International Journal on Advances in Engineering Technology and Science
Volume: 5, Issue:1, 2024, e-ISSN: 2455-3131
DOI: 10.5281/zenodo.11186653

Performance Analysis of Modified Convolutional Neural Network Model for Recognition of MODI Character Set

Anshika Jain[1], Maya Ingle[2]

SCS & IT, D.A.V.V, Indore (MP)
[1]anshikajain002@gmail.com, [2]mayaingle22@gmail.com

***Abstract*— The "MODI Script" is an ancient script that originated in Maharashtra, Western India, and is widely used to create official documents. In the area of handwritten character recognition, a recognition system for MODI might be developed in order to interpret it better. Deep learning algorithms are used in character recognition applications to identify patterns accurately and efficiently. We proposed Modified CNN model for recognizing handwritten MODI character set. Our model comprises of five convolutions, two pooling, and one fully connected layer. Increasing the number of convolutional layers improves recognition accuracy by extracting more features. We trained our model with modified layers using 56-character set of MODI including numerals, vowels, and consonants, along with matras for each consonant. A total of 33,600 images have been used for the training and testing process. The results of the experiment indicate that the proposed model achieved a higher accuracy rate of 99.95%.**

Keywords: Character Recognition, MODI Character Set, Modified CNN Model

I. INTRODUCTION

A multilayer neural network learning technique of deep learning is currently an emerging technique in the computer vision and pattern recognition community. Convolutional Neural Network (CNN) is well-known model that has revolutionized pattern recognition specifically offline handwriting recognition. In recognition of offline handwritten MODI characters, deep learning-based ResNet50 and InceptionV3 algorithms play a vital role with Otsu binarization pre-processing technique attaining accuracy of 94.55% and 93.92% respectively [1]. For the recognition of MODI characters, using CNN autoencoder into CNN model and Support Vector Machine (SVM) feature classification method obtaining accuracy of 99.3% for recognition [2]. In recognition of handwritten MODI script, the Augmented CNN (ACNN) model is designed using On-the-fly augmentation method acquiring a recognition rate of 99.78% [3]. Using Transfer Learning technique, handwritten MODI characters are recognized with AlexNet feature extraction and SVM classification method achieving 92.32% recognition accuracy [4]. In order to recognize ancient MODI characters, using deep learning algorithms such as CNN and VGG16 securing character recognition accuracy of 91.12% and 97.62% respectively [5]. However, there is ample possibility to enhance the performance of the deep learning CNN model for MODI character recognition. Recognition of handwritten characters possesses significant acceptance in the arena of character recognition. The CNN model effectively recognizes characters present in the image as the existing deep learning CNN model is discussed in Section II. Section III has demonstrated the architectural details of the proposed Modified CNN model. The details of dataset, parameters, and evaluation process of the proposed model are deliberated in Section IV. Experimental analysis is explained in Section V along with the comparison of various cases. Finally, we summarize our research work with the conclusion in Section VI.

II. EXISTING CONVOLUTIONAL NEURAL NETWORK MODEL

A Convolutional Neural Network (CNN) is composed of an input and output layer, alongside multiple hidden layers. These hidden layers consist of alternating convolution layers, pooling layers, and fully connected layers drawn in Fig. 1. We discuss them briefly as follows:

Convolutional Layer

A convolutional layer combines linear (convolution) and non-linear (activation function) processes to extract features. The linear process combines with each smaller region in the input image weights and bias known as kernels or filters will generate feature maps. The hyper-parameters that determine the performance of the convolutional layer are the number of filters, local region size, stride, and padding. These several feature maps are combined to produce the output of the convolutional layer. To determine its output various activation functions are used such as sigmoid, Rectified Linear Unit (ReLU), Softmax, etc. [6].

Pooling Layer

The pooling layer is utilized to decrease the spatial dimension, number of parameters, and consequently decrease the computation of the image. The input image is resized and then separately processed each depth size by a pooling layer. Several types of pooling exist including average, stochastic, and maximum pooling and the most popular pooling is called max pooling. The most frequently employed pooling is average and maximum and these lie between the convolutional layer [7].

Fully Connected Layer

The fully connected layer links neurons from one layer to another layer neurons which is used well for image classification by training the images. The output feature maps from the last convolutional layer are flattened and connected to one or more fully connected layers known as dense layers. Flattening involves reducing multidimensional feature maps to a one-dimensional array of numbers. The outputs from the convolution/ pooling operation are multiplied by the weights associated with the network connection path and passed using the activation function [8].

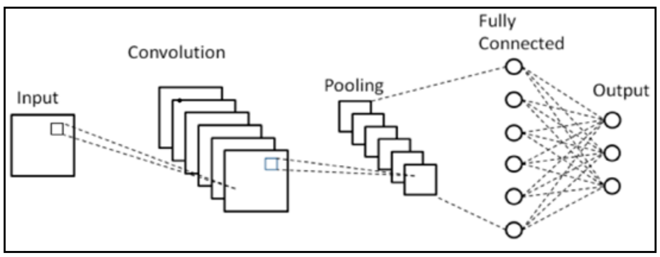


Fig. 1 Basic Model of Convolutional Neural Network

Source: https://www.mdpi.com/2076-3417/9/21/4500/htm

III. PROPOSED MODIFIED CONVOLUTIONAL NEURAL NETWORK MODEL (MODIFIED CNN)

The deep learning CNN approach has an advantage over other techniques for image processing since the images are processed using a highly optimized structure. This model has been developed for handwritten MODI character set recognition and is illustrated in Fig. 2. Our model comprises of numerous convolution layers preceding sub-sampling (pooling) layers, while the ending layers are fully connected layers along with input and output layers broadly. The input layer receives the input as an image with its height, width, depth (h×w×d), and pixel size of an image. Initially, the acquired input images (which may be colored) are pre-processed i.e., images are converted into grayscale.

Usually, the convolutional layer is used for feature extraction and its accuracy may be enhanced by the incorporation of additional convolutional layers. This layer extracts sharp features from input images, including corners, edges, and endpoints. In our model, we have constrained five convolutional layers where each layer performs a batch normalization to reduce overfitting after a ReLU activation function. A set of convolution filters or kernels of size 3×3 constitute the convolutional layer. Each filter possesses a specific set of weights to extract significant features. Subsequently, downsampling is accomplished using a pooling layer with Max-Pooling 2×2 sub-matrices for each convolutional layer. Generally, the Max-Pooling layer is accomplished to reduce the dimensions of the feature map. These maps are generated by convolutional operations to shrink large-size feature maps. The features are then passed throughout a dense layer of each node and the ReLU activation function is used for non-linearity purpose in the dense layer. The pooled feature map is transformed into a single column and transmitted to the fully connected with the inclusion of a flattened layer. For different filter counts such as 32, 64, 128, 256, and 512, the convolution and Max-Pooling operation is repeated respectively.

The output layer consists of m neurons, where m represents the number of predicted classes. Each neuron in a layer is connected to every neuron in the previous layer. The Fully connected layer receives its input in the form of a vector from the most recent pooling layer or convolutional layer. These vectors are created from feature maps after flattening and ultimately act as a CNN classifier.

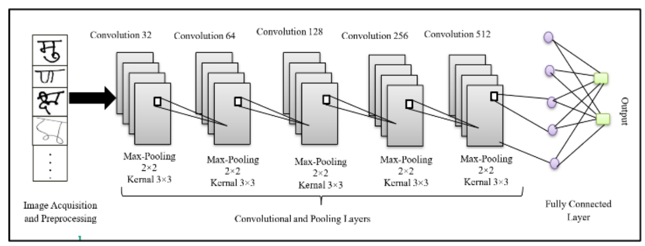


Fig. 2 Modified Convolutional Neural Network Model

IV. EXPERIMENTAL SETUP AND EVALUATION PROCESS

Our Modified CNN model is executed in Python 3.10 on a Jupyter Notebook environment. This is an open-source, web-based interactive environment that contains live code, graphics, plots, visualizations, etc., and integrates with Python programming language. The implementation of the Python (Jupyter Notebook) has been performed on the 64-bit Microsoft Windows operating system having 11th Gen Intel Core i5 processor, Installed memory 16.0 GB RAM, and Intel iRISxe graphics configuration. The evaluation process requires the specification of performance parameters, suitable datasets, and a training and testing process. To analyze the performance of our model, the parameters of interest such as epoch, batch size, learning rate, and optimizer are used. We discuss them in brief as follows:

Parameters

Performance parameters are crucial entities in determining the learning and convergence behavior of a deep learning CNN architecture. The epoch represents the number of rounds of training whereas batch size indicates the quantity of data that will be received in each epoch. To determine the speed at which the neural network learns is termed as learning rate. On the other hand, the optimizer drives the learning process of CNN to iteratively update parameters and minimize the error function in the model.

Dataset

A MODI character set is classified into 10 vowels, 36 consonants, and 10 numerals which are shown in Fig. 3. We have designed a dataset comprising of images of these numerals, vowels, and consonants with barahkhadi (i.e., Matras). A total of 56 classes for each character in MODI character set are generated involving 600 images with all matras in 36 consonants class. Thus, our dataset consists of 33,600 MODI character set images. A couple of consonants with all matras are represented in Table 1 along with the Devanagari characters.

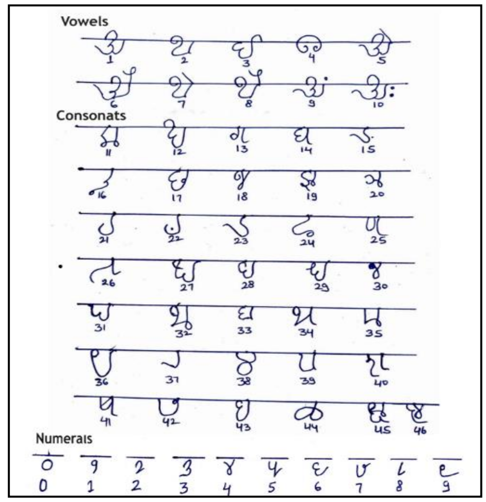
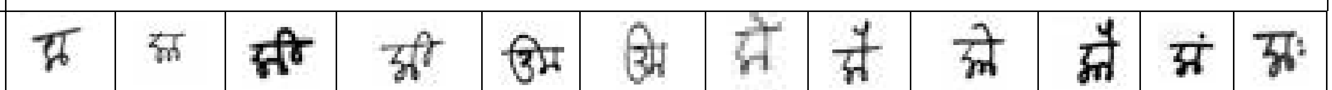
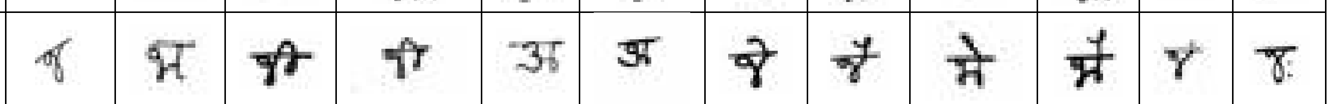
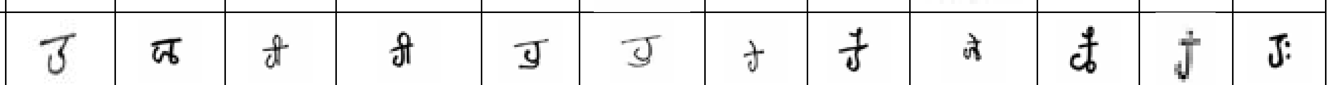
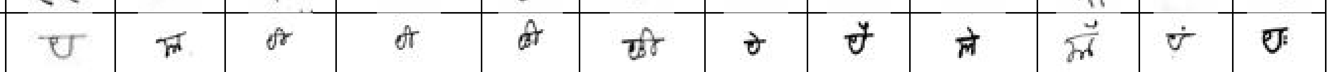
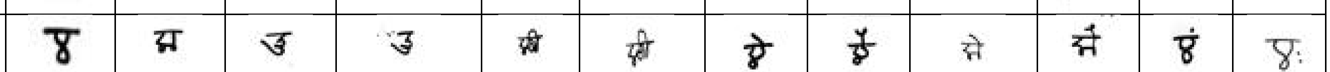
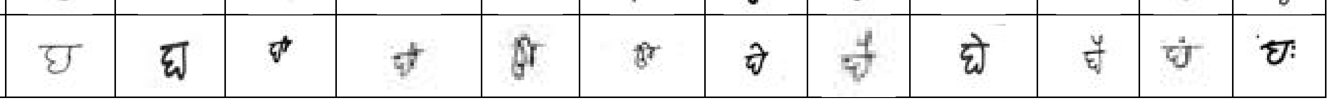


Fig. 3 MODI Characters Set

Table 1: Devanagari Characters with MODI Barahkhadi Characters – A Sample

Devanagari Characters	MODI Barahkhadi Characters
क	
ज	
ट	
त	
य	
ल	
ह	

Training and Testing Process

The training and testing of a CNN model with a limited amount of data without overfitting is a challenging task. All input images in our dataset are rescaled into grayscale using the rgb2gray Python function having 32×32 size images for better fitting in convolutional operation. This rescaled dataset is split into 70% for training and 30% for testing purposes. During the training process, a convolutional layer is added using Conv2D(), and 32 filters of size 3×3 for extracting 32 feature maps. To normalize the negative values of the features map, we use the ReLU activation function with a convolutional layer. Using MaxPooling2D(), a pooling layer is added to the sequential layers. Sequential() function attempts to add layers one by one to it sequentially. Further, the 2D feature maps are flattened into a 1D array with the function Flatten() to classify neural networks. The consistency is maintained with an epoch size of 100 per iteration in the training process. A batch size of 32 has been selected for compilation and a learning rate of 0.0001 to determine step size at each iteration. For minimizing the loss, Adam optimizer with categorical cross-entropy method has been employed for training the model. The features are subsequently routed over a Dence() layer made up of 1024 nodes using the ReLU activation function. Ultimately, the learned weights are saved for future predictions.

For the testing process, a multi-classification Softmax activation function is exercised in the output layer to construct 46 neurons for the classification. Out of the total images in a dataset, 30% of MODI character images are selected for testing and the testing accuracy is computed using evaluate function on a saved model. We have designed train_generator and test_generator functions to analyze model convergence. The model accuracy and loss matrices concerning epochs are depicted in Fig. 4.

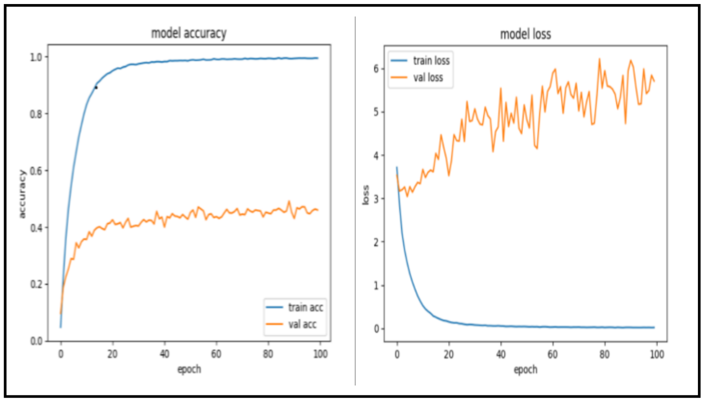


Fig. 4 Model Accuracy and Loss

V. CASE-BASED PERFORMANCE ANALYSIS

In this section, we discuss the experimental analysis of existing deep learning CNN and Modified CNN using various cases. These cases differ in the composition of character sets and size of classes in datasets predominantly. In our experimentation, the datasets mainly cover: MODI characters (without matras) images and MODI characters (with matras) images giving rise to three cases. Based on the algorithms used and the size of datasets in these cases, the performance accuracy of our Modified CNN model has been presented in Table 2. Also, its comparison with the existing deep learning CNN model is included in the same.

Case I

Case I deals with the performance of the deep learning CNN algorithm [9] in which the size of image dataset classes has been fixed as 48 and these classes are composed of characters and numerals (without matras) images. Each image in these classes is of 256×256-pixel size. When executed with deep learning CNN algorithm, the performance accuracy for recognition of MODI characters and numerals with 73.93% has been obtained. Here, 75% of images from the dataset for training and 25% for testing are used.

Case II

Case II pertains to the performance of an existing deep learning CNN model on our dataset of MODI numerals and consonant characters images (with matras) constituting 56 classes in the dataset. These classes possess 600 images, consisting of total 33,600 images of size 32×32 pixels. The model has been trained using 70% of the images and 30% images for testing. A convolutional layer of 32 filter counts with filter size 3×3 along with the parameters Adam optimizer, learning rate of 0.0001, epoch as 100, and batch size of 32 are configured for evaluation of an existing CNN model. In this order, the performance accuracy for recognition of MODI characters and numerals has been attained as 99.86%.

Case III

Case III demonstrates the performance of our Modified CNN model on the MODI dataset (with matras) including 56 classes and 33,600 images as stated above. The Modified CNN model comprises of five convolutional layers and two Max-Pooling layers. Our model has been trained on 70% of images and tested on 30% of images. Each convolutional layer consists of a set of filter counts such as 32, 64, 128, 256, and 512 with a filter size of 3×3 and each pooling layer of size 2×2. On the basis of the parameters mentioned in case II, the recognition rate has been achieved as 99.95% for recognition of MODI characters and numerals.

Cases	**Algorithms Used**	**Datasets**	**Composition of Character Sets**	**No. of Images**	**Percentage Accuracy**
[Case I] **[9]**	Deep Learning CNN	48 Classes	Characters (without Matras) & Numerals	---	73.93%,
[Case II] **[10]**	CNN	56 Classes	MODI Characters (with Matras) & Numerals	33,600 Images	99.86%
[Case III] **[Proposed]**	*Modified CNN*	56 Classes	MODI Characters (with Matras) & Numerals	33,600 Images	99.95%

VI. CONCLUSION

Handwritten character recognition for the Modi script is an interesting research field, despite the lack of research in this area. The main objective of the research is to develop a Modified CNN model that delivers better performance in recognizing handwritten MODI characters. The developed Modified CNN model comprises of five convolutions, two pooling, and one fully connected layer with a high level of recognition accuracy. As the number of convolutional layers increases more feature extraction which leads to better accuracy of the recognition. Based on the aforementioned cases, we have observed that the performance of our proposed model is well when compared to the existing deep-learning CNN model on the same dataset. Experimentation with various constraints such as the number of convolution layers, filter size, pooling layer, and fully connected layer, etc., helped for handwritten MODI characters set recognition and obtained the highest accuracy of 99.95%. Thus, we conclude that this model may be useful for character recognition in any other script.

REFERENCES

[1] Chandankhede C. and Sachdeo R., “Offline MODI Script Character Recognition using Deep Learning Techniques”, Multimedia Tools and Applications, pp.1-12, 2023.

[2] S. Joseph and J. George, "Handwritten Character Recognition of Modi Script using Convolutional Neural Network based Feature Extraction

Method and Support Vector Machine Classifier," IEEE 5th International Conference on Signal and Image Processing (ICSIP), pp. 34-36, 2020.

[3] Solley Joseph, Ayan Datta, Oshin Anto, Shynu Philip, and Jossy George, "OCR System Framework for MODI Scripts using Data Augmentation and Convolutional Neural Network", Data Science and Security Proceedings of IDSCS Lecture Notes in Networks and Systems, Springer, Vol. 132, 2020.

[4] Savitri Chandure and Vandana Inamdar, "Handwritten Modi Character Recognition using Transfer Learning with Discriminant Feature Analysis," IETE Journal of Research, pp. 1-11, 2021.

[5] Panchfula Lohkare, Rutuja Pol, Snehal Shinde, Omkar Zaware, and Prof. Vasudha Patil, "Classification of Ancient Modi Script Characters using Convolutional Neural Network", International Journal of Advanced Research in Science, Communication, and Technology, Vol. 3, Issue 10, May 2023.

[6] Ahlawat S, Choudhary A, Nayyar A, Singh S, and Yoon B, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)", MDPI- Sensors Basel, Switzerland, pp. 33-44, 12 June 2020.

[7] Raymond Ptucha, Felipe Petroski Such, Suhas Pillai, Frank Brockler, Vatsala Singh, and Paul Hutkowski, "Intelligent Character Recognition using Fully Convolutional Neural Networks", Pattern Recognition, Vol. 88, pp. 604-613, 2019.

[8] Jiuxiang Gua, Zhenhua Wangb, Jason Kuenb, Lianyang Mab, Amir Shahroudyb, Bing Shuaib, Ting Liub, Xingxing Wangb, Li Wangb, Gang Wangb, Jianfei Caic, and Tsuhan Chen "Recent Advances in Convolutional Neural Networks", Pattern Recognition, Vol. 77, pp. 354-377, 2018.

[9] Jidnyasa Kondhare, V.A.Yaduvanshi, Ruchira Patil, and Radhika Kaldate, "Recognition of Handwritten Modi Digits and Characters by Using Deep Learning Algorithm", Journal of Emerging Technologies and Innovative Research (JETIR) JETIRFP06101, Vol. 9, Issue 8, pp. 563-572, August 2022.

[10] Anirudha Ghosh, A. Sufian, Farhana Sultana, Amlan Chakrabarti and Debashis De, "Fundamental Concepts of Convolutional Neural Network" In Book- Recent Trends and Advances in Artificial Intelligence and Internet of Things, pp. 519-567, 2020.