

Performance Evaluation of Deep Learning Networks on Printed Odia Characters

¹Sanjibani Sudha Pattanayak, ¹Sateesh Kumar Pradhan and ²Ramesh Chandra Malik

¹Department of Computer Science and Application, Utkal University, Bhubaneswar, India

²P.G. Department of Odia Language and Literature, Utkal University, Bhubaneswar, India

Article history

Received: 25-04-2020

Revised: 05-07-2020

Accepted: 21-07-2020

Corresponding Author:

Sanjibani Sudha Pattanayak
Department of Computer
Science and Application, Utkal
University, Bhubaneswar, India
Email: sanjibani@gmail.com

Abstract: Deep machine learning includes a series of layers to mimic the working of the human brain for taking a decision. Deep learning networks have shown good results in character recognition in the past. This paper evaluates the performance of different deep learning networks like Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) based recurrent neural network and Convolutional LSTM on printed Odia characters. The Odia character database contains more than 24,000 images of printed Odia characters (including simple as well as complex characters) out of which 23,857 nos. of images are chosen for this work. Besides these three, a nested Convolutional neural network model is developed for different categories of printed character image groups which are formed based on their writing style. Here, in this study, the nested model is showing the best results in terms of error rate, accuracy and no. of epochs in comparison to the other three. Different pre-processing steps like binarization, size-normalization, blurring, interpolation, etc. are involved before passing the images to the deep neural networks to increase the recognition accuracy.

Keywords: Convolutional Neural Network, Long Short-Term Memory, Convolutional LSTM, Character Recognition, Odia Character Database

Introduction

Both the printed and handwritten character recognition field has received good attention from the researchers for the last few decades and successfully many tools are developed for different languages like English, Chinese, etc. But for some Indic languages like Odia, it is yet to achieve some noticeable milestone though for decades there are several pieces of papers on the same. There are thousands of priceless and magnificent books written in Odia language which are yet to be digitized. Due to the large size of the character set (classes), font variance and the complexity of the characters as well as the documents, it is found difficult to achieve some good result for character recognition of Odia language. It is shown that in the past, different deep learning networks have successfully tested for character recognition which motivates to experiment it for the Odia characters.

A Deep Belief Network is used for handwritten Bangla character recognition (10 numerals and 50 basic characters) (Sazal *et al.*, 2014). This Network works in two different phases; one unsupervised learning phase using a contrastive divergence algorithm followed by a supervised learning phase that uses a gradient-based

backpropagation method. Using Convolutional Neural Network, authors (Maitra *et al.*, 2015) have experimented with the recognition of handwritten English, Bangla, Odia and Telugu numerals and Bangla characters. With SVM as the classifier at the last level, good accuracy result has been revealed for most of the database. Convolutional Neural Network is also preferred for handwritten Devanagari character recognition with different network architecture (Acharya *et al.*, 2015). A CNN based automatic vehicle license plate recognition system is developed which shows good results after a series of pre-processing (Selmi *et al.*, 2017). Oyedotun *et al.* (2015) have done a thorough study and classification of different deep learning architecture which includes Deep Belief Network (DBN), Deep Boltzmann Machine (DBM), Autoencoders, Conditional Random Fields, Hybrid architecture, etc.,. Here the efficiency of the neural network against some common variance in the character image like scaling, rotation, translation has been investigated. The experiment is done on handwritten Yoruba characters in Nigeria. For sentiment analysis, an efficient deep learning model is used which utilizes both CNN and bidirectional RNN (Hassan and Mahmood, 2017). Here CNN is used as the first layer, then the

pooling layer is substituted with the Bidirectional RNN layer to maintain the long-term dependencies and shows good results. The performance of bidirectional LSTM for printed Latin and German Fraktur characters is evaluated against the ground truth (Breuel *et al.*, 2013). Here connected component, baseline and x-height information are gathered for text-line normalization. An open-source form of LSTM and line normalization is available as OCRopus for testing, evaluating and reusing the method.

Convolutional Neural Network (CNN) and Deep Belief Network (DBN) are compared on the handwritten numeral database MNIST as well as the real world handwritten data (Wu and Chen, 2015). As per the study, CNN shows better results than DBN in case of image recognition. For handwritten Persian word recognition, CNN and RNN along with the segmentation tool Connectionist Temporal Classification (CTC) are experimented and show good results (Safarzadeh and Jafarzadeh, 2020). Binary LSTM based deep network is used for English (Breuel, 2017) and Odia text recognition (Ray *et al.*, 2015). In both cases, connectionist temporal classification is used for ground truth realization. LSTM helps to reduce the error level to a minimum.

Methods

Convolutional Neural Network

The basic objective of a convolutional neural network is to view and perceive the object as the humans do. Unlike the traditional neural network, here the knowledge extraction is implicit. So it is also known as a deep learning network. This network seeks less amount of pre-processing. A series of kernels or filters are used to extract the features from the input image and form the convolutional layer of the network. Then the features will be down sampled by following any of the pooling layers; max pooling, sum pooling, average pooling, etc. The network may contain a series of convolutional and pooling layer followed by fully connected layers. Then different types of activation function; tanh, sigmoid, or softmax can be used to classify the features into different categories. CNN has been successfully applied for many advanced fields of computations including character recognition. The architecture of the convolutional neural network that is adopted here in this study is shown below in the Fig. 1.

Recurrent Neural Network

The human brain's understanding system is quite persistent. To understand something, instead of using only the current input, the brain also associates the past understanding with it. E.g., whenever we are reading a book, at a particular moment, previous context understanding helps to perceive the intended meaning of the current text. One of the major drawbacks of the

traditional neural network is it fails to mimic this specific idea. So, Recurrent neural network has been developed to solve this issue.

In the block diagram of Recurrent neural network Fig. 2, NN is a single or a series of neural network which receives input and generates output. The loop indicates that the same thing is repeated several times. In each phase, RNN is passing some output to the next phase. RNN has been successfully implemented for different fields like image recognition, speech recognition, language processing, etc.

Long-Short Term Memory network (LSTM) is a special type of or very popular recurrent neural network. It remembers the past information for a longer period. Each unit of the LSTM considers three different inputs:

I_i : Current input
 O_{i-1} : Previous output
 M_{i-1} : Previous memory

and generates two outputs:

O_i : Current output
 M_i : Modified Memory which will become the input for the next unit

The internal operations of the neural network decide whether to remember or forget the memory and how to merge the old memory with the new memory. This is possible with the help of a memory cell and three multiplicative gates known as input, output and forget. The input gate reads new information for the model, the output gate passes it to the next phase of the model and it retains the information as long as the forget gate is switched on (Hochreiter and Schmidhuber, 1997). This is demonstrated here in this Fig. 3.

Convolutional LSTM

LSTM is one of the most powerful tools for handling large range dependencies but it involves lots of redundancy (Xingjian *et al.*, 2015). So to eliminate this problem, the convolutional structure is introduced in LSTM. In Convolutional LSTM, the input cells, output cells including input gate, output gate and forget gate of the hidden state all become 3D tensors where two dimensions refer to rows and column of a spatial grid. It decides the future with the help of the current input and history of the neighbour. It can be achieved by introducing the convolutional operator in place of matrix multiplication between state-to-state and input-to-state transition (Xingjian *et al.*, 2015). The architecture of convolutional LSTM is suitable for video sequence recognition, but we have trained it here for static images.

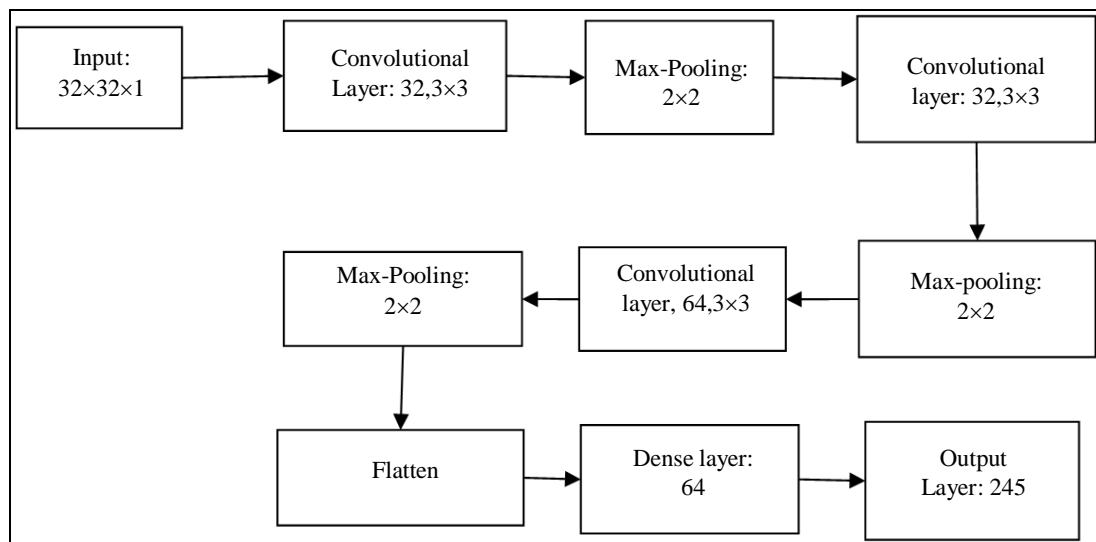


Fig. 1: Modelled convolutional neural network

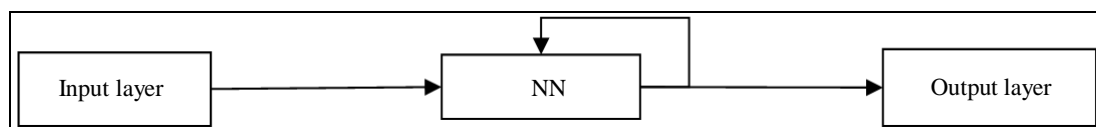


Fig. 2: RNN model

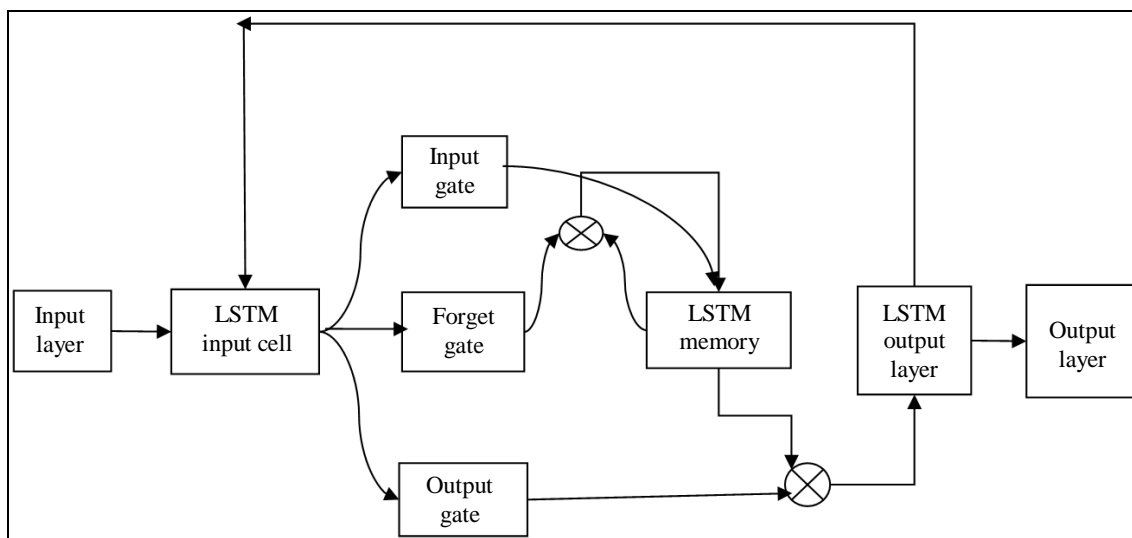


Fig. 3: Working principle of LSTM

Dataset

The database for the printed Odia characters that have been developed in (Pattanayak *et al.*, 2020) is used here for the experiment. It contains a total of 245 sub-folders for 245 different categories of Odia character symbols. These symbols include all the 10 digits and

basic as well complex characters. Out of 245 different symbols, 10 symbols are for digits, 47 symbols are of basic/simple characters and remaining 188 symbols are of complex Odia characters. Each sub-folder contains around 100 nos. of character images. So there are around 24,000 nos. of symbols in the database out of which 23,857 nos. of symbols are chosen. The chosen dataset is

then grouped into the training set and the testing set each having 245 nos. of subfolders. There are 17,459 nos. of images chosen for the training purpose, so kept in training subfolders and remaining 6398 nos. of images are kept in testing subfolders. The images in the dataset are pre-processed to size normalization of 32×32 . Binarization and tight boundary box creations were also a part of database creation. Here in this study, before passing the image data to a network, Gaussian blur and image interpolation is applied for smoothing the edge and increasing the resolution. Pre-processing refers to the transformation of the raw image data that is to be fed to the deep-learning models. The experiment shows that with pre-processed data, the models give better classification results than the raw data.

Experiments and Results Discussion

To evaluate the performance, a Convolutional Neural Network (CNN), Long-Short-Term Memory (LSTM) and the convolutional LSTM are considered. It has been implemented using TensorFlow-keras. The CNN model consists of a sequence of three nos. of convolutional layers followed by max-pooling layers and one fully connected layer. For the training purpose, the input layer contains 17,459 nos. of images of size 32×32 . For the first two convolutional layers, the size of the feature map is 32 and a 3×3 kernel is chosen to filter the images. For the third convolutional layer, the feature map size is 64. Then the feature maps generated by the convolutional layers are reduced/subsampled by the pooling layer choosing the maximum/largest value from a 2×2 dimension. So each time, the feature space is reduced by half. After the third convolutional layer, it is flattened and a fully connected layer of size 64 nodes is created. The output layer contains a total of 245 nos. of nodes. The training is performed with mini-batches of size 200. The model is repeated for 160 epochs.

The LSTM network architecture consists of two LSTM hidden layers followed by two fully connected layers. The vector form of the interpolated image (32×32) is passed as the input. So the input shape size is $17459 \times 32 \times 32$. The filter sizes of the two LSTM layers are 32 and 64 respectively. The dense layers consist of 128 nos. of nodes. Because of the network complexity, the time required for each epoch becomes more here. The last layer uses the softmax activation function.

In the case of convolutional LSTM, the data format used is (32,32,1) where 32×32 is the image size and 1 refers to the black and white channel. There are

17,459 nos. of samples as input and for the static images, the time step is fixed to 1. This is how the convolutional LSTM uses 5-D tensors. Here the convolutional LSTM consists of four hidden layers having filter size 32 and kernel size (3×3). Each Convolutional LSTM layer is followed by batch normalization and a max-pooling layer. The last layer is a fully connected layer that uses the softmax activation function for classification.

During the experiment, it is found that out of different adaptive learning rate methods like RMSProp, Adam and Adagrad optimizer, Adam optimizer is working better for all the above models. To avoid overfitting, after each layer, a 20% dropout is introduced. The above architectures are chosen after several experiments for the best result.

For the nested CNN model, all the 245 types of character symbols are grouped into 17 classes based on their writing style. Characters belong to a particular class appear similar. This is shown here in the Fig. 4. The convolutional neural network is applied here in a nested fashion. The outer CNN model is designed to classify the character symbols into 17 groups. In the second phase, 17 different inner CNN models have been designed for intra-class classifications of the symbols. Here training and testing time is pretty faster in comparison to other deep learning models.

For evaluation purposes, the categorical accuracy and loss of the outer CNN model are determined. Then, for the inner CNN models, average categorical accuracy and loss are calculated. To compute the actual categorical accuracy and loss of the system, the following formulae are adopted.

$$\text{Actual_categorical_accuracy} = \text{categorical_acc_of_outer_model} \times \text{average_categorical_acc of inner models}$$

Similarly,

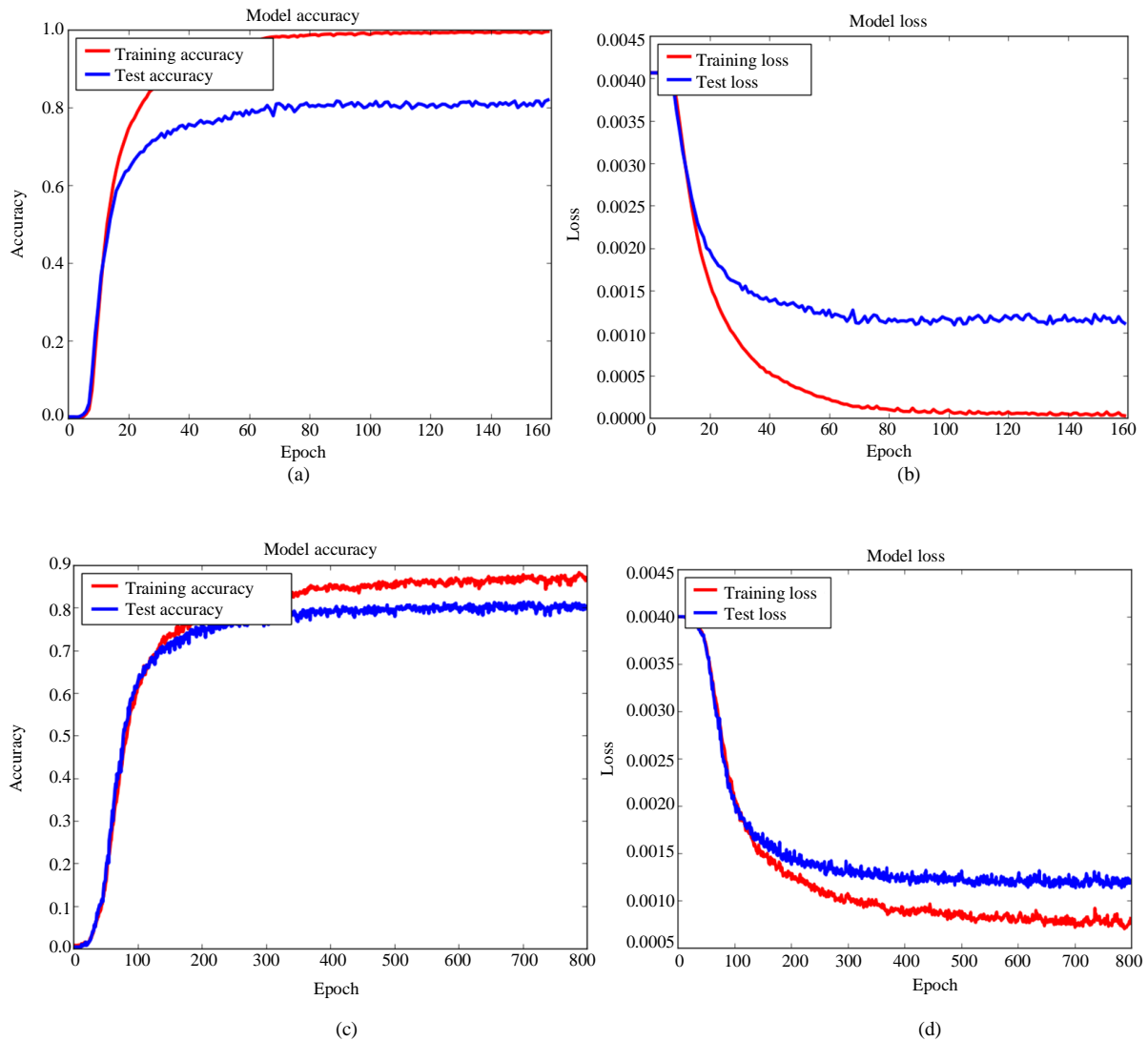
$$\text{Actual_loss} = \text{loss_rate of outer model} \times \text{average loss_rate of inner models}$$

Here, the categorical accuracy of the outer model is found to be 97.42% whereas the average categorical accuracy of the inner models is computed as 94.63%. Hence, the actual accuracy rate is 92.19%. Similarly, the actual loss is calculated to 0.018.

Analysis of the above models for the Odia characters shows that the nested CNN model gives the best result in terms of accuracy, error rate and no. of epochs required for convergence. Though the CNN model is run for 160 epochs, it converges in 100 epochs. LSTM takes around 800 epochs to converge. But the fast convergence is shown in the Convolutional LSTM model which is within 10 epochs. Similarly, for the nested CNN, all the models convergence within 20 nos. of epochs. For the given database, the CNN model gives an accuracy of 80.06%, the LSTM model shows 81.3% accuracy and the convolutional LSTM gives the best accuracy of 84.2%. whereas the nested CNN model is showing an accuracy of 92.17%. In all cases, the error rate is reduced to 0.1. Model accuracy and loss is presented here in the graphs shown in the Fig. 5.

Table 1: Class-wise accuracy rate for the nested model

Class	Accuracy rate
Class 1	97.01%
Class 2	100%
Class 3	90.24%
Class 4	98.14%
Class 5	96.12%
Class 6	87.5%
Class 7	96.36%
Class 8	88.86%
Class 9	97.29%
Class 10	98.41%
Class 11	92.53%
Class 12	88.04%
Class 13	91.50%
Class 14	96.42%
Class 15	96.15%
Class 16	99.27%
Class 17	95.03%



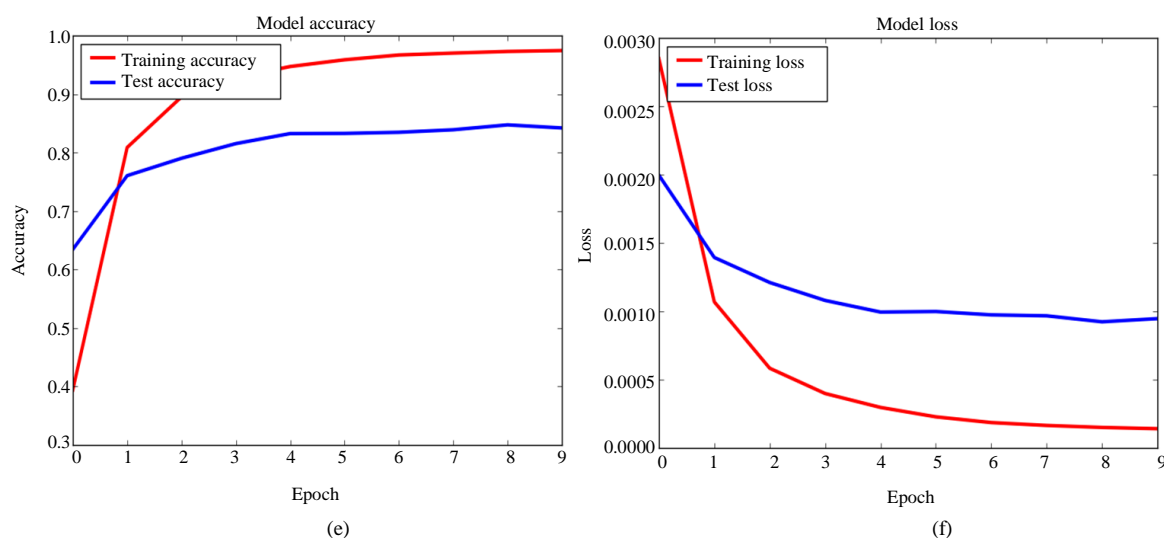


Fig. 5: Accuracy and loss measure of different networks (a) CNN model accuracy (b) CNN model loss (c) LSTM model accuracy (d) LSTM model loss (e) Convolutional LSTM model accuracy (f) Convolutional LSTM model loss

Conclusion

The objective of this work is to study the behaviour or performance of the different supervised deep learning networks for the printed Odia characters. Feature extraction is an implicit phase of the deep learning networks. Convolutional Neural Network uses a sequence of convolutional layers, pooling layers and fully connected layers for the classification. LSTM network employs a memory cell as well as the current input for the recognition process and the convolutional LSTM network introduces a convolutional structure in the LSTM to reduce the recurrence and introduce more sound spatial information. Nested models are built for different classes of the input symbols. Here in this observation, it is found that the nested CNN model gives a better recognition rate and converges in less no. of epochs. It gives an accuracy of 92.19%. For future enhancement, different unsupervised deep learning networks can be used.

Acknowledgement

All the authors contributed to conception, drafting and critical proof reading of this manuscript.

Author's Contributions

Sanjibani Sudha Pattanayak: Study, Analysis, Implementation and Drafting is done by her.

Sateesh Kumar Pradhan: He has contributed in designing the architecture of the deep neural network.

Ramesh Chandra Mallick: He helped in understanding the Odia language and script and suggested the methodology to prepare the dataset.

All authors discussed the results and contributed to the final manuscript.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

- Acharya, S., Pant, A. K., & Gyawali, P. K. (2015, December). Deep learning based large scale handwritten Devanagari character recognition. In 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 1-6). IEEE.
- Breuel, T. M. (2017, November). High performance text recognition using a hybrid convolutional-lstm implementation. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 1, pp. 11-16). IEEE.
- Breuel, T. M., Ul-Hasan, A., Al-Azawi, M. A., & Shafait, F. (2013, August). High-performance OCR for printed English and Fraktur using LSTM networks. In 2013 12th International Conference on Document Analysis and Recognition (pp. 683-687). IEEE.
- Hassan, A., & Mahmood, A. (2017, December). Efficient deep learning model for text classification based on recurrent and convolutional layers. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 1108-1113). IEEE.

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Maitra, D. S., Bhattacharya, U., & Parui, S. K. (2015, August). CNN based common approach to handwritten character recognition of multiple scripts. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 1021-1025). IEEE.
- Oyedotun, O. K., Olaniyi, E. O., & Khashman, A. (2015). Deep learning in character recognition considering pattern invariance constraints. *International Journal of Intelligent Systems and Applications*, 7(7), 1.
- Pattanayak, S. S., Pradhan, S. K., & Mallik, R. C. (2020). Printed Odia Symbols for Character Recognition: A Database Study. In *Advanced Computing and Intelligent Engineering* (pp. 297-307). Springer, Singapore.
- Ray, A., Rajeswar, S., & Chaudhury, S. (2015, January). Text recognition using deep BLSTM networks. In 2015 eighth international conference on advances in pattern recognition (ICAPR) (pp. 1-6). IEEE.
- Safarzadeh, V. M., & Jafarzadeh, P. (2020, January). Offline Persian Handwriting Recognition with CNN and RNN-CTC. In 2020 25th International Computer Conference, Computer Society of Iran (CSICC) (pp. 1-10). IEEE.
- Sazal, M. M. R., Biswas, S. K., Amin, M. F., & Murase, K. (2014, February). Bangla handwritten character recognition using deep belief network. In 2013 International Conference on Electrical Information and Communication Technology (EICT) (pp. 1-5). IEEE.
- Selmi, Z., Halima, M. B., & Alimi, A. M. (2017, November). Deep learning system for automatic license plate detection and recognition. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 1132-1138). IEEE.
- Wu, M., & Chen, L. (2015, November). Image recognition based on deep learning. In 2015 Chinese Automation Congress (CAC) (pp. 542-546). IEEE.
- Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation now casting. In *Advances in neural information processing systems* (pp. 802-810).