American Journal of Electrical and Computer Engineering
2024, Vol. 8, No. 2, pp. 21-30
https://doi.org/10.11648/j.ajece.20240802.11

Research Article

A Recognition System for Devanagari Handwritten Digits Using CNN

Nawaraj Ghimire*

Faculty of Science Health Technology, Tribhuvan University, Sanothimi Campus and Nepal Open University, Kirtipur, Nepal

Abstract

A Recognition System for Devanagari Handwritten Digits using CNN, a novel approach to recognizing transcribed digits in the Devanagari script using Convolutional Neural Networks (CNN). This framework represents a significant contribution to the field of pattern recognition and language processing objective of the research project is to perform a literature review, identify an algorithm for a digits recognition system implement the Devanagari digits recognition system for educational activities. In the first phase, a dataset of 150 transcribed digit images is curated, allocating 75% for training (113 images) and 25% for validation (37 images). A Convolutional Neural Network (CNN) is designed with five convolutional layers, each utilizing 3 × 3 filters with 16, 32, 64, 128, and 128 feature maps, respectively. The experiments conducted involve varying the number of epochs, with results captured at 5, 10, 20, and 100 epochs. This comprehensive evaluation aims to understand the model's convergence and performance over different training durations. The outcomes of this phase contribute to the fine-tuning and optimization of the model for subsequent phases. In the second phase, the dataset is expanded to 100*10 (1000) images, each resized to 28 × 28 pixels through cropping. The CNN architecture remains consistent, with the previously determined layer configuration. Similar experiments are conducted, assessing the model's performance over 5, 10, 20, and 100 epochs. This model with a data size of 1000 demonstrates superior accuracy (100% on mini-batches) compared to the 150 model, with consistently high validation accuracy, while both models exhibit decreasing trends in mini-batch and validation losses, favoring the larger dataset, and maintaining a constant learning rate at 0.0100, albeit with a slightly longer time elapsed for each epoch due to the increased data size. 98.37398 accuracy in the phase 2 experiment in 100 epochs. Similar research and contributions and Devanagari's character and word recognition system.

Keywords

Deep Learning, CNN, Image Processing, Digit Recognition, Ethnic Language

1. Introduction

Handwritten recognition systems turn handwritten text or characters into digital text. One component of handwritten recognition systems is the conversion of handwritten numbers into digital numbers. Handwritten digit recognition is a fundamental and common task in the broader field of handwriting recognition systems [1]. The Devanagari script is the most widely used in Nepal and India, and it is also used in other Asian countries [2] Nepali is an Indo-Aryan language that

*Corresponding author: nawaraj.ghimire@sac.tu.edu.np (Nawaraj Ghimire)

Received: 20 March 2024; **Accepted:** 7 April 2024; **Published:** 29 July 2024

uses the Devanagari script to write. Over 17 million Nepali speakers live in countries such as Nepal, Bhutan, Myanmar, Brunei, and India [3]. The Federal Democratic Republic of Nepal uses the Nepali language as its official working language which includes the Devanagari script. Devanagari digits and Devanagari numerals have been used in the Nepali language, Calendar, and in historical documents [4]. The Devanagari script consists of a single character, 12 vowels, 36 consonants, and 10 digits. Devanagari numbers are the same as English numbers 0-9 [5].

Handwritten character and number recognition is one of the most demanding and exciting fields of pattern recognition and image processing. Convolutional neural networks (CNN) are a subset of machine learning. This is a description of deep learning, a branch of machine learning that includes multi-layer neural networks, also known as deep neural networks. Convolutional Neural Network is a specialized type of neural network architecture designed primarily for visual data processing and analysis [6] CNNs play an important role in various fields such as image processing, CNN is used for fault detection and classification. A simple artificial neural network (ANN) has an input layer, an output layer, and several hidden layers between the input and output layers [7]. CNN has one Architecture very similar to ANN. Each layer of an ANN has multiple neurons. The weighted sum of all neurons in one layer becomes the input to the neurons in the next layer, and the next layer adds a bias value. In CNN, layers have many dimensions. Here, several neurons are connected. All neuron in the layer is connected to the end of the receptive field. A train of the network generates a cost function. Compare the input and provide the output of the network with the desired output. The signal continuously propagates through the system and updates the common weights and biases of all received fields to minimize the value of the cost function and improve the performance of the network [8].

The background of this study is to influence the advancement of deep learning techniques to provide a more accurate, efficient, and strong digiting the Devanagari Handwritten Digits Recognition System.

2. Statement of Problem

Devanagari handwriting poses a significant challenge due to the diverse nature of handwriting styles and the complex features of writing. Current recognition systems may not fully account for the nuances of Devanagari's digital representation. The need for a robust recognition system using a convolutional neural network (CNN) arose to improve the accuracy and efficiency of Devanagari handwritten digit recognition. This research aims to develop and optimize a CNN-based Recognition System specifically tailored for the complexities of the Devanagari script, addressing challenges such as varying writing styles, size variations, and the intricate nature of the script's characters.

3. Literature Review

Artificial Intelligence (AI) has grown exponentially, especially in the field of computer vision, which is the ability of a machine to interpret and understand the visual world. Devanagari scripts can expect differences in writing styles of different nationals researching the various Machine Learning Algorithms Support Vector Machines (SVM) [9], K-Nearest Neighbors (KNN) [10], Random Forest Classifier (RFC) [11] for the Neural Networks and Deep Learning Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) [12] and Long Short-Term Memory (LSTM) Networks CNN chosen for specialized for image processing and pattern recognition [13]. In general, a CNN consists of a convolutional layer, a spatial pooling layer, and a fully connected layer. The convolutional layer is responsible for extracting features from the feature map to the layer below. all layers of the CNN are trained together using a backpropagation algorithm. However, a strategy of interleaved training of a subset of layers of the CNN is proposed to normalize the neural network [14].

According to handwritten digit recognition [15] investigation of normalization and feature extraction techniques [16]. when creating character recognition systems, improved normalizing functions and direction feature extraction methodologies used methods on research. Based on three different data sources, we compare ten normalization algorithms (seven based on dimensions and three based on moments) and eight feature vectors. Eighty categorization accuracies for each dataset are generated by combining the normalization functions with feature vectors [17]. The comparison of normalizing functions reveals that aspect ratio mapping outperforms its baseline equivalents, whereas moment-based functions perform better than dimension-based ones [18].

The first step is to acquire a Devanagari digit can be done using a scanner or a camera. Image pre-processing Once the image is acquired, it needs to be pre-processed to improve the quality of the image and prepare it. This may include steps such as image enhancement, noise removal, and binarization. Feature extraction, various features such as text and image features are extracted from the handwritten digit These features are used as input for the deep learning model CNN.

Figure 1. *Devanagari Number Representation.*

Devanagari script has ten digits 0-9 Handwritten Devanagari script is peculiar compared to English script as there is no cursive connected writing and one has to scribble digits or even curves, Matras by lifting the handwriting system of Devanagari script is a mixture of characters, numerals, and syllabary digit is written as figure 1.

4. Materials and Methods

4.1. Related Work

For this research project, prepare data samples based on the data collection method, adding new variation data samples. This also allows you to contribute to the academic community for further research. Devanagari handwritten digits are collected from the undergraduate students of Sanothimi Campus provided a Piece of paper is given to them for writing isolated handwritten digits into the paper using a ball pen. Figure 2 Then all papers are scanned using a scanner separately to create a dataset of Devanagari digits. Each scanned image is labeled manually it and classified accordingly.

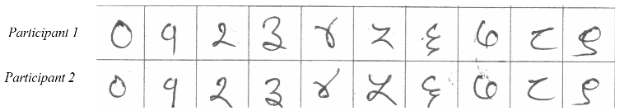


Figure 2. *Participant Devanagari Handwriting Digits.*

Devanagari handwritten image after some preprocessing techniques figure 2 original image figure 3 cropped image and figure 4 normalized image For example, first digit '0' and last digit "9' 0 is labeled as 0_1.jpg and 9_1.jpg. shown in Table 1 crop into 28*28 Class one consists of 100 samples of digits and a total number of 1000 numerals The newly created dataset has the following Characteristics.

1) In this Devanagari digit dataset consists of a total of 1000 used into 0-9 classes.
2) Regular use pen and pencil are used
3) different participants were used to create data sets.

Figure 3. *Cropped Image.*

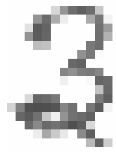
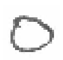


Figure 4. *Normalized image.*

Table 1. *List of Data sets with label.*

Handwritten Digit	Label	Crop file size	Normalize file size	filename
	0	78*40	28*28	0_1.jpg
	1	35*39	28*28	1_1.jpg
	2	105*100	28*28	2_1.jpg
	3	80*140	28*28	3_1.jpg
	4	58*55	28*28	4_1.jpg
	5	44*40	28*28	5_1.jpg
	6	29*37	28*28	6_1.jpg
	7	30*72	28*28	7_1.jpg
	8	70*70	28*28	8_1.jpg
	9	28*28	28*28	9_1.jpg

4.2. Proposed Model

Convolutional Neural Network (CNN) is the proposed model to determine Devanagari handwritten digit recognition (see. figure 5).

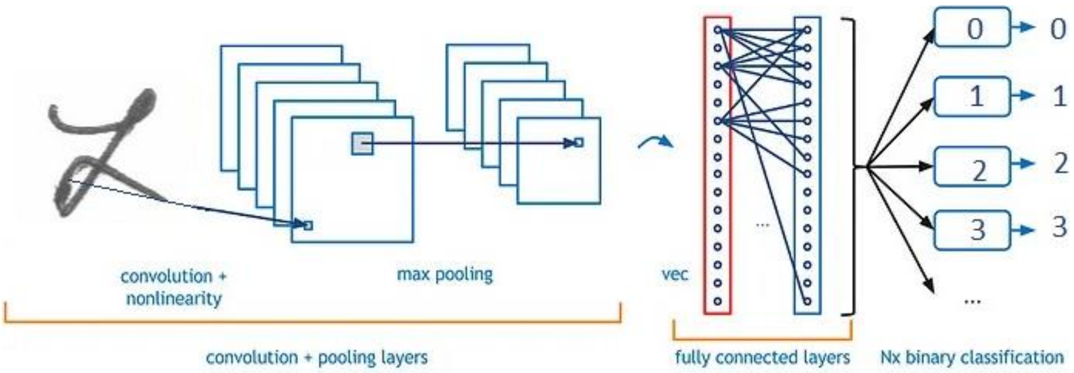


Figure 5. *Devanagari Number Representation.*

4.3. CNN Architecture

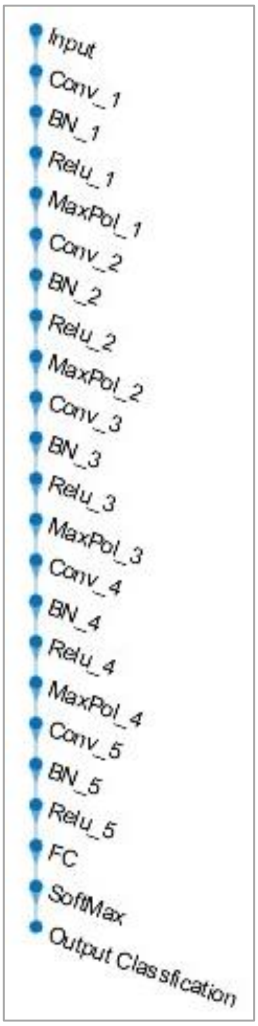


Figure 6. *Layers in CNN.*

To recognize digits, a CNN-based digit classifier is used [19]. Eight different layers are used in neural networks also six handwritten digit classifiers consist of several layers such as a convolutional layer, batch normalization Layer, Relu Layer, max-pooling layer, fully Connected Layer, softmax Layer, and classification Layer [20]. (see figure 6).

4.3.1. Image Input Layer

In this layer defines the input layer of the neural network [21]. 'imageInputLayer' creates an image input layer, specifying the input size as [28x28x3]. This implies that the input data is expected to be an image with dimensions 28x28 pixels and 3 channels (which are RGB color images. The parameter 'Name', 'Input' assigns the name 'Input' to this layer for reference in the network architecture this layer defines the input layer of the neural network. 'imageInputLayer' creates an image input layer, specifying the input size as [28x28x3]. This implies that the input data is expected to be an image with dimensions 28x28 pixels and 3 channels (which are RGB color images. The parameter 'Name', 'Input' assigns the name 'Input' to this layer for reference in the network architecture.

4.3.2. Convolutional Layer

The convolution layer line defines the first convolutional layer of the neural network. 'convolution2dLayer' creates a 2D convolutional layer. In this layer, a 3x3 filter is used for the convolutional filter and 16 filters in the layer will output 16 feature maps. This is a common setup for the initial layers of a CNN designed for image classification or feature extraction tasks [22]. Then batch normalization technique is used to improve the training of deep neural networks. It normalizes the input of a layer by adjusting and scaling the activations.

4.3.3. Activation Layer

In this particular stage of the process, the data undergoes a transformation using a specific function that adjusts its values within a certain range. In this research project rectified linear units (ReLU) were used as activation functions.

$$f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \quad (1)$$

ReLU (Rectified Linear Unit) is an activation function that introduces non-linearity by outputting the input for all positive values and zero for all negative values.

4.3.4. Pooling Layer

The pooling layer used reduces the spatial dimensions of the data, aiding in feature selection and computational efficiency [23]. In this project Max pooling is a downsampling operation that reduces the spatial dimensions of the input by taking the maximum value from a set of values within a window. 2 specifies a 2x2 pooling window. 'Stride', 2 means that the window moves with a step size of 2 pixels, reducing the spatial dimensions by half.

4.3.5. Fully Connected Layer

Fully connected layer processes features learned from previous layers. In this fully connected (dense) layer with 10 neurons or nodes. Each neuron in this layer is connected to every output from the previous layer, effectively creating a fully connected network. The 'Name', and 'FC' parameters assign the name 'FC' to this fully connected layer.

(i). Softmax Layer

Softmax layers were used as the final layer in a classification network. It converts the raw output scores from the previous layer into probabilities, making it suitable for multi-class classification problems. The 'Name', and 'SoftMax' parameters assign the name 'SoftMax' to this softmax layer.

(ii). Classification Layer

In this proposed model the classification layer defines the final classification output it is used in conjunction with a softmax layer. This layer helps the output layer which categorizes input data into different classes. The 'Name', and 'Output Classification' parameter assigns the name 'Output Classification' to this classification layer.

5. Results and Discussion

Researchers divide two datasets for experiments Phase 1: In this phase contain 0-9 Devanagari handwritten digits of normalized date of size 28*28.jpg each class has 15 images of a total of 150 images used for training and calculating accuracy.

The researcher used a dataset containing 150 Devanagari handwritten digits divided into 10 classes. Dataset normalized the size of each image into 28 × 28. So, Input to CNN is 28*28*3 in a color RGB image, we normally have 3 channels red, green, and blue. In this research project researchers have used 5 convolutional layers having 16, 32, 64, 128, and 128, and each layer 3 × 3 filters are used.

Applied RELU function in the activation layer introduces non-linearity by outputting the input for all positive values and zero for all negative values. Maxpooling is used in the pooling layer for down-sampling filters of size 2 × 2 in pooling layer.

The fully connected layer calculates the class score of a character using the Soft-max function and classifies the digit.

For the training, the network specifies 75% used for training and 25% dataset used for validation ie. 113 for training, and 37 images for validation Each experiment Number of Epoch 5, 10, 20, and 100.

Phase 2: In this phase contain 0-9 Devanagari handwritten digits date of size 28*28.jpg Each class has 100 images of a total of 1000 images for training and calculating accuracy. The researcher used a dataset containing 1000 Devanagari handwritten digits divided into 10 classes. Dataset normalized the size of each image into 28 × 28. So, Input to CNN is 28*28*3 in a color RGB image, we normally have 3 channels red, green, and blue. In this research project, researchers have used 5 convolutional layers having 16, 32, 64, 128, and 128, and each layer 3 × 3 filters are used.

Applied RELU function in the activation layer introduces non-linearity by outputting the input for all positive values and zero for all negative values. Maxpooling is used in the pooling layer for down-sampling filters of size 2 × 2 in pooling layer. The fully connected layer calculates the class score of a character using the Soft-max function and classifies the digit. For the training network specify 75% used for training and 25% dataset used for validation ie. 750 for training and 250 images for validation. Each experiment Number of Epoch 5, 10, 20, and 100.

Experiment

Phase 1: the researcher has used a dataset containing 150 Devanagari handwritten digits divided into 10 classes For the training network specified 75% was used for training and 25% dataset was used for validation ie. 113 for training and 37 images for validation for each experiment.

After the first epoch, the accuracy of mini-batches is low, but it improves significantly by the fifth epoch. Both training and validation accuracies are provided, indicating how well the model generalizes to new, unseen data. The loss decreases, suggesting that the model is learning to make better predictions.

Table 2. *Phase 1 Performance of the CNN with a number of epoch 5.*

Epoch	Iteration	Time Elapsed (hh:mm: ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
1	1	00:00:07	10.09%	20.51%	2.9026	2.4778	0.0100
5	5	00:00:09	95.41%	66.67%	0.5519	1.1754	0.0100

Accuracy 66.66846

Table 3. *Phase 1 Performance of the CNN with the number of epoch 10.*

Epoch	Iteration	Time Elapsed (hh:mm: ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
1	1	00:00:05	7.34%	7.69%	2.7801	2.4966	0.0100
10	10	00:00:08	100.00%	84.62%	0.1028	0.6449	0.0100

Accuracy 84.61538

Accuracy improves significantly after 10 epochs. Both training and validation accuracies are high, indicating good generalization. The loss decreases, showing further improvement in model predictions.

Table 4. *Phase 1 Performance of the CNN with number of epoch 20.*

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
1	1	00:00:06	6.42%	12.82%	2.7626	2.5114	0.0100
20	20	00:00:14	100.00%	79.49%	0.0265	0.5313	0.0100

Accuracy 79.48718

Continued improvement in accuracy and reduction in loss. The model might be plateauing in terms of accuracy on validation data.

Table 5. *Phase 1 Performance of the CNN with number of epoch 100.*

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
1	1	00:00:06	6.42%	12.82%	2.8182	2.5270	0.0100
50	50	00:00:21	100.00%		0.0063		0.0100
100	100	00:00:39	100.00%	82.05%	0.0039	0.4195	0.0100

Accuracy 82.05128

The model achieves 100% accuracy on mini-batches after a certain point. The loss on both mini-batches and validation data continues to decrease. The learning rate remains constant at 0.0100.

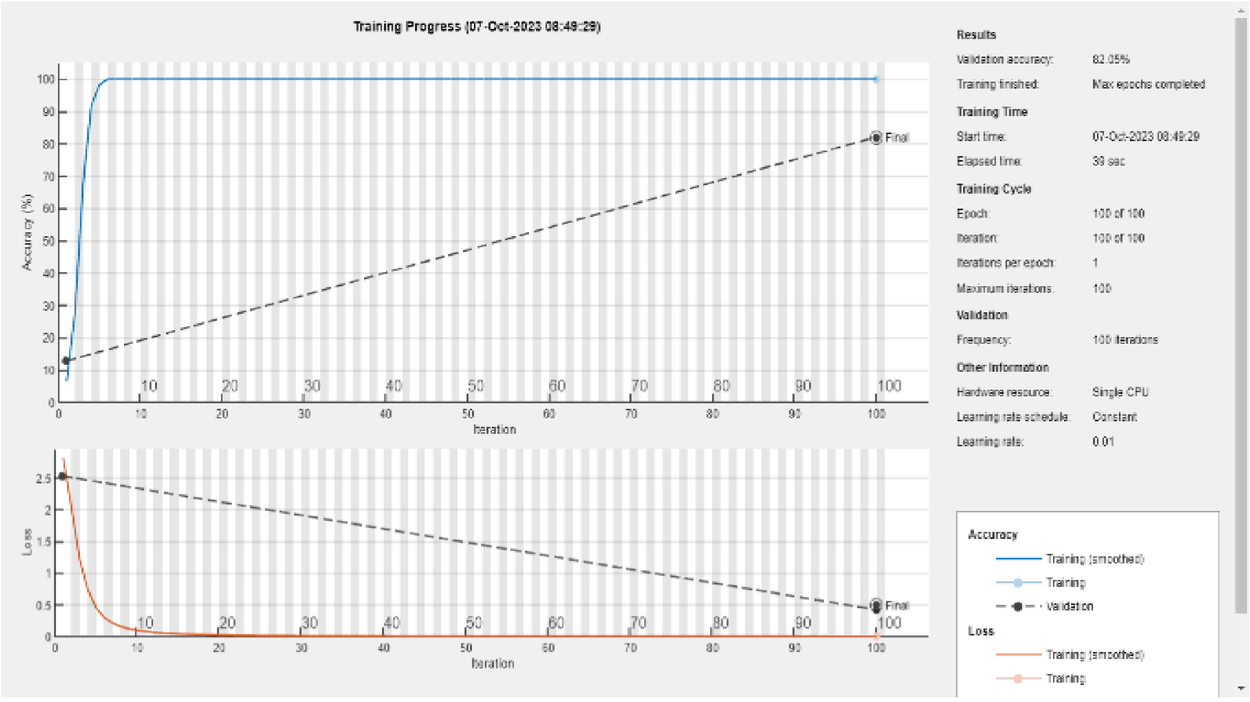


***Figure 7**. Phase 1 Training progress CNN with number of epoch 100.*

Phase 2 The researcher used a dataset containing 1000 Devanagari handwritten digits divided into 10 classes. For the training network specify 75% used for training and 25% dataset used for validation i.e. 750 for training and 250 images for validation each experiment Number of Epoch 5, 10, 20, and 100.

***Table 6**. Phase 2 performance of the CNN with number of epoch 5.*

Epoch	Iteration	Time Elapsed (hh:mm: ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
1	1	00:00:09	8.59%	13.82%	2.7490	2.5070	0.0100
5	25	00:00:23	100.00%	96.75%	0.0572	0.1764	0.0100

Accuracy 96.74797

The model quickly achieves high accuracy, reaching 100% on mini-batches. High accuracy is also observed in the validation set. The loss decreases significantly.

***Table 7**. Phase 2 performance of the CNN with number of epoch 10.*

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
1	1	00:00:06	12.50%	15.04%	2.6363	2.4576	0.0100
10	50	00:00:34	100.00%	97.56%	0.0221	0.0908	0.0100

Accuracy 97.56098

The model achieves high accuracy on both mini-batches and validation data. Losses are significantly reduced.

Table 8. Performance of the CNN with the number of epoch 20.

Epoch	Iteration	Time Elapsed (hh:mm: ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
1	1	00:00:07	4.69%	16.26%	2.7612	2.3145	0.0100
10	50	00:00:26	100.00%		0.0166		0.0100
20	100	00:00:51	100.00%	97.97%	0.0085	0.0755	0.0100

Accuracy 97.96748

High accuracy on mini-batches and validation data. The loss on both mini-batches and validation continues to decrease.

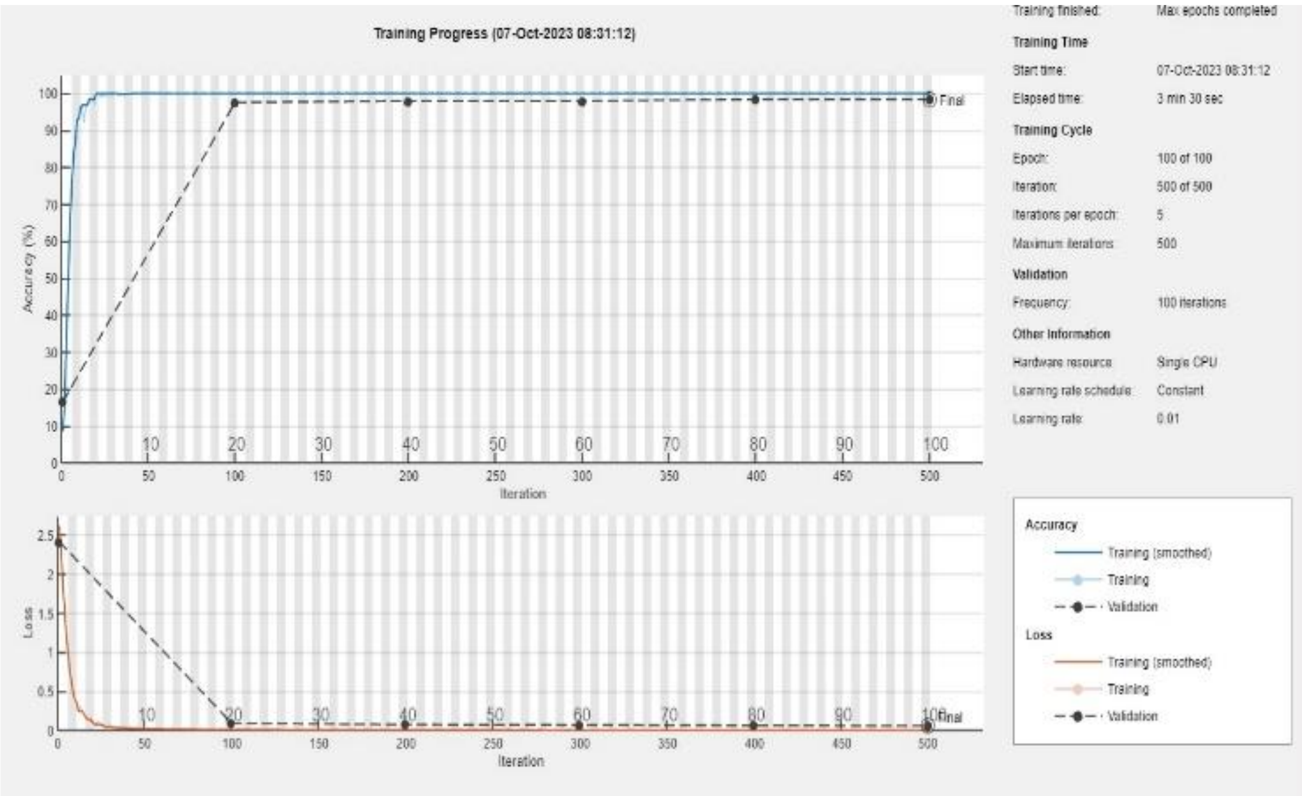


Figure 8. Phase 1 Training progress CNN with number of epoch 100.

Table 9. Performance of the CNN with number of epoch 100.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
1	1	00:00:07	8.59%	16.67%	2.6120	2.4114	0.0100
10	50	00:00:28	100.00%		0.0156		0.0100
20	100	00:00:50	100.00%	97.56%	0.0092	0.0885	0.0100
30	150	00:01:10	100.00%		0.0071		0.0100
40	200	00:01:30	100.00%	97.97%	0.0045	0.0729	0.0100
50	250	00:01:51	100.00%		0.0041		0.0100

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Learning Rate
60	300	00:02:11	100.00%	97.97%	0.0031	0.0675	0.0100
70	350	00:02:30	100.00%		0.0028		0.0100
80	400	00:02:52	100.00%	98.37%	0.0024	0.0596	0.0100
90	450	00:03:12	100.00%		0.0023		0.0100
100	500	00:03:30	100.00%	98.37%	0.0019	0.0552	0.0100

Accuracy 98.37398

High accuracy on mini-batches and validation data. The loss on both mini-batches and validation continues to decrease. The learning rate remains constant at 0.0100.

6. Conclusion

Accuracy The model with data size 100*10 achieves higher accuracy (100% on mini-batches) compared to the model with data size 15*10. The validation accuracy is consistently high in both cases. Loss: Both models show a decreasing trend in both mini-batch and validation losses. The loss on the larger dataset tends to be lower. Learning Rate: The learning rate remains constant at 0.0100 in both cases. Time Elapsed: The time elapsed for each epoch is slightly longer for the larger dataset, which is expected due to the increased data size. The larger dataset (100*10) allows the model to achieve higher accuracy, indicating better generalization.

The loss is lower for the larger dataset, suggesting that the model is learning more complex patterns. The learning rate is the same for both, indicating a consistent training approach. The larger dataset appears to contribute to better model performance, showcasing higher accuracy and lower loss. However, it's important to consider the computational resources and time required for training larger datasets.

Abbreviations

AI	Artificial Intelligences
ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
RFC	Random Forest Classifier
RNN	Recurrent Neural Networks
SVM	Support Vector Machines

Acknowledgments

I would like to express gratitude for any assistance and support my students at Sanothimi Campus to help create datasets. Also, I am grateful to Tribhuvan University and Nepal Open University for their help in providing guidance and contributing to and encouraging me to continue my research. I would like to thank my family and wife for supporting each part of the involvement. Thankful for guidance from mentors, and assistance from Dr. Bhojraj Ghimire and the entire Nepal Open University faculty.

Author Contributions

Nawaraj Ghimire is the sole author. The author read and approved the final manuscript.

Funding

This research did not receive any financial support.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] A. Siddiqa and C. D S, "A Recognition System for Handwritten Digits Using CNN," International Journal of Science and Research (IJSR), 2020.

[2] "DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition," Multimedia Tools and Applications, 08 03 2021.

[3] S. Singh, N. K. Garg, and M. Kumar, “Feature extraction and classification techniques for handwritten Devanagari text recognition: a survey.,” *Multimed Tools Appl*, vol. 82, no. 1, pp. 747-747–775, 2023, https://doi.org/10.1007/s11042-022-13318-9

[4] S. R. Narang, M. K. Jindal, and M. Kumar, “Ancient text recognition: a review.,” *Artif Intell Rev*, vol. 53, no. 8, pp. 5517-5517–5558, 2020, https://doi.org/10.1007/s10462-020-09827-4

[5] S. P. Deore and P. Albert, "Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset," Sadhana, vol. 45, no. 243, 17 08 2020.

[6] R. Sethi and I. Kaushik, "Hand Written Digit Recognition using Machine Learning," 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), pp. 49-54, 2020.

[7] G. Deng, M. Tang, Y. Zhang, Y. Huang, and X. Duan, "Privacy-Preserving Outsourced Artificial Neural Network Training for Secure Image Classification.," *Applied Sciences (2076-3417)*, vol. 12, no. 24, pp. 12873-12873–12887, 2022, https://doi.org/10.3390/app122412873

[8] F. Siddique, S. Sakib and M. A. B. Siddique, "Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers," 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), pp. 541-546, 2020.

[9] R. Geetha, T. Thilagam and T. Padmavathy, "Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks.," Neural Computing and Applications, 19 11 2020.

[10] Lindsay, G. W. (2022). *Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future.* Gatsby Computational Unit/Sainsbury Wellcome Centre.

[11] S. Sabbu and V. Ganesan, "LSTM-Based Neural Network to Recognize Human Activities Using Deep Learning Techniques.," *Applied Computational Intelligence & Soft Computing*, pp. 1-1–8, 2022, https://doi.org/10.1155/2022/1681096

[12] F. Kratbi, A. Y. and H. S., "Support Vector Machines for Evaluating the," Original scientific paper Impact of the Forward Osmosis Membrane Characteristics on the Rejection of the Organic Molecules, vol. 72, no. 7, p. 417−431, 2023.

[13] J. Zhou, H. Xu, Z. Zhang, J. Lu, W. Guo, and Z. Li, "Using Recurrent Neural Network Structure and Multi-Head Attention with Convolution for Fraudulent Phone Text Recognition.," *Computer Systems Science & Engineering*, vol. 46, no. 2, pp. 2277-2277–2297, 2023, https://doi.org/10.32604/csse.2023.036419

[14] C. Wu, W. Fan, Y. He, J. Sun and S. Naoi, "Handwritten Character Recognition by Alternately Trained Relaxation Convolutional Neural Network," 2014 14th International Conference on Frontiers in Handwriting Recognition, 2014.

[15] S. Ahlawat, A. Choudhary, N. Anand, S. Singh and B. Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," Sensors, vol. 20, no. 12, 2020.

[16] C.-L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," Pattern Recognition, vol. 37, no. 2.

[17] S. Dua *et al.*, "Developing a Speech Recognition System for Recognizing Tonal Speech Signals Using a Convolutional Neural Network.," *Applied Sciences (2076-3417)*, vol. 12, no. 12, pp. 6223-6223–6235, 2022, https://doi.org/10.3390/app12126223

[18] P. Sharma and J. Rai, "A Review of Feature Extraction Methods for Handwritten Character Recognition," Journal of Advances and Scholarly Researches in Allied Education, vol. 17, no. 2, 2020.

[19] D. Mungra, A. Agrawal, P. Sharma, S. Tanwar, and M. S. Obaidat, "PRATIT: a CNN-based emotion recognition system using histogram equalization and data augmentation.," *Multimed Tools Appl*, vol. 79, no. 3/4, pp. 2285-2285–2307, 2020, https://doi.org/10.1007/s11042-019-08397-0

[20] A. Sen, T. K. Mishra, and R. Dash, "A novel hand gesture detection and recognition system based on ensemble-based convolutional neural network.," *Multimed Tools Appl*, vol. 81, no. 28, pp. 40043-40043–40066, 2022, https://doi.org/10.1007/s11042-022-11909-0

[21] H. Singh, R. K. Sharma, and V. P. Singh, "Online handwriting recognition systems for Indic and non-Indic scripts: a review.," *Artif Intell Rev*, vol. 54, no. 2, pp. 1525-1525–1579, 2021, https://doi.org/10.1007/s10462-020-09886-7

[22] M. Á. Serrano, A. Flammini, and F. Menczer, "Modeling Statistical Properties of Written Text.," *PLoS One*, vol. 4, no. 4, pp. 1-1–8, 2009, https://doi.org/10.1371/journal.pone.0005372

[23] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning Pooling for Convolutional Neural Network," Neurocomputing, vol. 224. Elsevier BV, pp. 96–104, Feb. 2017. https://doi.org/10.1016/j.neucom.2016.10.049

Biography

Nawaraj Ghimire is lecturer of Tribhuvan university Nepal. He works in the Sanothimi Campus as head of the Department of ICT Education. He acquired MED in ICT and MPhil scholar at Nepal Open University, Nepal. He has participated in multiple international research collaboration projects in recent years.

Research Field

Nawaraj Ghimire: ICT Education, Technology in Education, Educational Technology, Computer Science and Education.