Discrete wavelet transform and convolutional neural network based handwritten Sanskrit character recognition

Shraddha V. Shelke, Dinesh M. Chandwadkar, Sunita P. Ugale, Rupali V. Chothe
Department of Electronics and Telecommunication Engineering, K. K. Wagh Institute of Engineering Education and Research, Savitribai Phule Pune University, Pune, India

Article Info

Article history:

Received Jul 18, 2024
Revised Aug 28, 2024
Accepted Nov 24, 2024

Keywords:

Convolutional natural network
GoogLeNet
Discrete wavelet transform
National mission for manuscripts
Sanskrit manuscripts

ABSTRACT

Sanskrit is one of the ancient languages from which the majority of present Indian languages are developed. Although the national mission for manuscripts (NMM) is digitizing handwritten Sanskrit manuscripts, there are still a lot of papers that need to be digitized. Recognition of handwritten script is a challenging task due to individual differences in writing styles and how those variations alter over time. The Sanskrit language is written in Devanagari script. A novel approach using discrete wavelet transform (DWT) and convolutional natural network (CNN) is proposed in this paper. Devanagari handwritten character dataset which includes 2000 handwritten images of 36 classes (2000*36=72000) is used in this research. Fine-tuned GoogLeNet model implemented here gave optimum values of epochs and learning rate of 15 and 0.01 respectively. Classification accuracy obtained by proposed DWT – CNN model is 98.97% with a loss of 0.098. Fine-tuned GoogLeNet model achieves 99.68% accuracy with a 0.0635 loss. Results obtained are also compared with existing approaches and found superior.

Corresponding Author:

Shraddha V. Shelke
Department of Electronics and Telecommunication Engineering, K. K. Wagh Institute of Engineering Education and Research, Savitribai Phule Pune University
Pune, India
Email: svshelke@kkwagh.edu.in

1. INTRODUCTION

Ancient Sanskrit texts include an abundance of knowledge on Hindu mythology, culture, science, mathematics and Indian history. Around 300 million individuals use Sanskrit script globally [1]. There is a lot of research being done in the field of recognizing handwritten text [2]. There are numerous official languages in India that are written in Devanagari script, such as Hindi, Marathi, Sindhi, Nepali, Sanskrit, and Konkani. A wide range of algorithms have been created for recognition of scripts written in several Indian languages, including Hindi, Bangla, Telugu, and Gurmukhi. But still there has been not enough study conducted on developing an efficient Sanskrit script recognition system, hence emphasizing the need to construct one [3]. A human can easily understand contents written on paper, but a machine finds it challenging to recognize handwritten text due to verity of writing styles.

For handwritten text identification, the researchers have used a variety of machine learning, deep learning techniques [4]. The initial and crucial step in an optical character recognition system is feature extraction from the input image following preprocessing and noise reduction. The classifier is then trained by feeding the extracted features to several classifiers. In the testing and validation phase, the output of the classifier is verified by testing its performance on new data [5]. The initial stages of a convolutional neural network (CNN) based method involve applying various edge detection filters to input images in order to extract feature maps. The input image is convolved with filters to extract features, which are then applied to the fully connected layer after pooling operation. To improve a model's performance, several convolutional, pooling, and fully connected layers are used. After applying activation functions for multiclass classification, such as rectified linear units (ReLU) and SoftMax, classified output is generated [6].

Sanskrit script is distinguished by the header line, or shirolekha, that appears above each character. Compared to other scripts like English, more consonants and vowels and different combinations of compound letters and conjunct consonants makes Sanskrit script recognition a challenging task. There are 16 vowels and 39 consonants in Sanskrit writing. Consonants can sometimes have identical shapes with just minor variations, such as dots, slanting lines, and curves [7] as shown in Figure 1.

Figure 1. Consonants with similar shape with minor variations

One of the challenge in recognition is different handwriting style of different people which changes over age. Other important parameters to be considered while developing recognition system are cursive style of writing, slope, thickness, average size of letters, spacing between letters and input image quality [8]. A system to recognize handwritten Devanagari characters is proposed by Deore and Pravin [9] where authors have implemented two stage approach using VGG16 to recognize Devanagari handwritten characters. RMSprop and Adam optimizers are used which resulted in good accuracy with RMSprop with faster convergence than Adam. In order to prevent over fitting, approaches used here are data augmentation and inclusion of 0.5 dropout. Results are obtained by changing number of epochs. Model is also trained by using VGG 19 which gives less accuracy and needs more storage space than than VGG16, which archives in classification accuracy of 96.55%.

An approach using CNN to recognize handwritten Devanagari characters is implemented by Bhardwaj [10]. Model A and Model B differ in number of convolutional layers, fully connected layers. In both models ReLU activation function and local response normalization is used. With a momentum of 0.9, stochastic gradient descent (SGD) is used to accomplish the backpropagation on feed-forward nets. Model A is able to give classification rate of 0.98471. Authors found improvement in classification accuracy up to 0.981326 by inclusion of dropout layer and extending dataset.

Aneja and Aneja [11] have proposed an approach using transfer learning to classify Devanagari characters. Fine tuning and ConvNet for fixed feature extractor are techniques used. In case of fixed feature extractor abr obtained from weights of convolutional layers which are frozen and weight vectors of fully connected layers are updated. Results shows that inception V3 model resulted in highest accuracy of 99%. Vgg11 have higher computational cost than inception V3 with 99% accuracy with average time of 5.7 min per epoch. Classifiaction accuracy of DenseNet 201 and DenseNet 121 is 89% and 90% respectively.

Lomte and Doye [12] have proposed an effective method for calling up Devanagari text and calligraphy recognition that combines wavelet, contour, and support vector machine using correlation functions ICF and ACF resulted accuracy of 98.98 %. Vina *et al*. [13] implemented modified CNN along with Alexnet to recognize handwritten Vedic Sanskrit text. Three CNN models with 3, 8 and 12 CONV layers respectively are proposed. It is observed that combining the Adam optimizer with ReLU allows for faster processing by increasing the training rate and preventing overfitting of the model. Model M2 found best with accuracy of 97.42% for Vedic Sanskrit. Authors have also proposed a novel system to recognize gesture-based text written in Devanagari. Two CNN models with 2 and 3 layers are used to recognize hand tip gesture character with accuracy of 97.6 % and 98.4% respectively [14].

Manocha and Tewari [15] proposed a system using CNN for feature extraction and support vector machine, random forest, decision tree, multilayer perceptrons and K-Nearest neighbor and extreme boost gradient for classification of DHCD dataset. Training and testing dataset is splitted in cases of 70:30, 75:25 and 80:20. Highest classification accuracy is obtained as 99% by using CNN and SVM-RBF.

In previous work several feature extraction methods and classifiers are used in a machine learning-based strategy for recognizing handwritten Devanagari characters through analyzing their performance [16]. In this work a novel approach using discrete wavelet transform (DWT) and CNN is employed to recognize handwritten characters. The fine-tuned GoogLeNet model undergoes multiple training cycles to determine the optimum epoch and learning rate.

2. METHOD

Architecture of proposed system is shown in Figure 2. Sankrit is written using Devanagari script, dataset including 92,000 handwritten images prepared by Acharya *et al.* [17] are used in this experiment. Dataset includes 2,000 handwritten images of 46 different classes of 36 consonants and 10 numerals. 72,000 images of 36 classes are used in this experiment. The image data store of all 72,000 images is created. The preprocessing step involves noise removal and resizing of the image. Further discrete wavelet transform is applied for feature extraction which is then applied to CNN model to generate classification results.

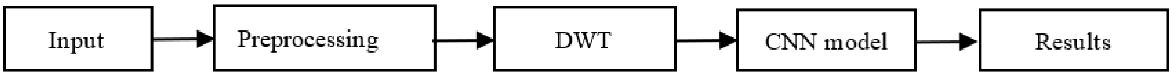

Figure 2. DWT CNN based system architecture

2.1. Discrete wavelet transform

A technique for extracting features that divides an input signal into several sub bands by applying a discrete wavelet transform to the input image. Analyzing signal data in both the time and frequency domains is one of the best features of DWT [18]. Figure 3 describe wavelet decomposition of the input character image in four sub bands. cA is approximation coefficient which is low frequency sub band. And three high frequency sub bands include horizontal detail coefficient (cH), vertical detail coefficient (cV), and diagonal detail coefficient (cD) [19]. After preprocessing and input image (I) is applied to discrete wavelet transform generates I={cA,cV,cH,cD}. The detail coefficients obtained from applying DWT to an input image of dimensions 128x128 are of size 64×64, and the resulting approximation coefficient cA undergoes CNN processing.

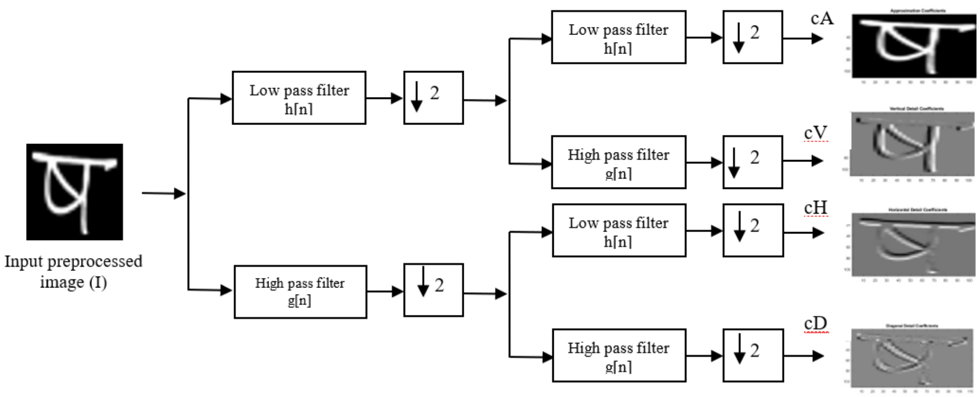

Figure 3. Feature extraction process by DWT

2.2. CNN architecture

Traditionally, an artificial neural network (ANN) is made up of many dense layers of neurons, often referred to as fully connected layers. Using these fully connected layers results in an extremely large total number of parameters [6]. Using multiple convolutional and pooling layers in CNN allows us to reduce the number of parameters because these layers only require a small number of parameters. The feature patterns of the input dataset are stored in feature maps that are generated as a result of convolutional and pooling layers [5]. Figure 4 shows the structure of the proposed CNN model.

Convolutional layer in CNN consists of applying convolutional operation using a filter and then adding bias to output and passing it to nonlinear activation function like ReLU. For multiple filters bias added will be different to generate multiple outputs of convolutional layers.

$$Dim\ (Input\ Image) = n\ x\ n\ x\ 3 \quad (1)$$

$$Dim\ (filter) = f\ x\ f\ x\ 3\ x\ u \quad (2)$$

$$Dim\ (Feature\ map) = (n - f + 1)\ x\ (n - f + 1)\ x\ u \quad (3)$$

In (1)-(3) indicates dimensions of input image, filter and feature map obtained after convolutional operation. When an image of size n×n×3 is convolved with filter of size f×f×3×u, where 3 indicates RGB image and u indicates number of filters, then size of feature map obtained after convolutional layer is (n-f+1)×(n-f+1)×u. Pooling operation is performed after convolutional operation. Function of pooling layer is to reduce dimensions of input image by preserving features in it. In max pooling operation maximum value feature is selected from given filter window, and then moved with a stride of s over the convolved image. If filter size of maxpooling operation is m x m and stride of 2 then feature map obtained at the output will have size of (m/2)×(m/2). There are multiple such layers connected one after the other [5]. Proposed CNN model consists of 8 convolutional layers with ReLU activation function and four max pooling layers. Activation size obtained at first and second convolutional layers is (62, 62, 32). Maxpooling operation is done with stride of 2 and padding is set as valid. Third and fourth convolutional layers have output of (19, 19, 32) whereas fifth and sixth convolutional layer with (8, 8, 64) and seventh and eighth convolutional layer have size of (3, 3, 64). Before applying fully connected layer output of last CONV layer is flatten in one dimensional vector. Multiple fully connected layers can be added but adding multiple fully connected layers increases amount of parameters to be dealt with [4].

Activation functions like the SoftMax for multiclass classification and the sigmoid for binary classification are included in the last layer. SoftMax activation function is used in proposed model. SoftMax activation function is used for multiclass classification where output is probability that input belongs to that class.

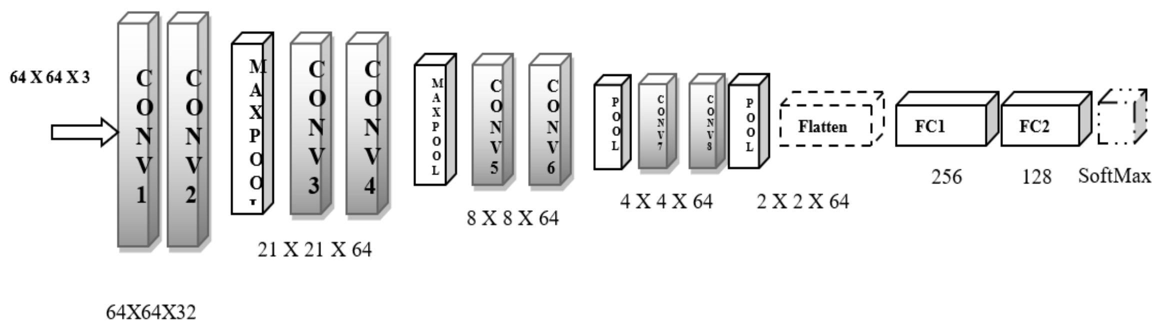

Figure 4. Structure of CNN model

Mathematical modeling of SoftMax function is given in (4), Where P(y=i) is output probability. Z_i is raw class score.

$$P(y=i) = \frac{e^{Zi}}{\sum_{j=1}^{K} e^{zj}} \tag{4}$$

Number of neurons in last fully connected layer should be equal to number of classes which gives predicted values of categories. As there are 36 distinct outputs last layer contains 36 neurons. Cost function is calculated from predicted values which signify amount of error that model is getting while making predictions. For binary classification binary cross entropy cost function can be used and for multiclass classification categorical cross entropy cost function can be used. To enhance the model's accuracy, it is necessary to minimize the cost function through training with backpropagation algorithms, such as stochastic gradient descent (SGD) or adaptive moment estimation (Adam) [2].

2.3. GoogLeNet architecture

PretrainedGoogLeNet model also called as Inception-v1 is also used to in this paper. In the transfer learning approach, a pretrained classification model is used rather of creating a new deep learning model with many convolutional and fully connected layers [20], [21]. For transfer learning, the MATLAB deep network designer toolbox is used and the GoogLeNet model is chosen since it has produced better results than Alexnet on the DHCD dataset.

GoogLeNet, the 2014 ILSVRC champion, greatly reduced its error rate than other models. Input image of size 224×224 with RGB channels is applied as an input to the model. One of the key components of this model is its 1×1 convolution, which significantly lowers the number weights and parameter biases, creating a deeper architecture. The 7×7 feature map is averaged into a 1×1 to reduce the number of trainable parameters [22]. When global average pooling is used, accuracy increases by 0.6%. To handle objects more effectively at different scales 1×1, 3×3, and 5×5 parallel 3×3 max pooling and convolution are executed. To obtain output of inception module, these modules are stacked. Inception module designed in MATLAB Deep Network designer is shown in Figure 5. Auxiliary classifier branches, including average pooling, convolution with 128 filters, and SoftMax activation function, are added to the initial models. The vanishing gradient problem is solved with layers [23].

A 22 layers deep architecture is computational efficient as it can run on devices with low computational cost. Output of inception (4a) and (4d) is connected to two auxiliary classifiers. With following properties:

- An average pooling layer, 5×5 filter and stride of 3.
- 1x1 Convolutional layer having 128 filters and activation function used is ReLU.
- A Fully Connected output layer with 1025 outputs and activation function used is ReLU.
- Dropout ratio 0.7 is set while dropout regularization.
- A SoftMax classifier with 1,000 classes output.

The model employs ReLU as the activation function for every convolution. ReLU is one of the most commonly employed activation function is CNNs. Output of ReLU is 0 when there is negative input and for any positive value of input output is same as input. Basically it is used to cancel all negative outputs [20]. Mathematically ReLU is represented by (5) Advantage of using ReLU is it can help to overcome vanishing gradient problem [24].

$$f(x) = max\,(0, x) \quad (5)$$

$$f(x) = 0\ for\ x \leq 0$$

$$f(x) = x\ for\ x > 0$$

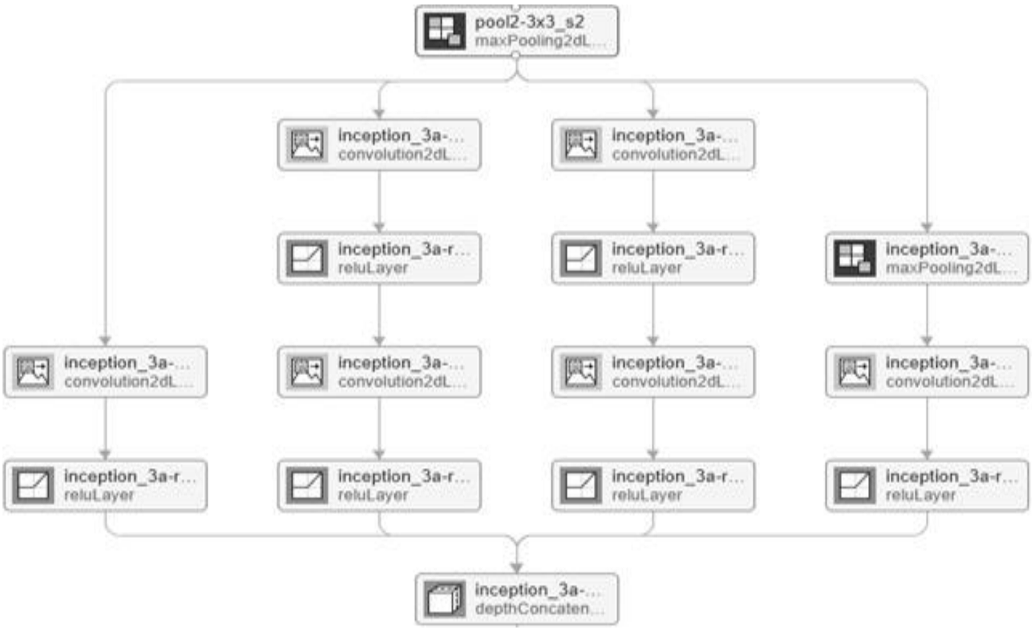

Figure 5. GoogLeNet inception module designed in MATLAB deep network designer tool

2.4. Stochastic gradient descent with momentum (SGDM)

During back propagation in order to fine tune parameters of network optimizers are implemented. Main objective of optimizers is to find global minima in order to minimize loss function. In stochastic gradient descent with momentum exponentially weighted moving average is calculated by (6), where β is hyper-parameter ranging from 0 to 1. Equation shows that velocity V_t depends on β. The experiment attempt to obtain an average of more historical data the greater the value of β, and vice versa.

$$Vt = \beta * Vt - 1 + (1 - \beta) * St \quad (6)$$
$$Where\ \beta \in [0,1]$$

While calculating global minima changes in gradient are indicated by velocity in SGDM given by (7), where Wt is weight.

$$W\ t+1\ = Wt - Vt \tag{7}$$

$$Where\ Vt\ = \beta\ *\ V\ t-1\ +\ \eta\Delta\ Wt$$

When β is set to 0, it causes a decay, however when it is equal to 1, there is no decay. Usually, only 0.5, 0.9, or 0.99 are used for β. Since SGDM involves momentum, in contrast to SGD, and can escape local minima and attain global minima, it is a better option than SGD for optimization [25].

3. RESULTS AND DISCUSSION

3.1. Results obtained for DWT-CNN model

MATLAB deep network designer is used for implementation. The image data store of 72000 images from the DHCD dataset is created, approximation coefficient (cA) obtained after discrete wavelet transform are applied to CNN model. Datastore images are split at random 80:20, with 80% of the images used for training and 20% for testing. The model is implemented on an i5 CPU. Learning rate is set to 0.001 and model is trained for 10, 15 and 20 epochs, optimizer used is SGDM. Table 1 shows results obtained for DWT – CNN model.

Table 1. The results of classification accuracy for DWT-CNN model

Epoch	% Training accuracy	% Testing accuracy	Loss
10	86.47	87.89	0.232
15	98.99	98.95	0.085
20	98.14	94.38	0.045

It is observed that best classification accuracy is obtained for 15 epochs. Testing accuracy is reduced for 20 epochs due to overfitting. Further model is trained for learning rates of 0.001, 0.01 and 0.015 by keeping number of epoch as 15. Results are shown in Table 2.

Table 2. Learning rate analysis for DWT-CNN model

Learning rate	0.001	0.01	0.015
% Training accuracy	98.99	99.15	88.34
% Testing accuracy	98.95	98.97	74.38
Loss	0.085	0.098	0.045

Results shows that maximum testing accuracy of 98.97% is obtained for learning rate of 0.01. Accuracy decreases if learning rate increases to 0.015 because a higher learning rate causes the model to learn rapidly, which affects network performance. It is also observed that training time decreases as learning rate increases because networks learn quickly.

3.2. Results obtained for GoogLeNet model

TheGoogLeNet model accepts RGB Input image of size 224 x 224, 'ColorPreprocessing', 'gray2rgb' property is set while creating image datastore. The SGDM optimizer with a learning rate of 0.001 is used to analyze the model over 10, 15, and 20 epochs. The trained model is then exported to the workspace, and 20% of data points from the image datastore are used for testing purposes. Table 3 displays the results obtained after training and testing the model for variable epochs.

Table 3. The results of classification accuracy for fine-tuned GoogLeNet model

epoch	% Training accuracy	% Testing accuracy	Loss
10	96.97	98.34	0.1589
15	99.78	99.65	0.0875
20	99.89	98.66	0.0646

Results show that as the number of epochs increases, training accuracy also increases as the model learns more from training data. It also results in an increase in average time per epoch. It is observed that training model for 15 epochs resulted in a maximum testing accuracy of 99.65% with a loss of 0.0875. Learning rate analysis of model is shown in Table 4.

Table 4. Learning rate analysis for GoogLeNet model

Learning rate	0.001	0.01	0.015
% Training accuracy	99.78 %	99.81 %	99.52 %
% Testing accuracy	99.65 %	99.68 %	98.87 %
Loss	0.0875	0.0635	0.0884

The best classification accuracy of 99.68% is obtained for learning rate of 0.01 with loss of 0.0875. Confusion matrix includes 400 images of 36 classes each are used for testing purpose. It demonstrates the confusion of similar characters with little variations. It can be observed for confusion matrix that out of 400 samples of character “भ”, 15 samples are predicted as “म” and remaining 385 samples are correctly predicted as “भ”. Similarly out of 400 sample of character “व”, 3 samples are incorrectly predicted as “ब”, 1 is classified as “प”, 1 sample as “त्र”, 1 sample as “न” and 1 sample as “त” with remaining 393 characters are correctly classified as “व”.

3.3. Validation of results

Table 5 gives comparison of proposed model with other methodologies. The findings show that the proposed DWT-CNN algorithm outperformed other techniques described in the literature with an accuracy of 98.97%. Using DWT allows the extraction of textural elements from an image, such as its structure or patterns. Additionally, the accuracy reached to 99.68% when combined with a fine-tuned GoogLeNet model. The 22-layer deep design of GoogLeNet contributes for its enhanced performance. By allowing the use of different filter sizes, the inception module improves GoogLeNet's efficiency.

Table 5. Comparison of results obtained for DHCD dataset

Author	Year	Algorithm	Percentage accuracy
Acharya *et al*. [17]	2015	Deep convolutional neural network	98.47
Aneja and Aneja [11]	2019	Transfer learning of DCNN AlexNet, DenseNe, Vgg, inceptionV3	Highest accuracy Alexnet: 98
Manocha and Tewari [15]	2021	Deep learning CNN + SVM, KNN, MLP	92
Vina *et al*. [13]	2022	Deep CNN models with 3,8,12 CONV layers	98.94
Shelke *et al*. [16]	2023	Hybrid DWT-DCT, SVM	83.09
		HOG, SVM	97.10
Proposed Algorithm	2024	Proposed DWT-CNN model	98.97
		Fine-tuned GoogLeNet model	99.68%

4. CONCLUSION

Handwritten Sanskrit manuscripts must be digitized in order to be preserved for future generations. In this paper an approach using DWT for feature extraction and convolutional neural network for classification of handwritten characters is implemented. Fine-tuned GoogLeNet model is also implemented here. DHCD dataset includes 72000 handwritten images is available freely for research is used for this study. Approximation coefficient obtained after applying DWT on input images is fed to CNN for improving feature learning capacity. After training both the models by varying hyper parameters, optimum values of learning rate and epochs is obtained as 0.01 and 15 respectively. Results shows that DWT-CNN model gives training accuracy of 99.15% and testing accuracy of 98.97% with the loss of 0.098. Fine-tuned GoogLeNet model has significantly increased accuracy to 99.81% while training and 99.68% while testing with loss of 0.0635. Additionally, the inception module allows several filter sizes to be used simultaneously. However, a challenge with GoogLeNet is that the model requires more time to train due to its deep layered design. The increasing number of epochs and learning rate resulted in overfitting of a model which can be reduced by increasing drop out regularization. Using the SGDM optimizer, which provides better generalization than other optimizers despite being slower than others, improves the overall performance of a model. The GoogLeNet model's effectiveness comes from the use of the inception module, which maintains precision while lowering processing complexity. The confusion matrix shows that there is confusion between characters “भ” and “म”, “ढ” and “द”, “घ” and “ध”. Comparison of results shows that proposed DWT-CNN model outperformed than other models. Hybrid DWT and histogram of oriented gradients (HOG) can be used for recognition of handwritten Sanskrit words. Although GoogLeNet performed better, the DWT-CNN model is a good alternative that requires lower computational cost.

REFERENCES

[1] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, "Offline recognition of Devanagari script: A survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 41, no. 6, pp. 782–796, Nov. 2011, doi: 10.1109/TSMCC.2010.2095841.

[2] D. T. Mane and U. V. Kulkarni, "Visualizing and Understanding Customized Convolutional Neural Network for Recognition of Handwritten Marathi Numerals," *Procedia Computer Science*, vol. 132, pp. 1123–1137, 2018, doi: 10.1016/j.procs.2018.05.027.

[3] A. Moudgil, S. Singh, V. Gautam, S. Rani, and S. H. Shah, "Handwritten devanagari manuscript characters recognition using capsnet," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 47–54, Jun. 2023, doi: 10.1016/j.ijcce.2023.02.001.

[4] P. P. Nair, A. James, P. Simon, and P. V. Bhagyasree, "Malayalam Handwritten Character Recognition using CNN Architecture," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 764–777, Sep. 2023, doi: 10.52549/ijeei.v11i3.4829.

[5] S. Singh, A. Sharma, and V. K. Chauhan, "Online handwritten Gurmukhi word recognition using fine-tuned Deep Convolutional Neural Network on offline features," *Machine Learning with Applications*, vol. 5, p. 100037, Sep. 2021, doi: 10.1016/j.mlwa.2021.100037.

[6] S. D. Pande *et al.*, "Digitization of handwritten Devanagari text using CNN transfer learning – A better customer service support," *Neuroscience Informatics*, vol. 2, no. 3, p. 100016, Sep. 2022, doi: 10.1016/j.neuri.2021.100016.

[7] R. Ghosh, C. Vamshi, and P. Kumar, "RNN based online handwritten word recognition in Devanagari and Bengali scripts using horizontal zoning," *Pattern Recognition*, vol. 92, pp. 203–218, Aug. 2019, doi: 10.1016/j.patcog.2019.03.030.

[8] R. Sarkhel, N. Das, A. Das, M. Kundu, and M. Nasipuri, "A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts," *Pattern Recognition*, vol. 71, pp. 78–93, Nov. 2017, doi: 10.1016/j.patcog.2017.05.022.

[9] S. P. Deore and A. Pravin, "Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset," *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 45, no. 1, p. 243, Dec. 2020, doi: 10.1007/s12046-020-01484-1.

[10] A. Bhardwaj, "An Accurate and Fine-tuned Deep-Learning Model for Handwritten Devanagari Character Recognition," vol. 20, no. 6, pp. 6717–6736, 2022.

[11] N. Aneja and S. Aneja, "Transfer Learning using CNN for Handwritten Devanagari Character Recognition," *1st IEEE International Conference on Advances in Information Technology, ICAIT 2019 - Proceedings*, pp. 293–296, 2019, doi: 10.1109/ICAIT47043.2019.8987286.

[12] V. M. Lomte and D. D. Doye, "Devanagari Text and Calligraphy Recognition Using ICF & ACF," *Computer Integrated Manufacturing Systems*, vol. 29, no. 1, pp. 88–114, 2023, doi: 10.24297/j.cims.2023.1.7.

[13] M. Vina, M. Lomte, and D. D. Doye, "Handwritten Vedic Sanskrit Text Recognition Using Deep Learning," *Journal of Algebraic Statistics*, vol. 13, no. 3, pp. 2190–2198, 2022, [Online]. Available: https://publishoa.com

[14] V. M. Lomte and D. Doye, "Gesture based Devanagari Text Recognition using Deep Learning," *Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology*, vol. 54, no. 6, pp. 239–247, 2022.

[15] S. K. Manocha and P. Tewari, "Devanagari Handwritten Character Recognition using CNN as Feature Extractor," in *2021 International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2021*, Oct. 2021, pp. 1–5. doi: 10.1109/SMARTGENCON51891.2021.9645786.

[16] S. V. Shelke, D. M. Chandwadkar, S. P. Ugale, and R. V. Chothe, "Combining Multiple Feature Extraction and Classification Methods to Study Performance of Handwritten Sanskrit Character Recognition," in *2023 7th International Conference on Computing, Communication, Control and Automation, ICCUBEA 2023*, Aug. 2023, pp. 1–6. doi: 10.1109/ICCUBEA58933.2023.10391986.

[17] S. Acharya, A. K. Pant, and P. K. Gyawali, "Deep learning based large scale handwritten Devanagari character recognition," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Dec. 2015, pp. 1–6. doi: 10.1109/SKIMA.2015.7400041.

[18] M. Wulandari, R. Chai, B. Basari, and D. Gunawan, "Hybrid Feature Extractor Using Discrete Wavelet Transform and Histogram of Oriented Gradient on Convolutional-Neural-Network-Based Palm Vein Recognition," *Sensors*, vol. 24, no. 2, p. 341, Jan. 2024, doi: 10.3390/s24020341.

[19] S. Shelke, "Handwritten Character Recognition using Wavelet Transform for Feature Extraction," *International Journal of Multidisciplinary Educational Research (IJMER)*, vol. 3, no. March 2014, pp. 3–7, 2016.

[20] N. Azawi, "Handwritten digits recognition using transfer learning," *Computers and Electrical Engineering*, vol. 106, p. 108604, Mar. 2023, doi: 10.1016/j.compeleceng.2023.108604.

[21] P. P. Nair, A. James, P. Simon, and P. V. Bhagyasree, "Malayalam Handwritten Character Recognition using CNN Architecture," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 764–777, Sep. 2023, doi: 10.52549/ijeei.v11i3.4829.

[22] C. Szegedy *et al.*, "Going Deeper with Convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9. doi: 10.48550/arXiv.1409.4842.

[23] Z. Zhong, L. Jin, and Z. Xie, "High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature maps," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, Aug. 2015, vol. 2015-November, pp. 846–850. doi: 10.1109/ICDAR.2015.7333881.

[24] M. Mishra, T. Choudhury, and T. Sarkar, "Devanagari Handwritten Character Recognition," *2021 IEEE India Council International Subsections Conference, INDISCON 2021*, 2021, doi: 10.1109/INDISCON53343.2021.9582192.

[25] L. Liu and X. Luo, "A New Accelerated Stochastic Gradient Method with Momentum," in *Proceedings of Machine Learning Research*, 2020, pp. 1–10. [Online]. Available: http://arxiv.org/abs/2006.00423.

BIOGRAPHIES OF AUTHORS

Shraddha V. Shelke is working as an Assistant Professor in Electronics and Telecommunication Engineering Department of K. K. Wagh Institute of Engineering Education and research, Nashik, Maharashtra since last 13 years. She is perusing her Ph. D. in Electronics and Telecommunication from Savitribai Phule Pune University. She can be contacted at email: svshelke@kkwagh.edu.in.

Dinesh M. Chandwadkar is a professor and head of E & TC Department at K. K. Wagh Institute of Engineering Education & Research, Nashik, India. His area of interest includes Signal Processing, Power Electronics, Mechatronics, and Automotive Electronics. He holds Ph.D. degree and he has published More than 50 research papers in reputed Journals. He is working as Board of Studies member of Electronics and Telecommunication Engineering for Pune University. He can be contacted at email: dmchandwadkar@kkwagh.edu.in.

Sunita P. Ugale is working as a Professor in Electronics and Telecommunication Engineering Department of K. K. Wagh Institute of Engineering Education and Research, Nashik, Maharashtra since last 28 years. She holds Ph.D. degree and her special fields of interest include Fiber Optics Communication, Optical Sensors, Automation and VLSI technology. She can be contacted at email: spugale@kkwagh.edu.in.

Rupali V. Chothe is working as an Assistant Professor in Electronics and Telecommunication Engineering Department of K. K. Wagh Institute of Engineering Education and Research, Nashik, Maharashtra since last 16 years. She is pursuing Ph.D. in Electronics and Telecommunication from Savitribai Phule Pune University. She can be contacted at email: rvchothe@kkwagh.edu.in.