

Offline Handwritten Hindi Character Recognition Using Deep Learning with Augmented Dataset



Ajay Indian, Karamjit Bhatia, and Krishan Kumar

1 Introduction

In the early recent years, tremendous developments have been observed in solving curious tasks by adopting Deep learning. Although neural networks have been utilizing by researchers for the last few decades, their real strength has only been unraveled in recent years. This advancement is because computer systems are well equipped with Graphics Processing Units (GPUs), enabling them to facilitate a superior computational power for improved learning [1]. The improvement and previously obtained great results have directed us to a fact where it is believed that DNNs are proficient enough to solve virtually any problem, provided a sufficient number of training samples are supplied. The main problem of concern in our study is to prepare a vast amount of data samples adequate to build an opposite model for the recognition of offline characters of Hindi script. In this paper, a handwritten dataset is prepared by collecting the samples of handwritten characters from various individuals. This dataset is then augmented by introducing some hypothetical variations in collected samples to enhance the variety of samples such that overall improvement in recognition is attained.

Extracting features has been a crucial stage of any pattern recognition system. This phase generally involves using domain-centric knowledge to extract the features from the image such that extracted features can be used to develop an efficient model. Therefore, one of the characteristics of such techniques that hamper generality is to rely on domain-specific features. To neutralize this phase, the use of Convolution Layers is proposed to extract the implicit features of character images instead of

A. Indian (✉)
Central University of Rajasthan, Ajmer, India
e-mail: ajay.indian@curaj.ac.in

K. Bhatia · K. Kumar
Gurukula Kangri Vishwavidyalaya, Haridwar, India

extracting some domain-specific features. These layers are capable of mining features from the character images supplied as input matrix, irrespective of their size, and pass on to the subsequent layer [1]. The models based on the concept of these layers are called Convolutional Neural Networks (CNNs). The best fascinating characteristic of these layers is that they are not known to extract a similar set of features. Rather, they evolve and learn over time.

Apart from these layers, the dataset adopted for the CNN model's training may also be limited to the level of variations in the shape of characters captured during collection [2]. For instance, in our study, it is described how an OCR system can be restricted to some samples when trained with inappropriate parameters using a restricted dataset. Thus, the proposed method is a combination of the two techniques. Convolution layers are employed to extract the features from the original dataset and augmented dataset of offline handwritten Hindi characters. Original dataset, augmented dataset, and the architecture of CNNs are elaborated in Sects. 3, 4 and 5.

2 Literature Review

A concise overview of the work done in the field of offline handwritten Hindi character recognition is presented in this section, including traditional method like Hidden Markov Models (HMMs), Support Vector Machines (SVMs), Decision Trees (DTs), Regular Expressions (REs), etc. to Deep Learning (DL) methods.

Siddhartha et al. [3] proposed the graph matching scheme to compute the similarity among the characters extracted from bank cheques with the samples of characters in the database to recognize the characters. The proposed method was able to identify the characters written on bank cheques with 87.2% accuracy. Gauri et al. [4] presented a hybrid feature extraction scheme and a feature selection technique based on a genetic algorithm with adaptive MLP classifiers to improve overall performance. Kamble et al. [5] proposed a technique for extracting features from handwritten offline Marathi characters using the features based on connected pixel-like area, perimeter, orientation, Euler number, and eccentricity. An accuracy of 85.88% was reported using the proposed technique with the k-NN classifier with fivefold validation. Indian and Bhatia [6] suggested a handwritten offline Hindi vowel "Swar" recognition method by employing a new wave-based feature extraction "Tarang" with back-propagation and attained the accuracy of 96.2%.

Anuvadiya et al. [7] stated that encouraging recognition rate can be achieved with CNN by considering some particular issues. This way, the CNN-based system for recognizing handwritten characters offers an improved recognition rate compared to other CNN-based recognition systems. Indian and Bhatia [8] projected a combinational feature vector using Gradient, Zernike moments, and wave-based features for Hindi numerals recognition and achieved 96.4% accuracy with Zernike complex using BPNN classifiers. Alom et al. [9] presented the experimental results. They revealed the higher performance of the Deep Convolutional Neural Network (DCCNN) model compared to other prevalent object recognition methods. It shows

that a DCNN can be a virtuous choice to develop an automatic system for handwritten Bangla character recognition for hands-on applications. Puri et al. [10] presented a proficient Devnagari character recognition method by utilizing SVM for handwritten and printed monolingual documents having Hindi, Sanskrit, and Marathi text. The experiments conducted on the projected system achieved moderate recognition accuracies of 98.35% handwritten characters. Recently, Indian and Bhatia [11] report the efficacy of Zernike moments and Zernike complex moments using a backpropagation neural network classifier for Hindi numeral recognition and accuracy of 80.8% and 94.8%, respectively, is achieved.

It is observed that neural networks, particularly CNN, are being utilized widely to recognize handwritten and printed characters. Yet, traditional methods, e.g., HMM, SVM, DT, etc. are also being adopted in combination with CNN. Memon et al. [12] presented a systematic literature review on OCR and confirmed the same observations.

3 Model Description

An artificial neural network can be best described by its underlying topology, which can be described with the help of different layers utilized in its topology. The current section is dedicated to elaborate two essential layers used in the proposed topology.

3.1 Convolution Layers

The convolution layer is the first and fundamental layer utilized in the proposed scheme. Convolutional layers are responsible for carrying out inference precisely; these layers work as a pre-processing tool with a key objective to extract features from a given input. This task is carried out by employing convolution kernels and convolution operations.

Each convolution layer is equipped with some fixed number of filters, also called kernels. When these layers are supplied with the input, all filters are convoluted for the given input, and each filter produces a single output. All convolution layers possess a crucial characteristic that if more kernels are applied for an input, which is generally not done in deep learning, then extracted features gradually become more abstract. It is critically essentials it supports the model to attain generalization. Moreover, these layers can be further improved to increase their competency and correctness.

3.2 Fully Connected Layers

Fully connected layers are other essential layers in the proposed model. Convolution layers are responsible for extracting specific significant features from supplied input data. The features received from the convolution layer are supplied to a fully connected layer, which produces the results. These layers of CNN are also used in the traditional neural network.

The convolution layer produces the 2D form output, but fully connected layers require input in the 1D form, preferably. Therefore, the output produced from the previous layer is first transformed into a 1D form. So, after converting the output into 1D, it is passed to the fully connected layer. Each of the output values is considered as distinct features and used to characterize the image. These layers carry out two transformations on the received data—a linear and a non-linear transformation. First, a linear transformation is done on the input data.

$$Z = W^T \cdot X + b, \tag{1}$$

where X denotes the input data vector, W is the randomly initialized weight matrix, and b signifies a constant bias value (Fig. 1).

At this moment, only non-linear transformation is left in the forward propagation. Merely performing linear transformation cannot acquire complex relationships hidden in data. Therefore, a component is introduced in the architecture, which enhances non-linearity in the data, known as the activation function. The activation functions support the network to utilize vital information and suppress inappropriate data [14]. Activation functions are used at every layer of the network. The specific activation function to be adopted for a problem depends on the problem’s complexity being solved. As our problem is multi-class, so ReLU (Rectified Linear Unit) is used to solve the problem, which is a non-linear activation function widely used in deep learning. The key benefit of adopting the ReLU function as activation functions is that it does not activate all neurons simultaneously, specifically if the output of the linear transformation < 0.

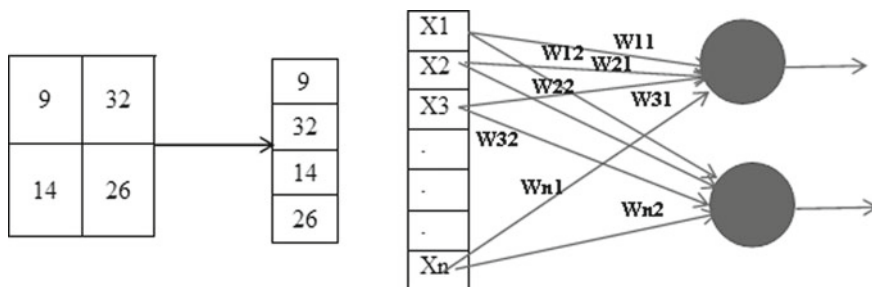


Fig. 1 Depicting the working of fully connected layer

$$f(x) = \max(0, x) \quad (2)$$

If input values are negative, it will return zero that means corresponding neurons will not be activated [14]. ReLU will activate the selected number of neurons. The ReLU function is very efficient in computation compared with other functions (such as sigmoid and tanh).

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3)$$

4 Dataset Description

Our study has used two different datasets- one is collected samples of 46 handwritten characters (10 numerals, 36 consonants) from different individuals of varying groups initially. Two hundred samples were collected for each character, making the complete dataset of 9200 samples (ORG). The second one is the augmented dataset (AUG) derived from the original dataset. With the help of randomized data augmentation, one can grow the size and variations in the training data. Distortions to the image data can be made invariant with the use of augmentation. For instance, randomized rotations can be added to input images such that a network is made invariant to the existing rotation found in input images. An augmented dataset is a handy technique to apply a fixed number of augmentations to 2D images to carry out classification tasks.

For conducting the experiments, the original dataset (ORG) was utilized to construct the model. However, the core competence of this dataset lies under the augmented dataset for the sake of generalization. In the process of augmentation, few image samples of the original dataset are randomly augmented. This augmented dataset is then used for the construction of the classifier. These augmentations comprise

- Random rotation of images in a direction with random value ranges from 10 to 10 and 15 to 15.
- Random translation of images in the horizontal direction.
- Random translation of images in the vertical direction.

As mentioned above, the augmentations are the key to enriching the original dataset with many image variations, which helps defend the key argument of this study to obtain an efficient model for handwritten Hindi character recognition utilizing this augmented dataset. Similar approaches for creating an artificial dataset for a specific problem have also been effectively adopted in other areas, such as scene-text recognition [13], where an appropriate number of data samples were not readily available. For both datasets, 85% of samples were utilized for training, and

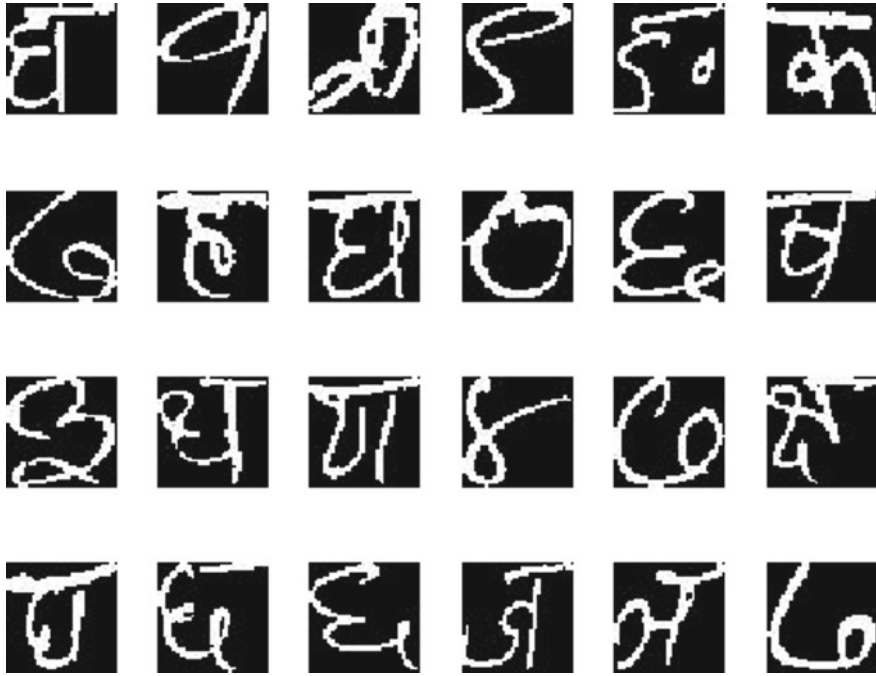


Fig. 2 Random samples of consonants and numerals of hindi script

the rest 15% of samples were utilized to validate the model. Hence, the efficacy of the original dataset over the augmented dataset was verified using the experiments. In this paper, a discussion on the model developed using the original dataset followed by a discussion on how augmented dataset enhances the model's performance over the original dataset is presented. In the validation phase, 15% of each dataset's samples were used to assess the models' performances for handwritten character recognition. Some random samples of characters from the prepared dataset can be found in Fig. 2.

5 Implementation

The proposed model based on CNNs for recognizing handwritten Hindi characters is implemented using MATLAB 2019a, and the model is trained and assessed on the above-stated datasets. The preceding sections cover the description of the layers adopted in the proposed model. First, several convolution layers are employed to extract the features of character images. These 2D features are then transformed into a linear vector and then supplied to the fully connected layer. At last, a fully connected layer reduces the size of the linear vector corresponding to the number of target class labels (in our case, 46 class labels for 36 consonants and 10 numerals).

Four strategies are formed using the original dataset and augmented dataset to assess the efficacy of the proposed model:

Strategy-I: CNN models without a dropout layer on original data.

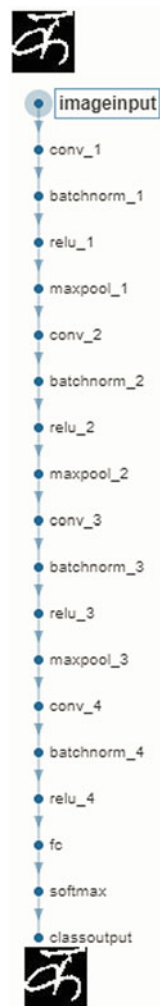
Strategy-II: CNN models with a dropout layer on original data.

Strategy-III: CNN models without a dropout layer on augmented data.

Strategy-IV: CNN models with a dropout layer on augmented data.

An outline of the network layers adopted in the proposed model (*Dropout Layer are only adopted in Strategy-II and Strategy-IV) is presented in Table 1, and the overall flow of the data in the model is depicted in Fig. 3.

Fig. 3 The flow of the data in the proposed CNN model



Another vibrant characteristic of this neural network model is that layers are facilitated to attain maximum generalization and performance gain. Following are the layers:

5.1 Dropout Layers

Neural networks based on deep learning paradigm can quickly over-fit with a training dataset having a small number of examples. Ensembles of network models with different configuration parameters are popularly acknowledged to overcome overfitting problems up to an extent. Still, they incur the extra computational cost to train and maintain multiple models. Preferably, a single model can be utilized to simulate with a huge number of various network structures by randomly dropping out the nodes while training the model. Which is known as dropout, which incurs less computational cost with a remarkable efficient regularization technique to reduce overfitting and improve generalization error in deep neural networks [15, 16].

5.2 Pooling Layers

Next to the convolutional layer, a novel-pooling layer is added to the network. Precisely, after a non-linear activation function (e.g., ReLU) is applied to the feature maps received from the convolutional layer. Adding a pooling layer next to the convolutional layer is a very general phenomenon adopted for sequencing layers in the CNN model, which can be used one or more than one time in a given model. These layers work upon all feature maps individually to produce a new set of pooled feature maps. Pooling comprises selecting a pooling operation similar to a filter to be applied to feature maps. The pooling or filter size remains smaller than the size of the feature maps [18]. The pooling is specified rather than learned. Generally, pooling is done by employing two functions:

- *Average Pooling*: Computes the average value for all the patches on the feature map.
- *Maximum Pooling*: Computes the max value for all the patches on the feature maps, called max-pooling [18].

5.3 Batch Normalization Layer

This layer is used to normalize each input channel across a mini-batch. These layers first normalize all channels' activations by subtracting the mini-batch mean and dividing by the mini-batch standard deviation. Small batches of mini sizes are adopted throughout the training phase to speed up the training of CNNs and reduce the

sensitivity to network initialization. Challenges that come with small size batches are resolved using the covariance shift [17, 19] and batch normalization.

6 Result and Discussion

Based on the strategies mentioned above, all the models, I, through IV, are trained with a hyper-parameter, *tolerance*, and dynamically defined tolerance—all the models assessed using the validation dataset by using the estimated their accuracies and losses. If the loss decreases after some no. of epochs, concluded that convergence had occurred; hence, training was halted. Here, the maximum number of epochs was decided based on the performance. If the performance of the model stabilizes after some number of epochs, training was stopped. Empirically observation has been made that all models based on the strategies mentioned above converged comparatively better with small datasets than some other state of the art where large datasets were used. A summary of all the outcomes obtained after experimentation is presented in Table 2.

Based on the original dataset (ORG), strategy-I achieved the recognition rate of 90.78% with an average training time/epoch of 13.7 s and a total training time of 6 min 51 s. Whereas the Strategy-II based on the original dataset (ORG) with dropout layer performed better, giving an accuracy of 91.90%, an average training time/epoch of 12.31 s, and total training time of 7 min 11 s that shows that for dataset (ORG) network generalizes well with dropout layer.

Strategy-III, based on the augmented dataset (AUG), achieved the recognition rate of 92.18% with an average training time/epoch of 12.47 s and a total training time of 9 min 34 s. Whereas, the Strategy-IV based on the augmented dataset (AUG) with the dropout layer performed better as compared to Strategy-I, Strategy-II, and Strategy-III. It achieved an accuracy of 94.19% with an average training time/epoch of 8.43 s and a total training time of 6 min 38 s, which is the highest accuracy achieved in the proposed work. Performance of the Strategy-IV affirms that the network generalizes very well with the dropout layer and improves the model's performance in terms of speed and accuracy. Performance and loss estimation of the Strategy-IV is shown in Figs. 4 and 5 while training the model. Performance comparison of the proposed strategy with other existing models based on CNN are presented in Table 3, which shows that the performance of the proposed approach (94.19%) is comparable to the performance (95.46%) of [21] as in the proposed work dataset is smaller, and numbers of classes are more than double. Performance of the strategies discussed in [20, 22] are much better than the proposed approach as the numbers of samples in these strategies are ten times more, but the average training time/ epoch is very high. It shows that a higher accuracy rate can be achieved by increasing the dataset's size in the proposed strategy.

Table 1 Outline of the network layers adopted with various parameters

Layers	Parameters used
Input layer	40*40
Convolutional	4 Filters of size 3*3, Stride of [1 1], with Padding "same"
Normalization	Batch
Activation function	ReLU
Pooling	2*2 Max-pooling, Stride [2 2], with Padding [0 0 0 0]
*Dropout	0.5
Convolutional	8 Filters of size 3*3, Stride of [1 1], with Padding "same"
Normalization	Batch
Activation function	ReLU
Pooling	2*2 Max-pooling, Stride [2 2], with Padding [0 0 0 0]
Convolutional	16 Filters of size 3*3, Stride of [1 1], with Padding "same"
Normalization	Batch
Activation function	ReLU
Pooling	2*2 Max-pooling, Stride [2 2], with Padding [0 0 0 0]
Convolutional	32 Filters of size 3*3, Stride of [1 1], with Padding "same"
Normalization	Batch
Activation function	ReLU
Fully connected	46 Fully connected layers
Softmax	Softmax
Classification	Crossentropyex

7 Conclusion

The work done in this paper is limited to study the performance of augmented dataset (AUG) for Hindi characters over the original dataset (ORG) of small size using a CNN with a dropout layer for the recognition of characters. The current study has shown that the augmented dataset (AUG) outperforms the original dataset, signifying the augmented dataset's importance. Using the dropout layer also enables the model to overcome over-fitting as the dataset used in this paper is comparatively small compared to other states of the art problem. Hence, the use of a dropout layer increases performance. In the future, this technique can be enhanced for the recognition of words as this study only focused on recognizing individual characters of Hindi script and some other scripts.

Table 2 Performances of the proposed strategies

Strategy	Dataset	Epoch number with best accuracy	Average training time/epoch (s)	Total training time	Accuracy (%)
Strategy-I	ORG (Collected Samples)	30	13.7	6 min 51 s	90.78
Strategy-II	ORG (Collected Samples)	35	12.31	7 min 11 s	91.90
Strategy-III	AUG (Augmented Collected Samples)	46	12.47	9 min 34 s	92.18
Strategy-IV	AUG (Augmented Collected Samples)	46	8.43	6 min 38 s	94.19

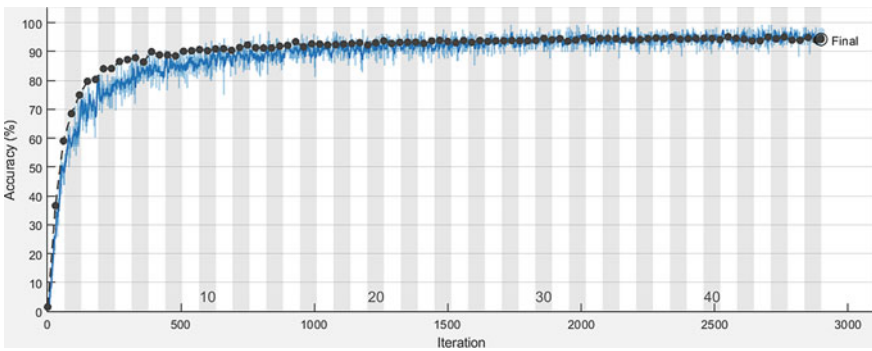


Fig. 4 Performance of the Strategy-IV in terms of accuracy versus iteration

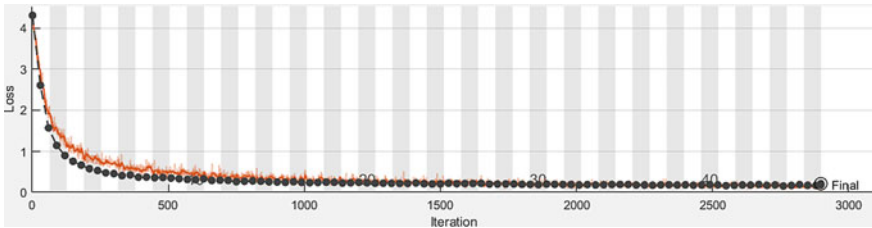


Fig. 5 Performance of the Strategy-IV in terms of loss versus iteration

Table 3 Performance comparison of strategies adopted in study

Authors	Dataset	No. of samples in the dataset	No. of distinct characters/class labels	Average training time/epoch	Accuracy (%)
Baranidharan et al. [20]	UC irvine machine learning repository	92,000	46	47 s	98.66
Aneja and Aneja [21]	UC irvine machine learning repository	92,000	46	16.3 min	99
Sonawane et al. [22]	Own created dataset	16,870	22	–	95.46
Proposed	AUG dataset	9200	46	8.43 s	94.19

Further, the offline handwritten character recognition system can be integrated with the security system to develop a more interactive and more secure system in this digital era. The user will be able to interact with the system and get authenticated by using his handwritten text. Hence, the system will be more personalized and secured for the individual users.

References

1. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
2. K. Peymani, M. Soryani, From machine-generated to handwritten character recognition: a deep learning approach, in *3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, Shahrekord, pp. 243–247. <https://doi.org/10.1109/PRIA.2017.7983055> (2017)
3. S. Banerjee, B.R. Ghosh, A. Kundu, Handwritten character recognition from bank cheque. *Int. J. Comput. Sci. Eng.* EISSN: 2347–2693, **4**(1), 099–104 (2016)
4. G. Katiyar, S. Mehruz, A hybrid recognition system for off-line handwritten characters. *Springer Plus* **5**(1), 1–18 (2016)
5. P.M. Kamble, R.S. Hegadi, Geometrical features extraction and KNN based classification of handwritten marathi characters, in *World Congress on Computing and Communication Technologies (WCCCT)*, pp. 219–222 (2017)
6. A. Indian, K. Bhatia, Offline handwritten hindi ‘SWARs’ recognition using a novel wave based feature extraction method. *Int. J. Comput. Sci. Issues* **14**(4), 8–14 (2017)
7. H.D. Anuvadiya, A.A. Abhangi, A research on improve handwritten character recognition by using convolutional neural network. *Int. J. Adv. Eng. Res. Dev.* **5**(05), 20–25 (2018)
8. A. Indian, K. Bhatia, Off-line handwritten hindi consonants recognition system using zernike moments and genetic algorithm, in *Proceedings of the IEEE Conference International Conference on System Modeling & Advancement in Research Trends, (SMART-2018)*, pp. 10–16 (2018)
9. Z. Alom, P. Sidike, M. Hasan, T.M. Taha, V.K. Asari, Handwritten bangla character recognition using the state-of-the-art deep convolutional neural networks. *Hindawi Computat. Intell. Neurosci.* **2018**, Article ID 6747098, pp 1–13. <https://doi.org/10.1155/2018/6747098> (2018)

10. S. Puria, S.P. Singh, An efficient devnagari character classification in printed and handwritten documents using SVM, in *International Conference on Pervasive Computing Advances and Applications—PerCAA 2019, Procedia Computer Science*, vol. 152, pp. 111–121 (2019)
11. A. Indian, K. Bhatia, Offline handwritten hindi numerals recognition using zernike moments. *Int. J. Tomograp. Simul.* **32**(2), 68–82 (2019)
12. J. Memon, M. Sami, R.A. Khan, Handwritten Optical Character Recognition (OCR): a comprehensive Systematic Literature Review (SLR), Cornell University (2020)
13. M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Synthetic data and artificial neural networks for natural scene text recognition (2014)
14. C.E. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation functions: comparison of trends in practice and research for deep learning. [arXiv:1811.03378v1](https://arxiv.org/abs/1811.03378v1) [cs.LG] 8.(2018)
15. H. Wu, X. Gu, Max-pooling dropout for regularization of convolutional neural networks. (2015)
16. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014)
17. I. Sergey, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift. [arXiv:1502.03167](https://arxiv.org/abs/1502.03167) (2015)
18. J. Brownlee, A gentle introduction to pooling layers for convolutional neural networks, in *A Web Article on Deep Learning for Computer Vision*, April 22 (2019)
19. <https://in.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.batchnormalizationlayer.htm>
20. B. Baranidharan, A. Kandpal, A. Chakravorty, Hindi handwritten character recognition using CNN. *Int. J. Adv. Sci. Technol.* **29**(6), 58–66 (2020)
21. P. K. Sonawane, S. Shelke, Handwritten devanagari character classification using deep learning, in *IEEE International Conference on Information, Communication, Engineering and Technology (ICICET)*, Zeal College of Engineering and Research, Narhe, Pune, India, pp. 1–4, Aug. 29–30 (2018)
22. N. Aneja, S. Aneja, Transfer learning using CNN for handwritten devanagari character recognition, in *Accepted for publication in IEEE International Conference on Advances in Information Technology (ICAIT)* (2019)