-processing | Result/ Accuracy |
|---|---|---|---|---|---|---|
| 1 | (Roy et al., 2017) | 2017 | Supervised layer wise training of a deep convolution neural network (SL-DCNN) | CMATERdb 3.1.3.3 | The available dataset used, they resized the image to 30 pixels of height and width. | CNN model provides an error rate of 9.67% compared to 15.96% for the DCNN. |
| 3 | (Alom et al., 2017) | 2017 | Deep Belief Network (DBN), CNN with dropout, CNN with dropout and Gaussian filters, CNN with dropout and Gabor Filters. | CMATERdb 3.1.1 | Raw images used as input and passed to the next phase (Normalization). | DBN produced 97.20 % CNN + GABOR + DROPOUT achieved an accuracy of 98.78% |
| 4 | (El-Sawy et al., 2017) | 2017 | Convolution Neural Network (CNN) | 16800 images 13340 images for Training and 3360 images for testing. | The input to a convolution layer is the M X M X C; where M is the height and width of the image and C is a number of channels per pixel. | They achieved an accuracy of 94.9%. |
| 2 | (Attigeri, 2018) | 2018 | Artificial Neural Network (ANN) | Own dataset of 49 characters and 4840 samples. | Each individual character is uniformly resized to 30 X 20 pixels. | The character recognition accuracy of 90.19% they have achieved. |
| 6 | (Adnan et al., 2018) | 2018 | Deep Belief Network(DBN), Stacked Auto encoder (AE), CNN, DenseNet | CMATERdb 3.11 | Each digit has 600 images that are rescaled to 32 X 32 pixels. | Digit-10 --- 99.13%. Alphabet - 50 --- 98.31%. Special Character - -- 98.18% accuracy. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | (Jangid & Srivastava, 2018) | 2018 | Deep Convolution neural network (DCNN) | ISIDCHAR and V2DMDCH AR database. | A normalization process has to follow for converting the image h X w X c size to m m X m X c size where m represents the height and width of an image. | Using a DCNN layer-wise training model, they obtained 98% recognition accuracy. |
| 10 | (Joshi & Risodkar, 2018) | 2018 | K-NN classifier and Neural Network | Own database with 30 samples | RGB to gray conversion, skew correction, filtering, morphological operation | Got accuracy nearby 78.6% |
| 12 | (Sen et al., 2018) | 2018 | Convolutional Neural Network (CNN) architecture | Own database with 200 samples from 100 different persons. | Resized all images to 28 X 28 pixels. | They have achieved an accuracy of 99.40% for character recognition. |
| 5 | (Weng & Xia, 2019) | 2019 | Deep Neural Network (DNNs) | 400 types of pictures used as a dataset. | 37500 data are used as a training and 12500 data used as test data. Shui character image normalized to 52 * 52 pixels. The class label is added to the clustering results. | The final test accuracy is around 93.3% achieved. |
| 7 | (Gan et al., 2019) | 2019 | 1-D CNN Databases | ICDAR-2013 IAHCC-UC AS2016 | Chainese character images rescaled into 60 X 60 pixel size. | 98.11 % on ICDAR-2013 and 97.14% on IAHCC-UCA2016 dataset. |
| 8 | (Kavitha & Srimathi, 2019) | 2019 | CNN model total of nine layers; five convolution layers, two max pooling layers, and two fully connected layers. | HPL-taiml-is o-char 156 characters 500 samples for each class 82,928 total samples. | Convert into a grayscale image from the original image. | Testing accuracy of 97.7%. |
| 11 | (Saha & Saha, 2018) | 2019 | Divide and Merge Mapping (DAMM) | Own database with 1,66,105 images | Resize all images to 128 X 128 pixels. | Achieved a training accuracy of 99.13% along with a validation accuracy of 98.87% |
| 13 | (Ukil et al., 2019) | 2019 | Convolutional Neural Networks (CNNs) | PHD_Indic_11 dataset | RGB to grayscale conversion and resized image to 28 X 28 pixels. | CNNprob performed the best with an accuracy of 95.45%, precision of 95.36%, recall of 95.32% and f-measure of 95.33%. |

In the case of deep learning approaches, the use of convolutional neural networks for the recognition of Modified national institute of standards and technology (MNIST) and Extended modified national institute of standards and technology (EMNIST) datasets has attracted the attention of numerous researchers, who have applied these models to languages with handwritten character recognition, including English (Deng, 2012), Hangul (Draman et al., 2009), and Chinese (Faraoun & Boukelif, 2006). Arabic-related research studies using deep learning models and techniques have been reported in this research.(Weng & Xia, 2019) The authors first designed an image processing module, and constructed a data set, for mobile devices. They proposed a lightweight CNN for optical character recognition for the dataset. Another work addresses the variability in writer's handwriting (Cilia et al., 2019), in which a feature-ranking technique was adopted. The authors considered different univariate measures to produce a feature ranking and proposed a greedy search approach for choosing the feature subset able to maximize the classification results. Raymond et al. (Ptucha et al., 2019) presented a fully convolutional network architecture that outputs arbitrary length symbol streams from handwritten text. A preprocessing step normalizes input blocks to a canonical representation, which negates the need for costly recurrent symbol alignment correction. The authors introduced a probabilistic character error rate to correct errant word blocks.

A recent study by Chaouki Boufenar et al. (Boufenar et al., 2018), investigated the use of convolutional neural networks for offline Arabic handwritten character recognition. Their architecture consisted of five layers in which three convolutional layers with a max pool were connected to two fully connected layers. They used OIHACDB-28 for training and evaluation of the model, gained a result of 97.32% accuracy. The CNN model was trained with a dropout technique under the Theano framework. Ahmed El-Sawy et al. (El-Sawy et al., 2016) suggested the Deep convolutional neural networks (DCNN)model for the recognition of isolated handwritten Arabic characters. They proposed a dataset referred to as Arabic handwritten characters dataset AHCD. The model was trained with an optimization method for 30 epochs that resulted in a significant increase in performance and a 94.9% classification accuracy.

The available literature shows unsatisfactory research results compared to other languages. In (Decerbo et al., 2004), the Byblos Pashto OCR system was proposed for script-free OCR using HMMs. This system was also subsequently tested for Chinese, English, and Arabic text with success. As previously mentioned, Devanagari Intelligent character recognition and Optical character recognition (OCR) area are the least explored to date. Based on this discussion, it can be concluded that there is a lack of a Devanagari handwritten character dataset. In addition, a research gap exists for classification of Devanagari handwritten characters based on deep learning techniques, such as the CNN. The current research fills this gap by proposing a CNN model and a Devanagari handwritten character dataset. For this purpose, a model was developed and extensive experimentation conducted. These are described in detail in Sections 3 and 4.

## Devanagari Handwritten Character Dataset

### Devanagari Script

Devanagari is part of the Brahmic family of scripts of Nepal, India, Tibet, and South-East Asia.(Fischer, 2004)(Gaur, 1992) The script is used to write Nepali, Hindi, Marathi and similar other languages of South and East Asia. The Nepalese writing system adopted from Devanagari script consists of 12 vowels, 36 base forms of consonant, 10 numeral characters and some special characters. Vowel characters are shown in Table 2, consonants characters in Table 3 and numeral characters in Table 4. Moreover, all 36 consonants could be wrapped with the vowels generating 12 other derived forms for each branch of consonant character. One such example for "ta (tabala)" and "pa" is shown in Table 5.

| Devanagari Character | अ | आ | इ | ई | उ | ऊ | ए | ऐ | ओ | औ | ऑ | ऒ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UNICODE | 905 | 906 | 907 | 908 | 909 | 090A | 090F | 910 | 913 | 914 | 911 | 912 |

Table 2: Devanagari vowels with UNICODE

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Devanagari Character | क | ख | ग | घ | ङ | च | छ | ज | झ | ञ | ट |
| UNICODE | 915 | 916 | 917 | 918 | 919 | 091A | 091B | 091C | 091D | 091E | 091F |
| Devanagari Character | ठ | ड | ढ | ण | त | थ | द | ध | न | प | फ |
| UNICODE | 920 | 921 | 922 | 923 | 924 | 925 | 926 | 927 | 928 | 092A | 092B |
| Devanagari Character | ब | भ | म | य | र | ल | व | श | ष | स | ह |
| UNICODE | 092C | 092D | 092E | 092F | 930 | 932 | 935 | 936 | 937 | 938 | 939 |
| CHARACTER | क्ष | त्र | ज्ञ | These three consonants have no specific UNICODE | | | | | | | |

Table 3: Devanagari consonants with UNICODE

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ० | १ | २ | ३ | ४ | ५ | ६ | ७ | ८ | ९ |
| 966 | 967 | 968 | 969 | 096A | 096B | 096C | 096D | 096E | 096F |

Table 4: Devanagari numerals

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| त | ता | ति | ती | तु | तू | ते | तै | तो | तौ | तं | तः |
| प | पा | पि | पी | पु | पू | पे | पै | पो | पौ | पं | पः |

Table 5: Derived forms of consonant “ta (tabala)” and “pa” when wrapped with vowels.

**Devanagari Handwritten Character Dataset**

Devanagari Handwritten Character Dataset is created by collecting the variety of handwritten Devanagari characters from different individuals from diverse fields. Handwritten documents are than scanned and cropped manually for individual characters. Each character sample is 32x32 pixels and the actual character is centered within 28x28 pixels. Padding of 0 valued 2 pixels is done on all four side to make this increment in image size. The images were applied gray-scale conversion. After this the intensity of the images were inverted making the character white on the dark background. To make uniformity in the background for all the images, we suppressed the background to 0 value pixel. Each image is a gray-scale image having background value as 0.

Devanagari Handwritten Character Dataset contains total of 92,000 images with 72,000 images in consonant datasest and 20,000 images in numeral dataset. Handwritten Devanagari consonant character dataset statistics is shown in Table 6 and handwritten Devanagari numeral character dataset statistics is shown in Table 7.

Table 6: Consonant Character Dataset

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Devanagari Character (Class) | क | ख | ग | घ | ङ | च | छ | ज | झ | ञ | ट |
| Individual statistics | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 |
| Devanagari Character (Class) | ठ | ड | ढ | ण | त | थ | द | ध | न | प | फ |
| Individual statistics | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 |
| Devanagari Character (Class) | ब | भ | म | य | र | ल | व | श | ष | स | ह |
| Individual statistics | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 | 2,000 |
| Devanagari Character (Class) | क्ष | त्र | ज्ञ | | | | | | | | |
| Individual statistics | 2,000 | 2,000 | 2,000 | | | | | | | | |
| **Total** | **72,000** | | | | | | | | | | |

Table 7: Numeral Dataset

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Devanagari Character (Class) | ० | १ | २ | ३ | ४ | ५ | ६ | ७ | ८ | ९ |
| Individual statistics | 966 | 967 | 968 | 969 | 096A | 096B | 096C | 096D | 096E | 096F |
| **Total** | **20,000** | | | | | | | | | |

**Challenges in Devanagari Character Recognition**

There are many pairs in Devnagari script that has similar structure differentiating each with structure like dots, horizontal line etc. Some of the examples are illustrated in Table 8. The problem becomes more intense due to unconstrained cursive nature of writings of individuals. Two such examples are shown in Table 9.

| | | |
|---|---|---|
| छ | ६ | Difference being horizontal line at top |
| ड | ङ | Difference being presence of single dot on right side |
| द | ढ | Difference being presence of small circle and small down stroke line |

Table 8: Structural formation of characters

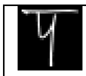
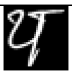
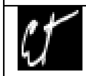
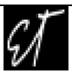
| | | | |
|---|---|---|---|
|  | प |  | य |
|  | ध |  | घ |

Table 9: Different characters written similarly

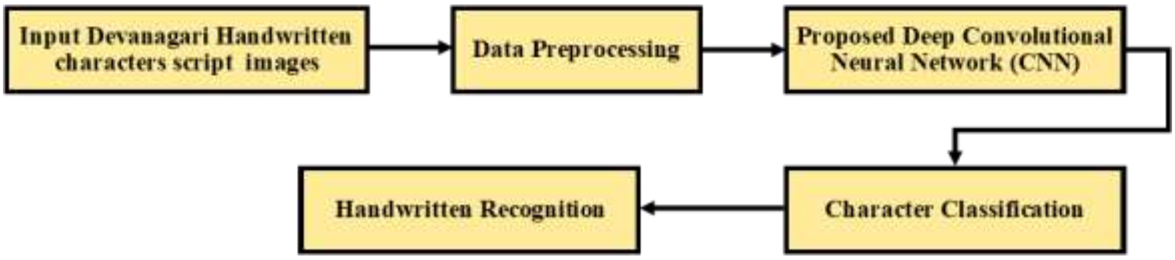
Proposed Model for Devanagari handwritten character recognition

This section proposed a means of Devanagari handwritten character classification and recognition model as shown in Figure 1. This recognition model relies on a CNN applied for the Devanagari handwritten character dataset with a feature mapped output layer. This suggested CNN model classifies Devanagari characters into 46 different classes. A detailed explanation of the suggested model is presented in the following subsections.

Figure 1: Block diagram of proposed model framework

### Devanagari handwritten characters scripts images dataset

A data-enriched dataset plays a vital role in the generation of accurate results in research activities related to deep learning. A concise and precise dataset is required for a true evaluation of mathematical models that are applied to it. Moreover, to achieve benchmark results in deep learning, a standard publicly available dataset is mandatory. During the experimental phase of this research, we considered dataset named Devanagari Handwritten Character Dataset publically available on UCI repository.[1] Detailed description about the data is described in previous section.


Input Devanagari Handwritten characters script images
Data Preprocessing
Proposed Deep Convolutional Neural Network (CNN)
Character Classification
Handwritten Recognition

### Data Preprocessing

Preprocessing steps were applied to the proposed dataset to prepare images for subsequent phases. Image preprocessing consisted of operations on images at the lower abstraction level with the aim of improving the image data. This improvement suppresses undesired distortions in the image dataset, or enhances important details or features that are essential for further processing.

The first step of the preprocessing phase was the removal of noise from images using Gaussian blur. The digital scanner induced spikes of noise into the scanned images, as shown in Figure 2(a). These spikes of noise were removed using the notable work of Soman.(Ganga Gowri & Soman, 2018) The second step was smoothing of the image using the Gaussian function. This can be considered to be low-pass filtering in a non-uniform manner, which conserves the low frequency, and decreases the noise and insignificant details in an image. This was accomplished by convolving the Gaussian kernel with an image.

$$G_{2D}(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where σ is the standard deviation, and x and y are location indices. The standard deviation controls the variance around the Gaussian distribution mean value, and establishes the blurring effect around certain pixels. In our study, we used σ = 2.8, which generates a good smoothing effect to suppress the scanner-induced noise. The images were then converted to gray-scale and, finally, the images were resized to 32 × 32 pixels and the aspect ratio was held constant. A CNN was subsequently used for detection and classification of features.

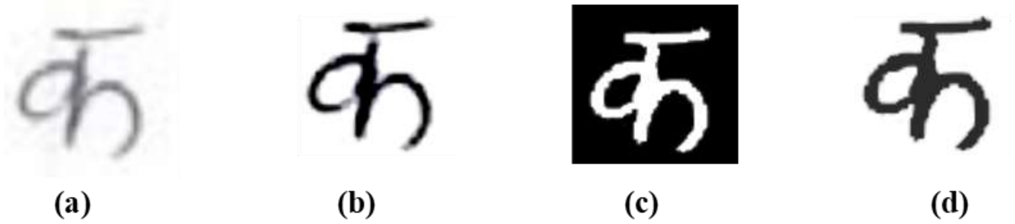


Figure 2: Image dataset denoising steps: (a) scanner-induced noise; (b) denoised image; (c) Binarized Image; (d) application of Gaussian blur filter

### Convolutional Neural Networks

Convolutional Neural Network (CNN or ConvNet) is a biologically-inspired trainable machine leaning architecture that can learn from experiences like standard multilayer neural networks. ConvNets consist of multiple layers of overlapped tiling collections of small neurons to achieve better representation of the original image. ConvNets are widely used for image and video recognition. There are three main types of layers used to build a ConvNet architecture.

1) **Convolution Layer:** The convolution layer is the core building block of a convolutional neural network. It convolves the input image with a set of learnable filters or weights, each producing one feature map in the output image.
2) **Pooling Layer:** The pooling layer is used to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. The pooling layer takes small rectangular blocks from the convolution

[1] https://archive.ics.uci.edu/ml/datasets/Devanagari+Handwritten+Character+Dataset

layer and subsamples it to produce a single output from that block. There are several ways to do this pooling, such as taking the average or the maximum, or a learned linear combination of the neurons in the block.

3) **Fully-Connected Layer:** The fully-connected layer is used for the high-level reasoning in the neural network. It takes all neurons in the previous layer and connects it to every single neuron it has. Their activations can be computed with a matrix multiplication followed by a bias offset as a standard neural networks.

A simple convolutional neural network similar to the one used in our recognition system is shown in Figure 3. The input layer consists of the raw pixel values from the 32x32 grayscale image and has no trainable parameters. The first convolution layer has 4 feature maps with 784 units/neurons each (28 x 28). Each feature map is shown in figure as 2D planes and they have different set of weights. All the units in a feature map share the same set of weights and so they are activated by the same features at different locations. This weight sharing not only provides invariance to local shift in feature position but also reduces the true number of trainable parameters at each layer. Each unit in a layer receives its input from a small neighborhood at same position of previous layer. So the number of trainable weights associated with each unit in a convolutional layer depends on the chosen size of the neighborhood of previous layer mapped to that unit. Since all the units are activated only from the input taken from a local neighborhood they detect local features such as corners, edges, end-points. This concept of local receptive field is inspired from study of the locally-sensitive, orientation selective neurons in the cats visual system.(Hubel & Wiesel, 1962)

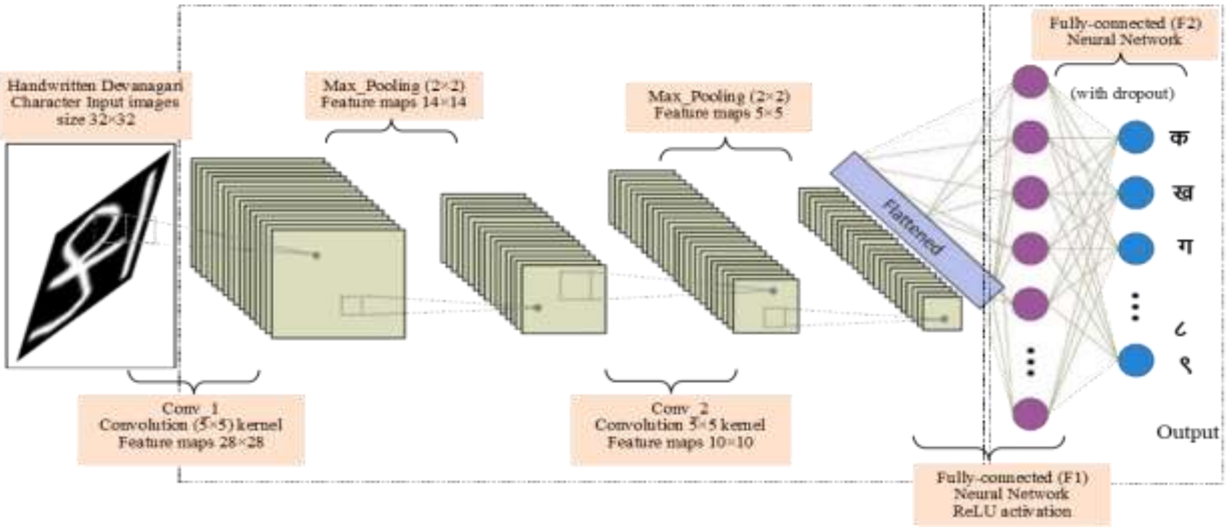


Figure 3: Proposed Convolutional Neural Network

For a 5x5 kernel as shown in Figure 3 the number of input weights for each unit is 25. In addition the units also have a trainable bias. The total number of units in a layer depends upon the size of kernel in the previous layer and overlap between the kernels.

The convolutional layer is followed by a subsampling/pooling layer. Sub sampling layer reduces the resolution of the feature map from convolution layer by averaging the features in the neighborhood or pooling for a maximum value. Because the exact position of features vary for different images of the same character, it is more desirable that the system does not learn the absolute position of feature but instead learn the relative position among the features. The pooling layer helps achieve this goal and makes the classifier more immune to shift and distortion. It aggregates the information within a set of small local regions and produces a pooled feature map as output. The number of units in a pooling layer thus depends upon the local region of the previous convolution layer feeding input to the units in pooling layer. So for a non-overlapping scheme and a 2x2 region from previous layer connected to units in pooling layer the dimension of feature maps reduce to half of the convolution layer. The max pooling method checks for the maximum value on its local receptive field, multiplies it by a trainable coefficient, adds a trainable bias and generates output.

The second convolution layer follows this sub-sampling layer. Each feature map in C2 layer is generated by taking input from S1. The units in C2 get their input from the 5x5 neighborhood at identical position of some layers in S1 and not all. The reason for not connecting C2 feature maps to all feature maps of S1 layer is to reduce the total number of trainable parameters and, this also introduces randomness in providing input to different feature maps with the assumption that this will help them to learn complementary features with one another. The output of this convolution layer is subsampled, convolved and forwarded to fully connected layer. From this point we obtain a 1D feature vector. The fully connected layers model the input by applying non-linearity like in a traditional feed-forward network. The type of nonlinearity used is ReLU-non linearity

$$f(x) = \max(0, x)$$

The reason for using it instead of the widely popular nonlinear functions like

$$f(x) = \tanh(x)$$

and

$$f(x) = (1 + exp^{-x})^{-1}$$

is because training with gradient-descent is comparatively much faster for ReLU than the other non-linearities.(Nair & Hinton, 2010) The depth of the network and the size of different layers to be used depends greatly on the dataset and the problem domain. Furthermore, the number of feature maps in a layer, the size of the kernel on each layer and the choice of non-overlapping or overlapping kernel and the extent of overlap also produces different results. So, in our case we tested different architectures by varying these parameters and presented results of the architecture producing the highest accuracy on the test data set. The result of the tests are summarized on the Experimental setting and results section.

The large and deep architecture of CNN with large bank of trainable parameter makes it susceptible to overfitting. While training the deep networks, it is very difficult to find optimal hyper parameters of the functions that share the parameters. These networks being large, require large amount of training data. Below given are some approaches we used to prevent our model from overfitting.

1. **Dataset augmentation:** For the better recognition models, we require to have more training samples while training the system.(Kobetski & Sullivan, 2013) This can be achieved by augmenting available dataset by mirroring, rotation, jittering, noise injection and random crops.
2. **Dropout:** Dropout simply refers to dropping out units; units representing both hidden and visible in the deep network.(Srivastava et al., 2014) We temporarily remove the random units from the network along with all its inlet and outlet connections. For each training iterations, there will be new lighter network that remains after dropping the random units from the common denser architecture which will be sampled and trained. Each unit is retained with the fixed probability of $p$ independent of other units and we set 0.5 for $p$, the number being optimal choice for most of the cases.

### Experiments and Result

We tested the dataset with different architectures by varying depth, width and number of parameters of network. The results of two of those experiments are presented in the coming sections. The first model is very wide and deep and consists of a large number of parameters. It will be referred to as model A in the coming section. It consists of three convolution layers and one fully connected layer. The sequence of the layers in model A is shown in Figure 4., where model consist of three convolution layer, three Rectified Linear Unit Layer, two Normalization layer implementing Local Response Normalization, two pooling layer implementing max pooling, one Dropout layer and one Fully Connected Layer and a accuracy layer for test set and a Softmax Loss layer that computes multinomial logistic loss of the softmax of its input.

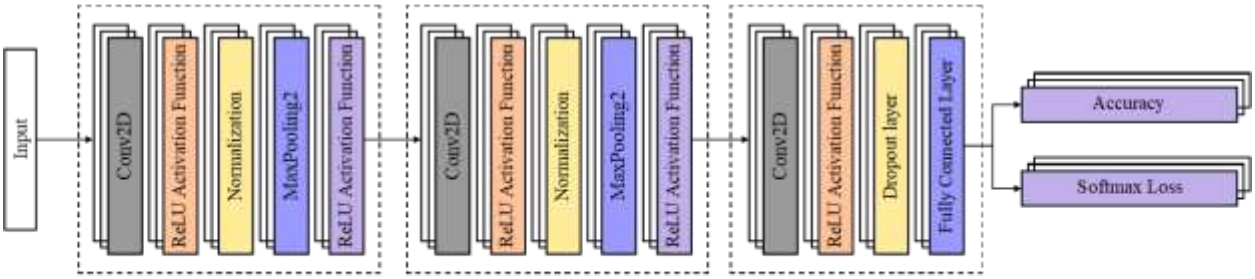


Figure 4: Architecture of model A- deep CNN model for handwritten character recognition.

The second model is derived from the lenet family. It has a shallower architecture and consists of fewer number of parameters than model A. It will be referred to as model B in the coming section. It consists of two convolutional layers followed by two fully connected layers. The sequences of layers in model B is shown in Figure 5 where each notation holds similar meaning as discussed for model A.

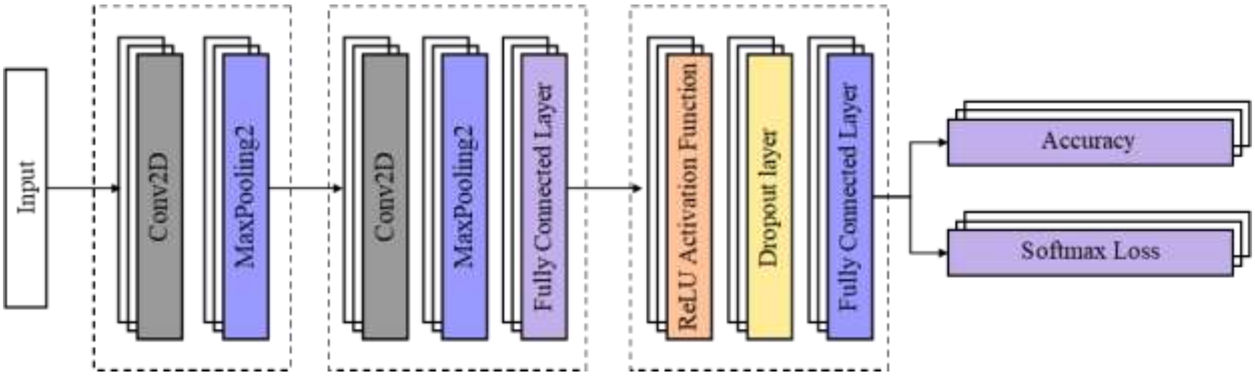


Figure 5: Architecture of model B- deep CNN model for handwritten character recognition.

In all cases, Convolution is implemented with overlapping Filter (Kernel) of Size 5∗5 and stride 1 on both direction. Pooling is implemented with a non-overlapping Filter of size 2 ∗ 2 and stride 2 on both directions. Local response Normalization is achieved by dividing each input value by the expression

$$\left(1+\left(\frac{\alpha}{n}\sum_{i} x_i^2\right)\right)^{\beta}$$

,where n is the size of each local region, and the sum is taken over the region centered at that value. The value of αparameter used is 0.001 and β-parameter is 0.75.

Our deep neural network was trained on the DCD as a multi-class classification problem. For both the models, the standard back-propagation on feed-forward net is implemented by stochastic gradient descent(SGD) with momentum of 0.9. The mini-batch size is 200 and the network was trained for 50 epochs. The base learning rate was initialized for all trainable parameters at 0.005 for Model A and 0.001 for Model B. The learning rate was updated by an inverse function using the relation

$$LR = BLR \times (1 + \gamma \times iterations)^{-power}$$

Where BLR is the Base Learning Rate and iterations is the number iterations completed. The value of γ was set to 0.0001 and power was set to 0.75.

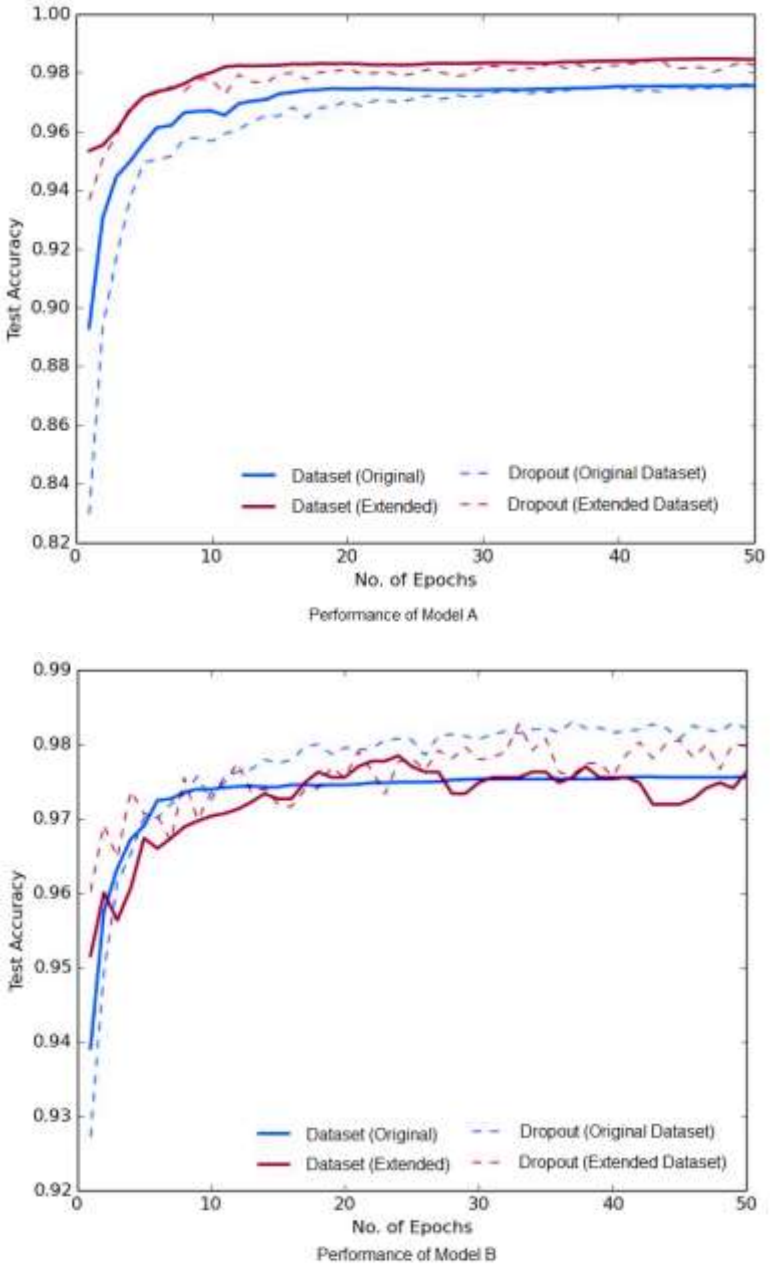


Figure (a)

Figure (b)

Figure 6: Accuracy testing of the models

The result of training for 50 epoch is presented in Figure 6. Test Accuracy remained nearly constant after 50 epochs. For model A, Extending Dataset showed superior result in Test Accuracy. So, increasing number of training sample is effective to increase performance of wide and deep network with large bank of parameters. The highest testing accuracy obtained for Model A is 0.98471. For model B, addition of dropout showed better improvement in Test accuracy. However, extending dataset also resulted slight improvement in Test accuracy. The highest value of Testing Accuracy obtained for this model is 0.981326.

## Conclusion

The presented research work proposed a deep learning CNN model that provide a high accuracy rate ≈ 98.13% to recognizing handwritten Devnagari Character, the model is trained using the publically available Devnagari Character Dataset for any researcher. It consists 92 thousand images of 46 different characters of Devnagari script. The proposed model explored the challenges in classification of characters in Devnagari Dataset. The challenges result due to the fact that the dataset consists many characters that are visually similar or written in a similar way by most people. Also, In Devnagari script, the base form of consonant characters can be combined with vowels to form additional characters which is not explored in this research. For recognition, we proposed two deep learning models to train the dataset. We also analyzed the effect of dropout layer and dataset increment to prevent overfitting of these networks. The experimental results suggested that Deep CNNs with added Dropout layer and Dataset increment technique can result in very high test accuracy even for a diverse and challenging dataset like ours.

## References

1. Abandah, G. A., Younis, K. S., & Khedher, M. Z. (2008). Handwritten Arabic character recognition using multiple classifiers based on letter form. *Proceedings of the 5th International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA)*, 128–133.
2. Abdel-Hamid, O., Mohamed, A., Jiang, H., & Penn, G. (2012). Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4277–4280.
3. Acharya, S., Pant, A. K., & Gyawali, P. K. (2015). Deep learning based large scale handwritten Devanagari character recognition. *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 1–6.
4. Adnan, M., Rahman, F., Imrul, M., AL, N., & Shabnam, S. (2018). Handwritten Bangla character recognition using inception convolutional neural network. *Int. J. Comput. Appl*, *181*(17), 48–59.
5. Alom, M. Z., Sidike, P., Taha, T. M., & Asari, V. K. (2017). Handwritten bangla digit recognition using deep learning. *ArXiv Preprint ArXiv:1705.02680*.
6. Arica, N., & Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *31*(2), 216–233.
7. Attigeri, S. (2018). Neural network based handwritten character recognition system. *International Journal of Engineering and Computer Science*, *7*(03), 23761–23768.
8. Boufenar, C., Kerboua, A., & Batouche, M. (2018). Investigation on deep learning for off-line handwritten Arabic character recognition. *Cognitive Systems Research*, *50*, 180–195.
9. Cilia, N. D., De Stefano, C., Fontanella, F., & di Freca, A. S. (2019). A ranking-based feature selection approach for handwritten character recognition. *Pattern Recognition Letters*, *121*, 77–86.
10. Decerbo, M., MacRostie, E., & Natarajan, P. (2004). The bbn byblos pashto ocr system. *Proceedings of the 1st ACM Workshop on Hardcopy Document Processing*, 29–32.
11. Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, *29*(6), 141–142.
12. Dongre, V. J., & Mankar, V. H. (2011). A review of research on Devnagari character recognition. *ArXiv Preprint ArXiv:1101.2491*.
13. Draman, N. A., Wilson, C., & Ling, S. (2009). Bio-inspired audio content-based retrieval framework (B-ACRF). *World Academy of Science, Engineering and Technology*, *29*(5), 785–790.
14. El-Sawy, A., Hazem, E.-B., & Loey, M. (2016). CNN for handwritten arabic digits recognition based on LeNet-5. *International Conference on Advanced Intelligent Systems and Informatics*, 566–575.
15. El-Sawy, A., Loey, M., & El-Bakry, H. (2017). Arabic handwritten characters recognition using convolutional neural network. *WSEAS Transactions on Computer Research*, *5*, 11–19.
16. Erhan, D., Courville, A., Bengio, Y., & Vincent, P. (2010). Why does unsupervised pre-training help deep learning? *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 201–208.
17. Faraoun, K. M., & Boukelif, A. (2006). Artificial Immune Systems for text-dependent speaker recognition. *INFOCOMP Journal of Computer Science*, *5*(4), 19–26.
18. Fischer, S. R. (2004). *History of Writing (pp. 34-40)*. London: Reaktion Books.
19. Fujisawa, H. (2008). Forty years of research in character and document recognition—an industrial perspective. *Pattern Recognition*, *41*(8), 2435–2446.
20. Gan, J., Wang, W., & Lu, K. (2019). A new perspective: Recognizing online handwritten Chinese characters via 1-dimensional CNN. *Information Sciences*, *478*, 375–390.
21. Ganga Gowri, B., & Soman, K. P. (2018). Enhancement of white Gaussian noise affected speech using VMD-ℓ 1 trend filter method. *Journal of Intelligent & Fuzzy Systems*, *34*(3), 1701–1711.
22. Gaur, A. (1992). *A history of writing*. British library London, UK.

23. Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, *160*(1), 106.
24. Jangid, M., & Srivastava, S. (2018). Handwritten devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods. *Journal of Imaging*, *4*(2), 41.
25. Joshi, D. S., & Risodkar, Y. R. (2018). Deep learning based Gujarati handwritten character recognition. *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, 563–566.
26. Kavitha, B. R., & Srimathi, C. (2019). Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks. *Journal of King Saud University-Computer and Information Sciences*.
27. Kobetski, M., & Sullivan, J. (2013). Apprenticeship learning: transfer of knowledge via dataset augmentation. *Scandinavian Conference on Image Analysis*, 432–443.
28. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.
29. Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring strategies for training deep neural networks. *Journal of Machine Learning Research*, *10*(1).
30. LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 396–404.
31. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.
32. Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning*, 609–616.
33. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *ICML*.
34. Pant, A. K., Panday, S. P., & Joshi, S. R. (2012). Off-line Nepali handwritten character recognition using Multilayer Perceptron and Radial Basis Function neural networks. *2012 Third Asian Himalayas International Conference on Internet*, 1–5.
35. Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 63–84.
36. Ptucha, R., Such, F. P., Pillai, S., Brockler, F., Singh, V., & Hutkowski, P. (2019). Intelligent character recognition using fully convolutional neural networks. *Pattern Recognition*, *88*, 604–613.
37. Quiles, M. G., & Romero, R. A. F. (2005). A computer vision system based on multi-layer perceptrons for controlling mobile robots. *ABCM Symposium Series in Mechatronics*, *2*, 661–668.
38. Roy, S., Das, N., Kundu, M., & Nasipuri, M. (2017). Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach. *Pattern Recognition Letters*, *90*, 15–21.
39. Ruck, D. W., Rogers, S. K., & Kabrisky, M. (1990). Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, *2*(2), 40–48.
40. Saha, S., & Saha, N. (2018). A Lightning fast approach to classify Bangla Handwritten Characters and Numerals using newly structured Deep Neural Network. *Procedia Computer Science*, *132*, 1760–1770.
41. Sainath, T. N., Mohamed, A., Kingsbury, B., & Ramabhadran, B. (2013). Deep convolutional neural networks for LVCSR. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 8614–8618.
42. Sen, S., Shaoo, D., Paul, S., Sarkar, R., & Roy, K. (2018). Online Handwritten Bangla Character Recognition Using CNN: A Deep Learning Approach. In *Intelligent Engineering Informatics* (pp. 413–420). Springer.
43. Sinha, R. M. K. (2009). A journey from Indian scripts processing to Indian language processing. *IEEE Annals of the History of Computing*, *31*(1), 8–31.
44. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.
45. Steinherz, T., Rivlin, E., & Intrator, N. (1999). Offline cursive script word recognition–a survey. *International Journal on Document Analysis and Recognition*, *2*(2–3), 90–110.
46. Ukil, S., Ghosh, S., Obaidullah, S. M., Santosh, K. C., Roy, K., & Das, N. (2019). Improved word-level handwritten indic script identification by integrating small convolutional neural networks. *Neural Computing and Applications*, 1–16.
47. Weng, Y., & Xia, C. (2019). A new deep learning-based handwritten character recognition system on mobile computing devices. *Mobile Networks and Applications*, 1–10.
48. Yang, J.-B., Shen, K.-Q., Ong, C.-J., & Li, X.-P. (2009). Feature selection for MLP neural network: The use of random permutation of probabilistic outputs. *IEEE Transactions on Neural Networks*, *20*(12), 1911–1922.