STM JOURNALS

Current Trends in
Signal Processing

ISSN: 2277-6176 (Online)
ISSN: 2321-4252 (Print)
Volume 15, Issue 2, 2025
May–August
DOI (Journal): 10.37591/CTSP

https://journals.stmjournals.com/ctsp

Research CTSP

Handwritten Sanskrit Word Recognition: A Deep Learning Approach Using AlexNet

Shraddha V. Shelke[1,*], D.M. Chandwadkar[2],
S.P. Ugale[3], Rupali V. Chothe[4]

Abstract

Handwritten Sanskrit word recognition poses significant challenges due to the intricate structure of the script and the considerable variations in handwriting across individuals. To address these challenges, this research introduces a novel methodology employing transfer learning with the AlexNet convolutional neural network. The study utilized two distinct datasets: a specifically curated Sanskrit word image dataset containing 2616 samples, alongside a broader Devanagari character dataset used for validation purposes. The established AlexNet architecture underwent a process of fine-tuning, primarily focused on the Sanskrit word dataset. Key hyperparameters, such as the learning rate and the number of training epochs, were carefully optimized to maximize the model's performance. The proposed approach demonstrated promising outcomes, achieving results that surpassed existing state-of-the-art techniques on both datasets. Notably, with a training duration of 20 epochs and an optimized learning rate of 0.015, the model achieved a peak testing accuracy of 97.9%. To enhance the model's training efficiency, the Stochastic Gradient Descent with Momentum (SGDM) optimizer was strategically implemented, effectively accelerating the convergence process and mitigating oscillations during learning. This research highlights the effectiveness of leveraging transfer learning with deep convolutional neural networks for the complex task of handwritten Sanskrit word recognition.

Keywords: Sanskrit word recognition, Alexnet, convolutional neural network, stochastic gradient descent with Momentum

***Author for Correspondence**
Shraddha V. Shelke
E-mail: svshelke@kkwagh.edu.in

[1]Research Scholar, Department of Electronics and Telecommunication, Karmaveer Kakasaheb Wagh Institute of Engineering Education & Research, Nashik, Savitribai Phule Pune University, Pune, Maharashtra, India
[2]Professor and Head, Department of Electronics and Telecommunication, Karmaveer Kakasaheb Wagh Institute of Engineering Education & Research, Nashik, Maharashtra, India
[3]Professor, Department of Electronics and Telecommunication, Karmaveer Kakasaheb Wagh Institute of Engineering Education & Research (KKWIEE & R), Nashik, Maharashtra, India
[4]Assistant Professor, Department of Electronics and Telecommunication, Karmaveer Kakasaheb Wagh Institute of Engineering Education & Research (KKWIEE & R), Nashik, Maharashtra, India

Received Date: May 14, 2025
Accepted Date: May 16, 2025
Published Date: June 25, 2025

Citation: Shraddha V. Shelke, D.M. Chandwadkar, S.P. Ugale, Rupali V. Chothe. Handwritten Sanskrit Word Recognition: A Deep Learning Approach Using AlexNet. Current Trends in Signal Processing. 2025; 15(2): 33–43p.

INTRODUCTION

Handwritten character recognition (HCR) has been a significant research area for decades, with applications ranging from document digitization to automated form processing. However, the complex nature of the characters and the variation in handwriting styles make it particularly difficult to recognise scripts like Sanskrit. Traditional HCR methods, such as template matching and Hidden Markov Models, have limitations in handling complex scripts and often require extensive feature engineering. Convolutional Neural Networks have given better results and found very effective for recognition of handwritten scripts. By learning discriminative features, CNNs can effectively distinguish between different characters, even in the presence of noise, variations in handwriting styles, and other challenging conditions. This leads to robust and accurate character recognition systems [1]. Transfer learning technique is used in this work which allow us use patterns or information gained by the pretrained network on large dataset to

© STM Journals 2025. All Rights Reserved

classify our dataset. Study shows that using transfer learning approach, deep learning models give best results even if limited amount of training data is available, because these models are already pretrained on large amount of dataset. By fine-tuning a CNN that has already been trained on a particular job, the model may adjust to the particulars of the new data. Handwritten Sanskrit word recognition poses additional challenges due to the complexity of the script, which includes touching characters, a large number of consonants and vowels, the presence of *shirolekha* (a horizontal line at the top of characters), and joint or compound characters. These factors can significantly impact the accuracy of recognition systems [2].

In this research, we propose a novel approach to handwritten Sanskrit word recognition that combines transfer learning with the AlexNet architecture. AlexNet, a deep CNN originally designed for image classification, has proven effective in various computer vision tasks. By leveraging the powerful feature extraction capabilities of AlexNet and fine-tuning it on a large dataset of handwritten Sanskrit words, we aim to achieve superior accuracy and performance, addressing the challenges posed by the complex nature of the Sanskrit script.

LITERATURE REVIEW

Gurav *et al.* developed an offline Deep Convolutional Neural Network (DCNN) model using a dataset of 300 standardized 28×28 pixel images. After 15 training iterations, the model achieved a 99.65% accuracy. The sequential convolutional layers in the architecture effectively extract higher-level features. To mitigate overfitting, a dropout layer was incorporated. However, the confusion matrix revealed potential misclassifications due to similarities between certain characters [3].

Due to the complexities of the Devanagari script, recognizing handwritten Sanskrit characters poses a significant challenge. A study by Shelke *et al*. explored this by developing a new algorithm that utilizes four distinct feature extraction methods like edge detection algorithms, combination of wavelet and cosine transform etc. For classification, both Support Vector Machines (SVM) and Neural Networks were evaluated. In all feature extraction scenarios, SVM achieved superior performance compared to Neural Networks. Furthermore, HOG-based features delivered the highest accuracy for SVM, reaching 97.10% [4].

Lomte and Doye explored three different CNN architectures and the AlexNet model to recognize Sanskrit words from a 7000-word dataset. Their model demonstrated impressive performance, achieving a 97.42% accuracy with a rapid recognition time of 0.364 msec. When compared to other methods on the same dataset, their approach proved superior. However, the model's accuracy was compromised when dealing with blurred or overlapping text images [5].

Aneja and Aneja investigated the potential of already trained models on large datasets (AlexNet, DenseNet, VGG, and InceptionV3) for recognizing handwritten Devanagari characters. Through transfer learning with a DCNN, these models were adapted to the Devanagari character dataset. InceptionV3 proved to be the most effective model, achieving a 99% accuracy rate, although it required a longer training time of 16.3 min per epoch. While AlexNet was significantly faster, training in 2.2 min per epoch, it exhibited a slightly lower accuracy of 98% [6].

Gupta *et al*. developed a novel algorithm which was trained on a substantial dataset comprising 378,951 characters. To boost the model's performance, data augmentation techniques were implemented to generate supplementary training images. Convolutional and pooling layers were among the six layers in the model design, which were followed by fully linked layers. This method produced a 95.6% accuracy rate. Despite the benefits of a large dataset, there is a significant risk of overfitting. In this scenario, the model may become overly reliant on the specific characteristics and patterns present in the training data [7].

© STM Journals 2025. All Rights Reserved

METHODOLOGY

Dataset

A diverse dataset comprising 2616 handwritten Sanskrit word images was meticulously collected. The dataset encompassed variations in handwriting styles, pen widths, and age groups, ensuring a robust training set. Data augmentation methods like rotation, scaling, and noise addition were used to further enhance the dataset and the model's capacity for generalisation. Devanagari Handwritten character dataset available freely for research includes (100 images of 36 characters) 3600 images collected by Deore and Pravin are also used for validation purpose [8]. Figures 1 and 2 show few word images of Handwritten words and handwritten characters (SD-dataset).

The powerful AlexNet architecture was employed as the backbone for the proposed model [8]. This pre-trained model, originally designed for image classification, was leveraged through transfer learning. By fine-tuning the final layers of AlexNet on the Sanskrit word dataset, the model was adapted to the specific task of recognizing handwritten Sanskrit words. The model's parameters were optimised using the SGDM optimiser, a stochastic gradient descent variation. This optimizer accelerates convergence and reduces oscillations during training, leading to faster and more stable learning.

The model's performance was evaluated by plotting training and validation loss curves. These curves provide insights into the model's training progress and help identify potential overfitting or underfitting. To further illustrate the model's classification accuracy for every class, a confusion matrix was created.

CNN Architecture

Another form of deep learning architecture created especially for image identification applications is the Convolutional Neural Network (CNN). It consists of multiple layers, each with a distinct function:

Convolutional Layers

Convolutional layer is first layer in CNN which includes various filters. There filters are nothing but moving windows, which move over input image to extract features like edges, corners etc. If filter size is f×f and image size is N×N then output convolutional feature map is having size of (N–f+1)×(N–f+1). One can use multiple filters with different filter sizes. Stride is nothing but number of pixels by which filter can move horizontally or vertically. Convolutional layer includes convolutional operation between input image and filter with addition of bias function, which can be ReLU, sigmoid or softmax etc.

Pooling Layers

This layer is included to minimize dimensions of feature map after convolutional layer along with enhancing the image. Max pooling selects maximum pixel value of a U×U window whereas average pooling selects average value of pixels.

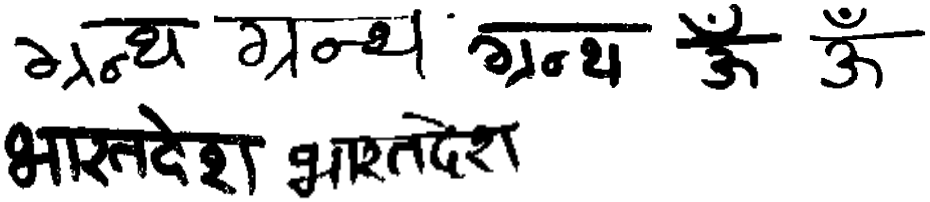

Figure 1. Images of handwritten words.

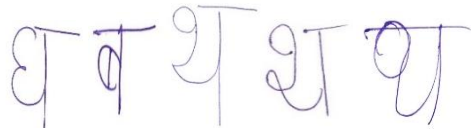

Figure 2. Images of handwritten characters SD dataset.

© STM Journals 2025. All Rights Reserved

Fully Connected Layers

The pooling layer's output is applied to the fully connected layer after first being flattened. These layers connect each neuron in the present layer to every other neuron in the layer above, making them nothing more than standard neural network layers.

CNNs may learn more complicated characteristics from the input data by adding multiple convolutional and pooling layers. This hierarchical representation of features allows CNNs to effectively recognize patterns and classify images with high accuracy [9, 10].

Transfer Learning

A machine learning technique known as transfer learning makes use of the data gathered from one task to improve performance on a related one.

By transferring pre-trained models to a new task, we can significantly reduce training time and improve accuracy, especially when limited data is available. In deep learning, this method works especially well since it allows pre-trained models from big datasets, like ImageNet, to be improved on smaller, specific to the domain datasets.

AlexNet Architecture

AlexNet, a groundbreaking neural network model, revolutionized the field of computer vision. Eight layers make up this system: two fully connected layers, five convolutional layers, and a final softmax layer for classification. By using 96, 11×11 filters with a stride of 4, the first convolutional layer greatly reduces the size of the input image. In the second convolutional layer, 256 5×5 filters with a stride of 1 are used. 384, 384, and 256 3×3 filters with a stride of 1 are used by the third, fourth, and fifth convolutional layers, respectively [6].

To add non-linearity, ReLU activation functions are applied after every convolutional layer. The first, second, and fifth convolutional layers are followed by max pooling layers, which add translational invariance and decrease spatial dimensions.

A softmax layer for classification comes after the fully linked layers, each of which has 4096 neurons. Figure 3 shows AlexNet model implemented in MATLAB.

Stochastic Gradient Descent with Momentum (SGDM)

One popular optimisation approach for deep neural network training is Stochastic Gradient Descent with Momentum (SGDM). It combines the benefits of momentum with stochastic gradient descent to accelerate convergence and reduce oscillations [11].

The momentum term, denoted by β, accumulates the gradients from previous iterations, providing a sense of direction. This helps the optimizer to overcome local minima and accelerate convergence. The update rule for SGDM is given by Eqs. (1) and (2).

$$v\_t=\beta\times v\_(t-1)+(1-\beta)\times\nabla\theta\_t \ldots\ldots \quad (1)$$

$$\theta\_t=\theta\_(t-1)-\alpha\times v\_t\ldots\ldots\ldots\ldots\ldots \quad (2)$$

where:

- v_t: Velocity vector at time step t,
- β: Momentum parameter,
- ∇θ_t: At time step t, the gradient of the loss function with respect to the parameters θ,
- α: Learning rate, and
- θ_t: Model parameters at time step t.

By incorporating momentum, SGDM can effectively navigate the parameter space and find optimal solutions more efficiently [12].

© STM Journals 2025. All Rights Reserved

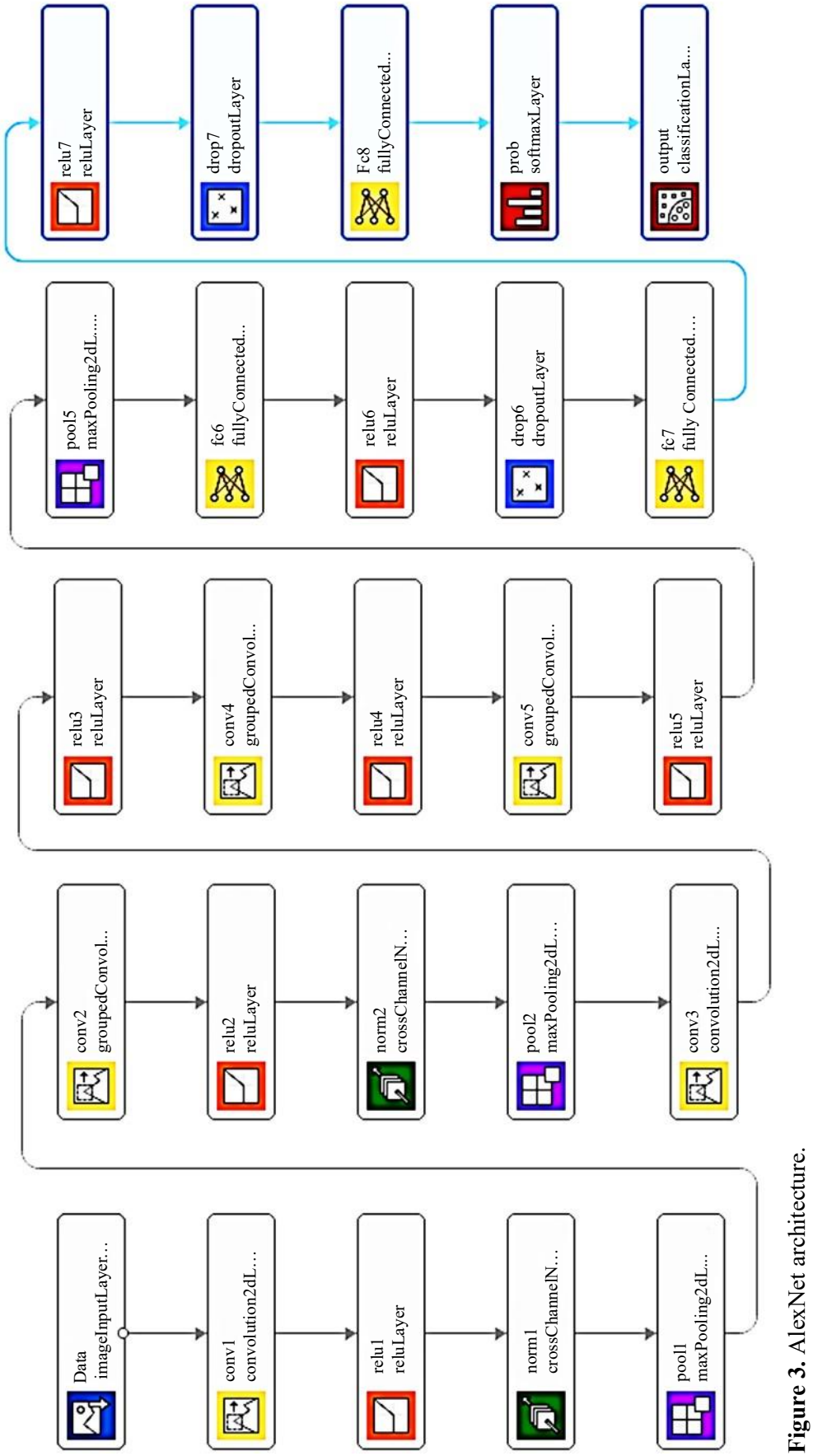

Figure 3. AlexNet architecture.

© STM Journals 2025. All Rights Reserved

RESULTS AND DISCUSSION

The model was trained using the SGDM optimizer, which enhances convergence efficiency by incorporating momentum. To maximise performance, hyperparameters like learning rate and epoch count were changed. The model was trained for 10 and 20 epochs, with learning rates of 0.001, 0.01, and 0.015. Results obtained for 10 EPOCHS are shown in Table 1 and for 20 EPOCHS are shown in Table 2.

It is observed from the Table 1 that learning rate of 0.01 resulted in maximum classification accuracy of 97.51% with loss of 0.0345. The model is trained on CPU which required 1.031 min time per epoch. This time can be further reduced by training on GPU.

The model performed best when learning at a rate of 0.01 and 20 epochs, and it demonstrated good accuracy. The result of this combination produced a 97.32% testing accuracy and a 100% training accuracy. The high testing accuracy shows that the model can generalize to new data.

The use of the SGDM optimizer further accelerated the training process and improved the model's convergence. By incorporating momentum, SGDM helps the optimizer to overcome local minima and find the global optimum more efficiently.

The confusion matrix further corroborates the model's performance, highlighting the areas where the model may struggle to correctly classify certain word classes.

Figures 4–6 display training and validation loss curves that provide information about the model's training procedure. From these graphs we can observe that learning rate of 0.01 is the optimum learning rate. The curves show a steady decrease in loss over epochs, indicating effective learning and convergence.

Although it helps speed up training, an increased learning rate can cause instability and overshoot the ideal response. A reduced learning rate, on the other hand, can stabilise training but might cause slower convergence. A learning rate of 0.01 was determined to offer a suitable compromise between stability and convergence speed in this experiment.

Overall, the results demonstrate the effectiveness of the proposed approach for handwritten Sanskrit word recognition. The model's high accuracy and good generalisation to new data demonstrate the promise of deep learning and transfer learning approaches for this difficult task.

Table 1. Results obtained for 10 Epochs.

Learning Rate	0.001	**0.01**	0.015
Training accuracy	82.03	**98.44**	97.66
Testing Accuracy	92.73	**97.51**	95.75
Loss	0.522	**0.0345**	0.0345
Time/epoch (min)	0.92	**1.031**	0.95

Table 2. Results obtained for 20 Epochs.

Learning Rate	0.001	**0.01**	0.015
Training accuracy	97.66	**100**	100
Testing Accuracy	95.41	**97.32**	97.9
Loss	0.0785	**0.0007**	0.0098
Time/epoch (min)	0.96	**1.01**	0.93

© STM Journals 2025. All Rights Reserved

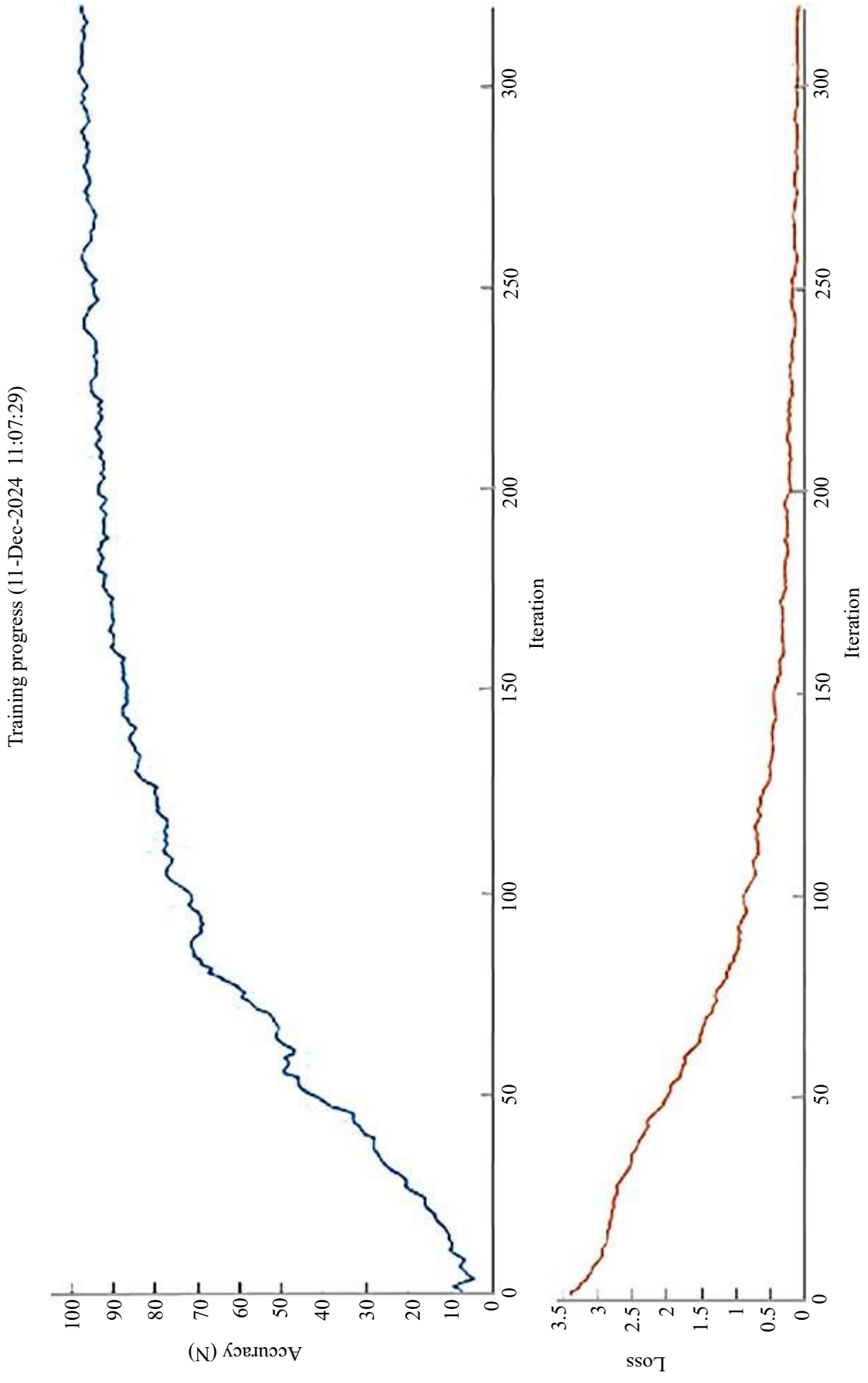

Figure 4. Validation and loss curve for learning rate of 0.001 and #Epochs=20.

© STM Journals 2025. All Rights Reserved

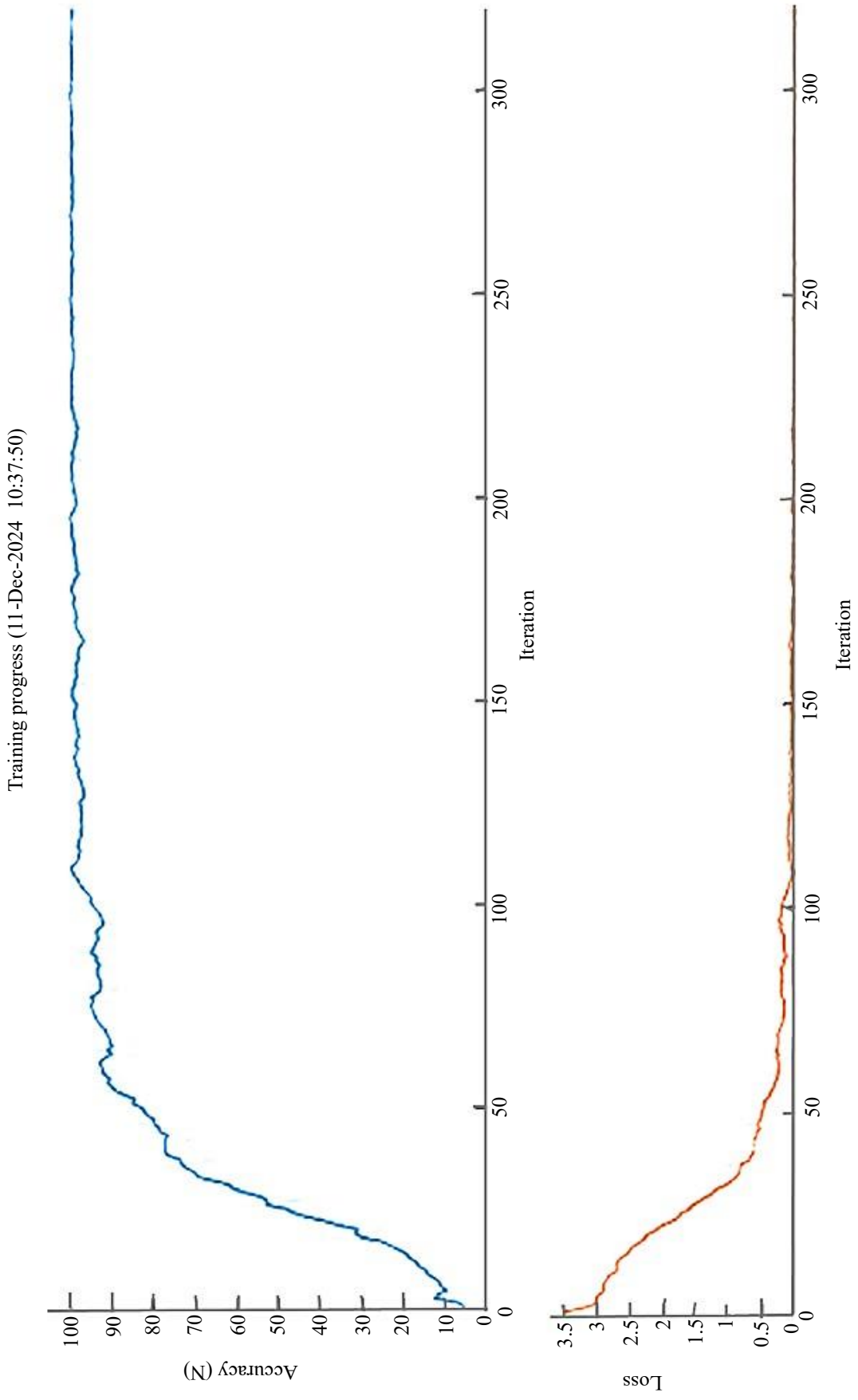

Figure 5. Validation and loss curve for learning rate of 0.01 and #Epochs=20.

© STM Journals 2025. All Rights Reserved

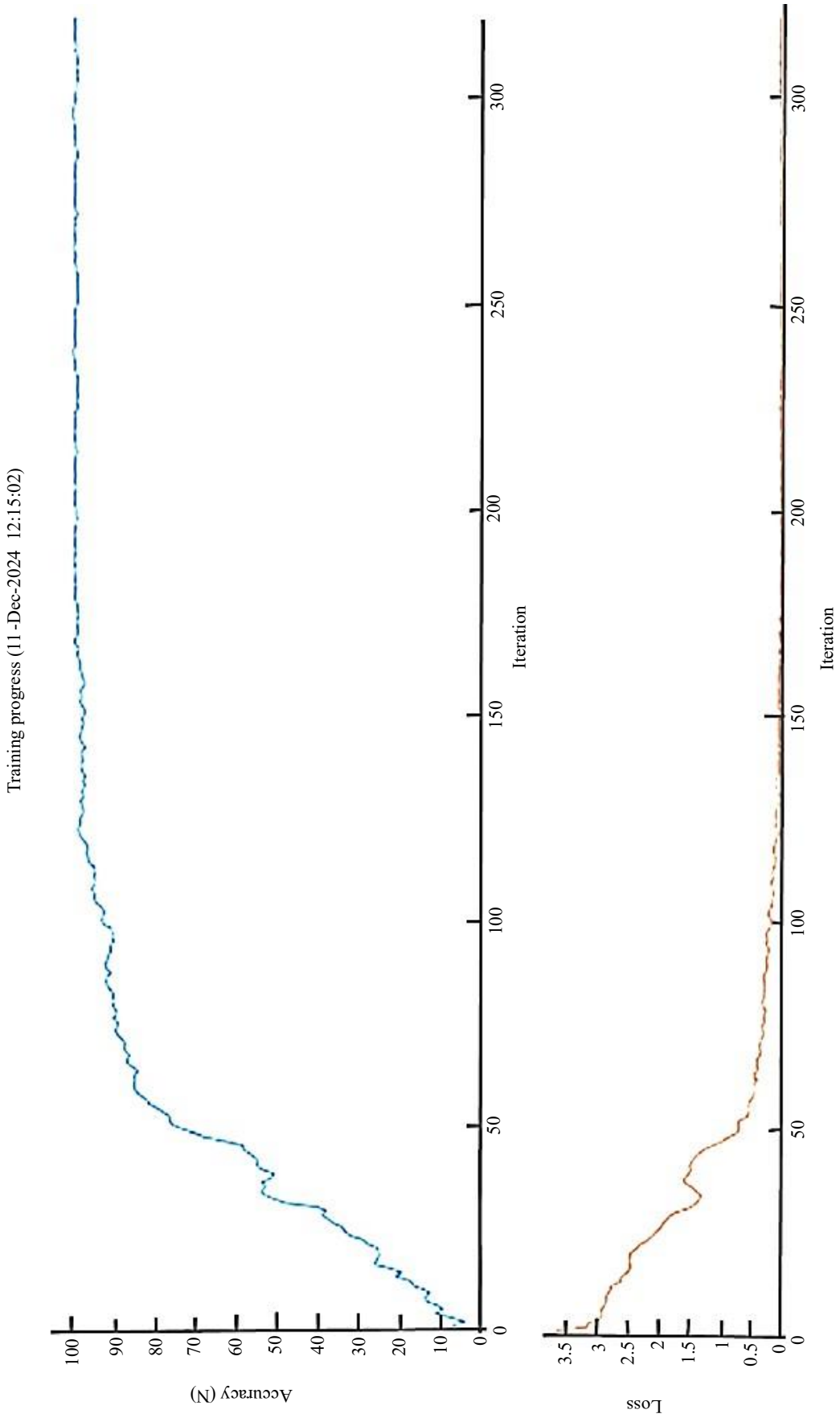

Figure 6. Validation and loss curve for learning rate of 0.015 and #Epochs=20.

© STM Journals 2025. All Rights Reserved

Table 3. Validation of results.

Author	Deore *et al*. [8]	Proposed Model
Dataset	SD	SD
Algorithms	First VGG-16 model with RMS Prop Optimizer Second fine-tuned VGG16 Model with RMSprop optimizer	Fine tunned AlexNet with SGDM optimizer
% Testing Accuracy	94.84% 96.55%	**95.12%** **96.67%**
Loss	0.18 0.12	0.073 0.045
Epochs	20, 10	20, 10

Validation of Results

The model that we propose in this study outperforms the two-stage fine-tuned VGG16 model proposed by Deore and Pravin after testing it on additional benchmark dataset produced [8]. With a reduced loss of 0.0169, the suggested GoogLeNet model with SGDM optimizer gives testing accuracy of 96.81%. Validation results are given in Table 3.

CONCLUSION

In this research, we successfully implemented a deep learning-based approach for handwritten Sanskrit word recognition using the AlexNet architecture. The model was trained on a diverse dataset of 2616 Sanskrit words, leveraging transfer learning to initialize the network with pre-trained weights. The SGDM optimizer was employed to efficiently optimize the model parameters.

After training the model several times, we determined the optimal hyperparameters, including learning rate and number of epochs, which significantly impacted the model's performance. The optimal outcomes, which balanced accuracy and convergence speed, were attained with a learning rate of 0.01 and 20 epochs. The experimental findings show how successful the suggested strategy is. The model achieved a high accuracy of 97.32% on the test dataset, outperforming existing methods. The analysis of the confusion matrix provides insights into the model's strengths and weaknesses.

Results are validated after training the proposed fine tunned model for standard SD dataset of handwritten Devanagari characters. Our fine tunned model gave classification accuracy of 96.67% with loss of 0.045 for 10 epochs.

REFERENCES

1. Moudgil A, Singh S, Gautam V, Rani S, Shah SH. Handwritten Devanagari manuscript characters recognition using capsnet. Int J Cogn Comput Eng. 2023 Jun 1; 4: 47–54.
2. More V, Kharat M, Gumaste S. Segmentation of devanagari handwritten text using thresholding approach. Int J Sci Technol Res. 2020 Mar; 9(3): 2277–8616.
3. Gurav Y, Bhagat P, Jadhav R, Sinha S. Devanagari handwritten character recognition using convolutional neural networks. In 2020 IEEE International Conference on Electrical, Communication, and Computer Engineering (ICECCE). 2020 Jun 12; 1–6.
4. Shelke SV, Chandwadkar DM, Ugale SP, Chothe RV. Combining Multiple Feature Extraction and Classification Methods to Study Performance of Handwritten Sanskrit Character Recognition. In 2023 IEEE 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA). 2023 Aug 18; 1–6.
5. Lomte MV, Doye DD. Handwritten Vedic Sanskrit text recognition using deep learning. J Algebr Stat. 2022 Jul 12; 13(3): 2190–8.
6. Aneja N, Aneja S. Transfer learning using CNN for handwritten Devanagari character recognition. In 2019 IEEE 1st International Conference on Advances in Information Technology (ICAIT). 2019 Jul 25; 293–296.

© STM Journals 2025. All Rights Reserved

7. Gupta P, Deshmukh S, Pandey S, Tonge K, Urkunde V, Kide S. Convolutional neural network based handwritten Devanagari character recognition. In 2020 IEEE International conference on smart technologies in computing, electrical and electronics (ICSTCEE). 2020 Oct 9; 322–326.
8. Deore SP, Pravin A. Devanagari handwritten character recognition using fine-tuned deep convolutional neural network on trivial dataset. Sādhanā. 2020 Dec; 45(1): 243.
9. Shingate MD, Gumaste SV. Improving Ensemble Classifier for Natural Language Situational Information Analysis. In 2024 IEEE International Conference on Emerging Innovations and Advanced Computing (INNOCOMP). 2024 May 25; 305–311.
10. Shelke SV, Chandwadkar DM. Automatic system for recognition of handwritten character using multiscale neural network. In: Proceedings of the International Conference in Computational Intelligence (ICCIA); 2012 Mar;6:16–20.
11. Bisht M, Gupta R. Multiclass recognition of offline handwritten Devanagari characters using CNN. Int J Math Eng Manag Sci. 2020; 5(6): 1429–39.
12. Avadesh M, Goyal N. Optical character recognition for Sanskrit using convolution neural networks. In 2018 13th IAPR international workshop on document analysis systems (DAS). 2018 Apr 24; 447–452.

© STM Journals 2025. All Rights Reserved