IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Hindi Handwritten Character Recognition using Convolutional Neural Network

Karishma Verma[1*], Manjeet Singh[2]

[1] Department of Information and Technology, YMCA University of Science and Technology, Faridabad, India
[2] Department of IT & CA, YMCA University of Science and Technology, Faridabad, India

**Corresponding Author: karishmaverma555@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 16/Jun/2018, Published: 30/Jun/2018

Abstract— Convolutional Neural Networks (CNNs) have been confirmed as a powerful technique for classification of visual inputs like handwritten digits and faces recognition. Hindi handwritten character recognition (HHCR) is one of the challenging issues in machine vision. This study aims to investigate the performance of Convolutional neural networks (CNNs) on HHCR problems. To investigate the performance of different CNNs, a dataset of Hindi handwritten characters has been used as ground truth data. Different optimizers have been implemented on different parameters to determine the test accuracy of the proposed architecture.

***Keywords*—** Convolutional neural network, Handwritten character recognition, Deep learning, Hindi character dataset.

I. INTRODUCTION

Hindi written in the Devanagari script is the official language of the Government of India. The Constituent Assembly of India adopted Hindi written in Devanagari script as the official language of the Republic of India. Hindi is the fourth most-spoken first language in the world, after Mandarin, Spanish and English. Alongside Urdu as Hindustani, it is the third most-spoken language in the world, after Mandarin and English. In computer vision the main core problem is image classification in which input images should be assigned to a label from a fixed set of categories based on its visual content. One of the most important applications of image classification is optical character recognition (OCR).

In OCR an algorithm will be trained on a dataset of known characters in order to learn how to classify characters included in test set accurately. During last decades, a variety of algorithms has been developed for classifying letters/digits. However, most of the developed algorithms in this area are focused on digit recognition. In the early stage of OCR, template matching and structural analysis are used popularly [1]. In these types of algorithms, the templates for recognition problem had been made by averaging a few samples of letters/digits. In a lot of samples, these algorithms were so simple to accommodate the various shapes of all samples and yielded poor results for OCR problems.

The late 80’s in order to take the advantages of massive samples, classification methods such as artificial neural networks had been utilized popularly for recognition problems [2]. In the last decade, machine learning methods such as support vector machines (SVMs) have been applied for pattern Recognition problems [3]. Neural networks (NNs) are another solution to resolve recognition problems. In this approach, a large number of handwritten letters/digits known as training set are fed into the algorithm in order to infer rules automatically for handwritten character recognizing [4].

Handwritten character recognition is a nontrivial task as it seeks to recognize the correct class for user independent handwritten characters. This problem becomes even more challenging for a highly stylized, and potentially juxtaposition characters comprising language like Hindi. Since, Hindi character set encompasses a wide range of characters that are morphologically complex. There are some other characters that are made of two or more letters of the language (known as compound letters), which make the recognition task even more challenging. As a result, the improvements over the years in Hindi character recognition are significantly less as compared to the other languages. In this paper, we propose a convolutional deep model to recognize Hindi handwritten characters. In this paper, we decomposed Hindi character set into its basic subsets (i.e., vowels and consonants) and trained a deep convolutional neural network to recognize the characters to the extent. We have developed a convolutional model by adding different parameters in order to recognize Hindi characters with great accuracy using TensorFlow in CPU. TensorFlow is the open source library which developed by Google brain team for internal use of Google, across range of task.

The coming of deep learning led to using the convolutional neural network (CNN) in machine vision problems [5]. Based on Hubel and Weisel's early work on cat's visual cortex, CNN is a biological inspiration of multilayer perceptron (MLP) [6]. In the multilayer artificial neural network proposed by Fukushima has been used for recognition of handwritten character and other computer vision problems [7]. LeCun et al. [8] in this paper he proposed a gradient based learning form of CNN applied on document to recognize it efficiently. Considering the impact of previous works to growing CNN up, the most impressive research had been performed by Krizhevsky and et al. in [9]. They used the deep convolutional neural network to classify Image Net dataset. Recent researches on CNN have been focused on computer vision problems such as image segmentation, image captioning and image classification [10] [11] [12]. There have been a lot of researches on handwritten digit recognitions and recognition of characters in various languages [13]. As a result of potential confusion due to the similarity of handwritten characters and abundance in categories, Hindi character recognition (HCR) is much harder than English character recognition problems.

II. Methodology

A. Convolutional Neural Network (CNN)

Convolutional Neural Network is biologically inspired models which inspired by the research of the two researchers **D. H. Hubel** and **T. N. Wiesel.** They proposed an explanation about how the mammals perceived the world around them using layered architecture of neurons in the brain, which in turn inspired the engineers to develop pattern recognition mechanism in computer vision. CNN was introduce by Yann LeCun in 1998 and is used in many application now.

In CNN, Convolution is the process where two source of information are intertwined. In other words we can say that convolution is the mixing of two buckets having some information forming a single bucket to produce some meaningful information. When we apply convolution on images then we consider two dimension of images width and height of image. The first bucket is considered as our input image and the second bucket is the kernel or filter. A filter is also called kernel which is a small size matrix in comparison to input image that consist of real value. The real value of filter changes by learning iteration over training dataset which indicates network is learning to find out which region is significant for feature extraction from the data. We compute the dot product of filter matrix with input image. The convolved value is obtained by summing up the resultant terms from the dot product. The patch selection is then slided by a certain amount called stride value, the process is repeated until the entire image is processed. The output of this convolved layer is called as the feature map in deep learning. This output is then subsampled, convolved, and forwarded to the fully connected layer and we obtained a 1D feature vector. The fully connected layer models the input by applying non linearity using Rectified Linear Unit given by f(x) =max (0, x). There are many popular non linearities but here we are using ReLu instead of f(x) =tanh(x) and f(x) = (1 + e−x) -1 because training with gradient is much faster with ReLu than on other non linearities.

In our proposed work the architecture of convolution neural network is a sequence of layers which transform one volume of activation to other using different function. Two different type of CNN models were created and implemented with different optimizer to evaluate the performance of the network. We use mainly four types of layer to build the CNN architecture: Convolution layer, Relu layer, pooling layer, fully connected layer. In more detail:

- INPUT[50*50*3] will hold the input value as pixel, here we are consider the image of size 50 by height, 50 by width and 3 indicate the RGB value of an image.
- CONV layer will compute the output by the summation of dot product of input layer and kernel or filter and the output by this layer is the called feature map. This may result in the volume [50*50*50] if we use filter size 50.
- RELU is then applied to this to get the result unchanged. RELU is the activation function such as max**(0,x)** .
- POOLING layer will downsample the volume to [25, 25, 50].
- FC (fully connected layer) will compute the class score using Softmax function.

B. The convolution theorem

To develop the concept of convolution, we use the convolution theorem, this theorem is very powerful and applied in many sciences. The convolution theorem is the reason why fast Fourier transform algorithm is the most important algorithm of the 20th century.

$$h(x) = f \otimes g = \int_{-\infty}^{\infty} f(x-u)g(u)\,du = \mathcal{F}^{-1}\left(\sqrt{2\pi}\mathcal{F}[f]\mathcal{F}[g]\right)$$

$$\text{feature map} = \text{input}\otimes\text{kernel} = \sum_{y=0}^{\text{columns}}\left(\sum_{x=0}^{\text{rows}} \text{input}(x-a, y-b)\,\text{kernel}(x,y)\right) = \mathcal{F}^{-1}\left(\sqrt{2\pi}\mathcal{F}[\text{input}]\mathcal{F}[\text{kernel}]\right)$$

The first equation is the one dimensional continuous convolution theorem of two continuous functions. The second equation is the 2D discrete convolution. An example of CNN model is illustrated in the figure below:

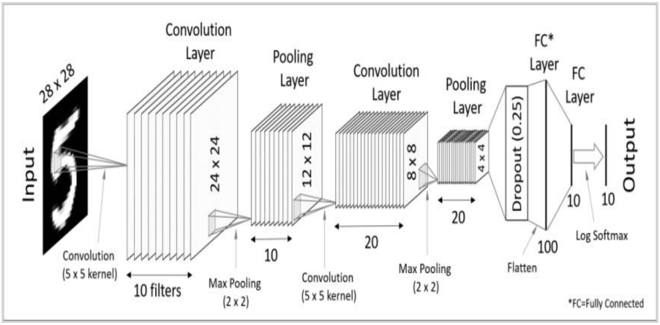

Figure 1. Convolution Neural Network

III. RESULTS AND DISCUSSION

In order to determine the performance of the different architectures of CNN, Hindi handwritten dataset was utilized which was proposed by Ashok Kumar pant [18]. All the implementation has been done in Python language with Google's library TensorFlow on CPU.

A. The dataset

Hindi is the official language accepted by the government of India. It is the fourth most language which is spoken in the world after English, Mandarin, and Spanish and written from left to right. The Hindi language has two classes, vowel class and consonant class. The numbers of vowel are twelve (12) and the numbers of consonant are thirty six (36) and also we were using digits from 0 to 9. The sample images of the dataset are shown below in the figure 2. Ashok kumar pant [18] proposed the dataset in which there are 221 samples of vowels in per class, 205 samples of consonants in per class and 228 samples of digits in per class. From these classes we divide the dataset into test class and train class.

Figure 2. Sample image of consonants, vowels and digits

During the training of the CNN model we increment the dataset by augmenting more number of images than the original dataset. Increasing number of images in the dataset will not only used for the learning purpose of the model but it will also take the possible shift of the isolated character. The size of each image is 36 by 36. The final dataset after augmentation is given in table 1.

Table 1: Dataset of Hindi character

Dataset	**Vowels**	**Consonants**	**Digits**
Classes	12	36	10
Sample per Class	221*5	205*5	228*5
Total	13,260	36,900	11,400

B. Convolutional neural network

In this section we built two models of CNN, the first model is Model A and the second model is Model B. In model A the batch size is 64, the filter size is 5*5 in all convolution layers and the number of filters is 16. Two convolution layers were used so the value of first hidden layer (h1) is 64 and second hidden layer (h2) is 32, pooling layer of size 2*2 and two fully connected layers were implemented. A stride of 1 is used by convolution layer and a stride of 2 is used by pooling layer with a keep probability of 0.75 with base value 0.86 for SGD optimizer and 0.9 for Adam optimizer.

In model B two convolution layers were used the first hidden layer(h1) value is 64 and the second hidden(h2) layer value is 32, the first convolution layer used the filter of size 3*3 and the second convolution layer used a filter size of 5*5. We did this to improve the performance of the model B as now it is capable to learn those characters which are different by a small point we use the value of keep probability as 0.5 in the dropout layer instead of 0.75 as in model A and a alpha value of 0.86 for SGD optimizer and beta value of 0.9 for Adam optimizer. The table 1 shows the summarized view of all three categories of experiment done on both the models of CNN.

So in the first category of experiment, we use two different optimizer and evaluate the performance of the models, model A achieved a test accuracy of 99.9% using SGD and 99.9% using Adam optimizer over 40,000 iterations, similarly in the model B we achieved an accuracy of 98.9% on SGD optimizer and 99.5% on the Adam optimizer with same iterations as mention above. The result of loss and accuracy for both the models are shown below.

C. For Model A

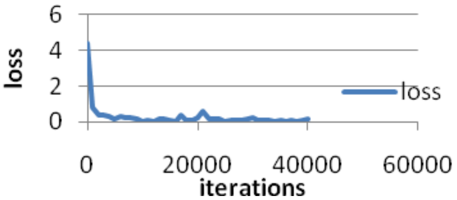

Figure 3. Loss in model A using SGD optimizer

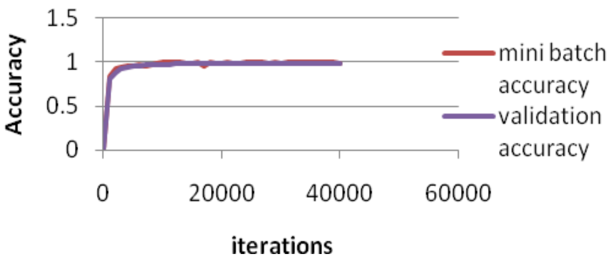

Figure 4. Mini batch accuracy vs validation accuracy in model A using SGD optimizer

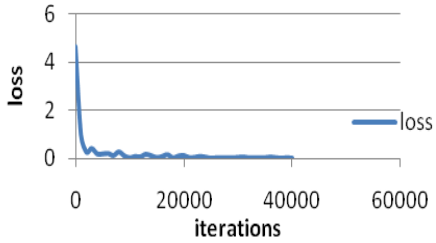

Figure 5. Loss in model A using Adam optimizer

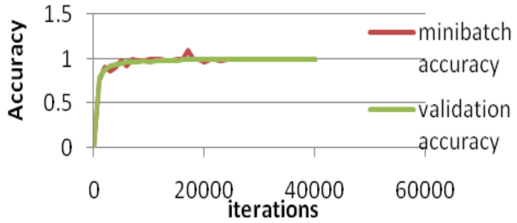

Figure 6. Minibatch accuracy vs validation accuracy in model A using Adam optimizer

D. For Model B

The experimental results for model B is shown in the figure below.

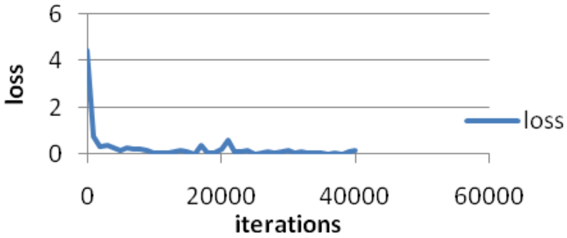

Figure 7. Loss in model B using SGD optimizer

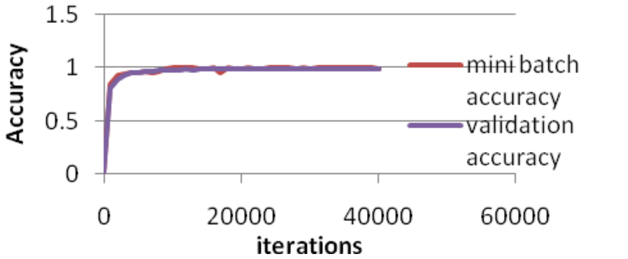

Figure 8. Minibatch accuracy vs validation accuracy in model B using SGD optimizer

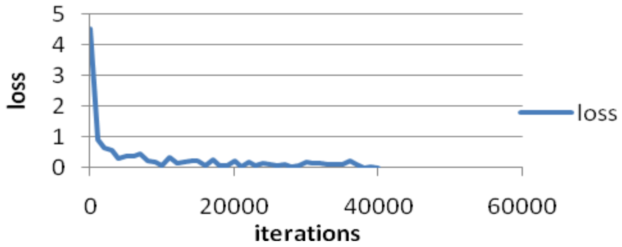

Figure 9. Loss in model B using Adam optimizer

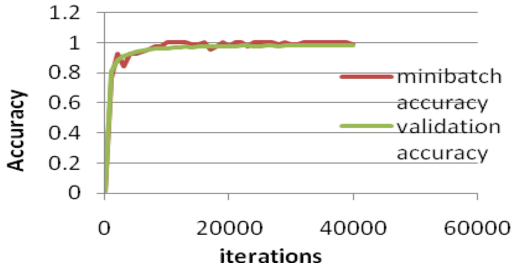

Figure 10. Minibatch accuracy vs validation accuracy in model B using Adam optimizer

The result of training model A using SGD optimizer over 40,000 iterations is found to be 99.9% and by using Adam optimizer the test accuracy is 99.9%. For model B the test accuracy using SGD optimizer is 98.9% and 99.5%.

E. Summarized view of various category of experiment

For the second category we had changed the numbers of hidden layer in model A we kept changing the number of hidden layers (h1, h2). In this first we took the pair 32, 32 using SGD optimizer with 20,000 iterations we get97.8% accuracy while the same experimented is conducted using Adam optimizer we get 99.3%. Second time we use the pair 128, 32 using SGD and Adam optimizers we achieved a test accuracy of 98.2% and 3.0% respectively. After that we use the pair as 128,128 on the same optimizer and we achieve 98.4% and 99.8% respectively.

For the third category we did the same as we done in the second category but here the model is model B. In this first we took the pair 32, 32 using SGD optimizer with 20,000 iterations we get96% accuracy while the same experimented is conducted using Adam optimizer we get 96.3%. Second time we use the pair 128, 32 using SGD and Adam optimizers we achieved a test accuracy of 98.3% and 99.8% respectively. After that we use the pair as 128,128 on the same optimizer and we achieve 98.4% and 99.9% respectively.

From the above conducted experiments it was clear that this model work best with Adam optimizer rather than SGD optimizer but if we had GPU and large memory then we would increase its accuracy by increasing its iteration because the time to train a network is very high.

Table 2. Comparison of Models with different parameters

s. no.	**model used**	**Iteration**	**optimizer**	**Beta**	**hidden Layer h1**	**hidden Layer h2**	**accuracy**
Experiment 1							
1.	A	40,000	SGD	0.86	64	32	99.9%
2.	B	40,000	SGD	0.86	64	32	98.4%
3.	A	40,000	Adam	0.9	64	32	99.9%
4.	B	40,000	Adam	0.9	64	32	99.5%
Experiment 2							
5.	A	20,000	SGD	0.86	32	32	97.8%
6.	A	20,000	SGD	0.86	128	32	98.2%
7.	A	20,000	SGD	0.86	128	128	98.4%
8.	A	20,000	Adam	0.9	32	32	99.3%
9.	A	20,000	Adam	0.9	128	32	3.0%
10.	A	20,000	Adam	0.9	128	128	99.8%
Experiment 3							
11.	B	20,000	SGD	0.86	32	32	96%
12.	B	20,000	SGD	0.86	128	32	98.3%
13.	B	20,000	SGD	0.86	128	128	98.4%
14.	B	20,000	Adam	0.9	32	32	96.3%
15.	B	20,000	Adam	0.9	128	32	99.8%
16.	B	20,000	Adam	0.9	128	128	99.9%

IV. CONCLUSION and FUTURE SCOPE

In this work we effectively utilize the CNN and achieved a better performance in the classification and recognition of handwritten character and numeral digits than the shallow learning model for example ANN, SVM etc.

If the hardware we had are more supportive than we will able to achieve a higher accuracy even with complicated dataset of handwritten characters. The results that we achieved are analogous with the previous related work however; they did not test on large amount of hand written character data set. For recognition of Hindi handwritten characters we implemented two different models with different parameters to evaluate the accuracy and efficiency of the network and we conclude that the parameters like dropout, beta value and learning highly influenced the network. The size of filter also has a great effect on the learning of the model.

Our future endeavour for this system is to improve the accuracy in complicated characters, end hardware that can help in building more complex model over network, and also we would like to introduce some more number of classes that would contain the compound character combined with the single characters to create a more complex class of Hindi characters. The scope of this system can be expanded by including hand written Hindi words in which first we will segment the words to get the isolated character and then the classifier will then perform its task separately.

REFERENCES

[1] S. Mori, C. Y. Suen, and K. Yammamoto, "historical review of OCR research and development," Proc. IEEE, vol. 80, pp. 1029-1058, 1992.

[2] A. Rajavelu, M. T. Musavi, and M. V. Shirvaikar, "A neural network approach to character recognition," neural network, vol. 2, pp. 387-393, 1989.

[3] H. Byun and S. W. Lee, "Applications of support vector machines for pattern recognition: A survey," in pattern recognition with support vector machine, Springer, pp. 213-236, 2002.

[4] P. D. Gader, M. Mohamed, and J. H. Chiang, "Handwritten word recognition with character and inter-character neural networks," IEEE Trans. Syst. Man Cybern. Part B Cybern., vol. 27, pp. 158 - 164, 1997.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol.521, pp. 436-444, 2015.

[6] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," J. phyiol., vol. 195, pp. 215- 243, 1968.

[7] K. Fkushima, "Neocognitron: A self- organizing neural network for a mechanism of pattern recognition unaffected by shift in position," boil. Cybern., vol. 36, pp. 193-203, 1980.

[8] Y. Lecun, L. Buttou, Y. Bengio, and P. Haffner, " Gradient-based learning applied to document recognition," proc. IEEE, vol. 86, pp. 2278- 2324, 1998.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, " Imagenet classification with deep convolutional neural networks," in advances in neural information processing system, pp. 1097-1105, 2012.

[10] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, "Learning hierarchical features for scene labeling," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, pp. 1915-1929, 2013.

[11] O. vinyals, A. Toshev, S. Bengio, and D. Erhan, " show and tell: Aneural image caption generator," in proceeding of the IEEE conference on Computer Vision and Pattern Recognition, pp. 3156-3164, 2015.

[12] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "flexible High performance convolutional neural networks for image classification," in IJCAI proceedings-International Joint conference on Artifical Intelligence, vol. 22, pp.1237, 2011.

[13] E.Kussul and T. Baidyk, "Improved method of handwritten digit recognition tested on MNIST database," Image Vis. Comput., vol. 22, pp. 971-981, 2004.

[14] Bishwajit Purkaystha, Tapos Datta, Md Saiful Islam, "Bengali Handwritten Character Recognition Using Deep Convolutional Neural Network", 20th International Conference of Computer and Information technology(ICCIT), 22-24 December, 2017.

[15] Samad Roohi, Behnam Alizadehashrafi, "Persian Handwritten Character Recognition Using Convolutional Neural Network," 10th Iranian Conference on Machine Vision and Image Processing, Nov, 22-23, 2017.

[16] Mahesh Jangid and Sumit Srivastava, "Handwritten Devnagari Character Recognition Using Layer Wise Training of Deep Convolutional Neural Networks and Adaptive Gradient Methods", journal of imaging 2018.

[17] Jia xiaodong, gong wednog, yuan jie, "Handwritten Yi Character Recognition with Density Based Clustering Algorithm and Convolutional Neural Network", IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference Embedded and Ubiquitous Computing (EUC) 2017.

[18] Ashok Kumar Pant, Prashnna Kumar Gyawali, Shailesh Acharya, "Deep Learning Based large Scale Handwritten Devanagri Character Recognition", 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) 2015.

[19] Ajay Indian, Karamjit Bhatia, "A combination of feature extraction for offline handwritten hindi numerals recognition", vol-6, Issue-5, May 2018

Authors Profile

Ms. Karishma verma received the degree of Bachelor of Technology in computer science and engineering in 2016 from DCRUST, Murthal, India and pursued Master of Technology in Information Technology from YMCA University of Science and Technology, Faridabad, India in 2018. Her main research work focuses on Aritifical intelligence, Machine learning and IOT.

Dr.Manjeet singh pursued Master of Science from MDU, Rohtak Haryana in 1999, received the degree of Master of Technology in computer science and engineering in 2002 from Guru Jambheshwar University and pursed Ph.D. in Computer Engineering from MDU, Rohtak in 2002. He is currently working with a designation of professor and proctor in Department of IT & Computer Application in YMCA University of science and technology Faridabad, India. He has a teaching experience of 15 years. He has published more than 50 research papers in reputed international journals and conferences including IEEE and it's also available online. His main research work focuses on Artificial Intelligence, Soft Computing, Natural Language Processing, Computer Networks, Information Retrieval, Semantic Web, Compiler Design, and Data Strucures and Algorthim Design.