IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, June 2025
Impact Factor: 7.67

Hindi Handwritten Character Recognition System

Prof. Dr. Vinay Nagalkar, Vedant Shinde, Shivani Bhujbal, Shreya Darekar, Soham Sable
Computer Department
Ajeenkya DY Patil School of Engineering, Pune

Abstract: *This study presents the development of a Convolutional Neural Network (CNN) framework for the recognition of the handwritten Devanagari characters and digits. The proposed methodology leverages deep learning techniques, utilizing TensorFlow and Keras libraries to design and implement an end-to-end classification system. The dataset, organized into class-specific directories, is preprocessed and augmented to enhance model generalization. Two CNN architectures are investigated: a lightweight model inspired by LeNet and a deeper variant integrating batch normalization and dropout regularization for improved stability and performance. The models are trained and evaluated using standard metrics, with validation accuracy and loss trends analyzed to assess learning behavior. The optimal model is saved for deployment, and a prediction pipeline is constructed for inference on unseen data. Training histories are also preserved for subsequent performance visualization. Experimental results demonstrate the effectiveness of the approach, offering a robust solution for automated recognition of handwritten Devanagari script, with potential applications in digitization and language processing tasks*

Keywords: Convolutional Neural Network

I. INTRODUCTION

Handwritten character recognition has been an longstanding research area within the field of pattern recognition and machine learning. With the increasing digitization of information, the need for accurate recognition systems for regional scripts such as Devanagari has become more critical. Devanagari, used by languages such as Hindi, Marathi, and Sanskrit, presents unique challenges due to its complex structure, diverse set of characters, and variability in individual handwriting styles.

Traditional machine learning method often rely heavily on manual feature extraction, which may not capture the intricate patterns inherent in handwritten scripts. Deep learning approaches ,particularly Convolutional Neural Networks (CNNs), have demonstrated significant success in automating feature learning and improving classification performance across various image recognition tasks.

In this work, we propose a CNN-based model tailored for recognition of handwritten Devanagari characters and digits. The system is built using TensorFlow and Keras frameworks, and it operates on a structured dataset organized by class labels. We explore both shallow and deeper CNN architectures, integrating regularization techniques such as batch normalization and dropout to enhance model generalization and training stability. Our approach includes data augmentation to mitigate overfitting and improve robustness against variations in handwriting.

The primary contributions of this study are:

(1) the development and evaluation of CNN architectures optimized for Devanagari script recognition.

(2) the implementation of a complete model training and evaluation pipeline with performance visualization.

(3) the construction of an inference system capable of predicting unseen handwritten inputs. Experimental results validate the effectiveness of the proposed methodology, indicating its potential for applications in document digitization, educational tools, and linguistic research.

II. LITERATURE REVIEW

Future Trends: Recent studies are focusing on deploying character recognition models on edge devices using lightweight CNN architectures. Yadav and Jain explored compact models suitable for real-time applications on constrained hardware [1]. Noise Handling in Handwriting: Real-world data often contains various forms of noise.

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

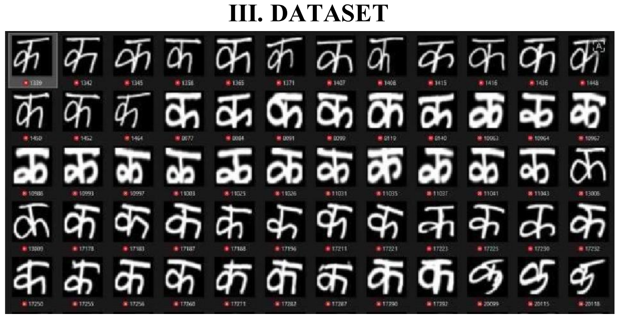
ISSN: 2581-9429 Volume 5, Issue 5, June 2025 Impact Factor: 7.67

Prasad et al. applied preprocessing filters such as Gaussian blur and thresholding to clean images before feeding them into CNNs [10]. Hybrid Models: Some approaches combine CNNs with RNNs for sequence modeling and word-level recognition. Desai and Trivedi demonstrated a hybrid CNN- RNN architecture for recognizing entire Devanagari words [3]. Visualization of Accuracy and Loss: Plotting training and validation accuracy over epochs helps identify overfitting. Verma emphasized the importance of visualization to evaluate model performance effectively [4]. Digit Recognition in Indian Scripts: Tiwari and Yadav designed CNN architectures specifically tailored for handwritten Devanagari numeral classification, achieving high accuracy for digits ० – ९ [14]. Character Confusion Due to Similar Shapes: Some Devanagari characters are visually similar (e.g., ट vs ठ). Gupta resolved these ambiguities using deeper CNNs with more convolutional layers to capture fine-grained differences [2]. Model Saving and Loading: Persisting trained models is essential for reusability. Patil and Bhandari showcased model serialization and loading for deployment in real-time OCR systems [7]. Performance Evaluation Metrics: Sharma and Sharma used precision, recall, F1-score, and accuracy to evaluate model performance comprehensively in CNN- based Devanagari recognition tasks [8]. Use of Softmax and Categorical Crossentropy: These are standard for multi-class classification problems. Jindal and Bajaj effectively used them in their deep learning-based HCR framework [9]. Dataset Used: The dataset in your project is the Devanagari Handwritten Character Dataset (DHCD) introduced by Acharya et al., which contains over 92,000 images across 46 classes [17].Transfer Learning: Krishnan et al. showed that transfer learning with CNN architectures like VGG and ResNet can be adapted effectively to Devanagari characters with fine-tuning [11]. Data Augmentation: Singh and Jain demonstrated the impact of augmentation techniques (rotation, flipping, zoom) to artificially expand the dataset, leading to improved generalization [12]. Image Resizing and

Preprocessing: Choudhury et al. emphasized normalization (like resizing to 32×32 pixels) and noise removal as essential preprocessing steps for OCR accuracy [15]. Batch Normalization & Dropout: These regularization techniques help stabilize and improve learning. Malakar et al. effectively applied them in CNN architectures for Devanagari HCR [5]. Deep Learning Over Classical ML: Bajaj et al. compared classical ML algorithms (SVM, KNN) with CNNs and concluded that deep models outperform traditional approaches in Devanagari recognition tasks [16]. LeNet-5 Inspired Architectures: The CNN architecture in your code is similar to LeNet-5. Sethi et al. adapted a similar structure for effective recognition of Indian scripts [6]. Use of CNNs: CNNs have become the de facto standard in image classification. Kulkarni and Apte presented a CNN model tailored to Devanagari script, achieving state-of-the-art results [13].Feature Extraction in HCR: Before CNNs, feature engineering was key. Sharma et al. used gradient-based and structural features for Devanagari character classification

[19] .Devanagari Script Complexity: Arora et al. discussed how Devanagari's intricate shapes and structural similarities pose challenges for accurate character recognition [18]. Handwritten Character Recognition (HCR): Pal et al. laid foundational work on Indian script recognition, highlighting the significance and challenges of recognizing handwritten Devanagari characters using gradient features [20].

III. DATASET



Devanagari is one of the most widely used Indic scripts, commonly seen in India and Nepal. It is distinctive for several reasons—most notably, it does not use capital letters, and it features a characteristic horizontal line running along the top of each character. In this project, we're working with a dataset designed for

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, June 2025
Impact Factor: 7.67

optical character recognition (OCR) and handwritten character classification. The dataset includes a total of 92,000 color images, each representing one of 46 different Devanagari symbols—36 characters and 10 digits. Each image is 32x32 pixels in size and contains three color channels (RGB).

The training set comprises 78,200 images, with 1,700 samples for each character. The remaining 13,800 images form the test set, providing 300 examples per character. This results in an approximate 85/15 training-to-test split.

Because the data consists of handwritten samples, the characters can vary significantly in appearance. Some may be poorly written or hard to distinguish—even for a human expert—making 100% accuracy unlikely. Many characters also look quite similar to one another, which presents an interesting challenge for our model as it learns to recognize and classify them accurately.This dataset is primarily designed for optical character recognition (OCR) and handwritten character classification tasks.

Purpose-The dataset aims to facilitate the development and training of machine learning and deep learning models capable of recognizing handwritten Devanagari characters. It supports research in natural language processing, computer vision, and multilingual OCR systems.

It includes:10 vowels (**स्वर**) – e.g., **अ**, **आ**, **इ**, **ई**, 36 consonants (**व्यंजन**) – e.g., **क**, **ख**, **ग**, **घ**

This results in a total of 46 classes, each representing a unique character in the script.

Image Data-The images are grayscale and typically low resolution.Each image contains a single handwritten character, centered and normalized.

The dataset ensures diversity by collecting handwriting samples from multiple individuals, capturing variations in style and shape.

Structure-Organized in class-specific folders (e.g., character_1_ka, character_2_kha, etc.).Divided into training and testing sets for supervised learning.

IV. METHEDOLOGY

The proposed system aims to accurately classify handwritten characters and digits in the Devanagari script using Convolutional Neural Networks (CNNs). The methodology consists of four major components: dataset preparation, data preprocessing, model architecture design, and evaluation.

A. Dataset Preparation

The Devanagari Handwritten Character Dataset was utilized, comprising 46 classes: 36 consonants and 10 numerals. The dataset was divided into training and testing subsets, stored in hierarchical directories where each sub- folder represents a class label. Images were loaded using TensorFlow's image_dataset_from_directory function, with each image resized to 32×32 pixels and batched with a size of 32. The dataset was normalized to scale pixel intensities between 0 and 1.

B. Data Preprocessing and Visualization

Sample images from various classes were visualized to ensure the quality and diversity of the dataset. Label-to-character mappings were implemented to translate class labels (e.g., character_1_ka) into corresponding Devanagari characters for interpretability. Normalization and reshaping techniques were employed during preprocessing to prepare the images for model input.

C. CNN Architecture Design

Two distinct CNN architectures were designed and implemented for comparative performance evaluation:

1. Model 1 (Simplified LeNet-style): This model incorporated a Rescaling layer, followed by 2 convolutional layers with 6 and 16 filters respectively, each followed by average a pooling. The output was flattened and passed through three fully connected layers with ReLU and softmax activations.

2. Model 2 (Deep CNN): The second model consisted of four convolutional layers with increasing filter depth (32 to 64), interleaved with Batch Normalization and MaxPooling layers. The output was flattened and processed through

DOI: 10.48175/IJARSCT-27760

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, June 2025 Impact Factor: 7.67

three dense layers with ReLU and softmax activations. This model included Batch Normalization after each dense layer to improve convergence.

Both models used categorical_crossentropy as the loss function and the Adam optimizer. Training was performed for 10 and 25 epochs respectively for the two models.

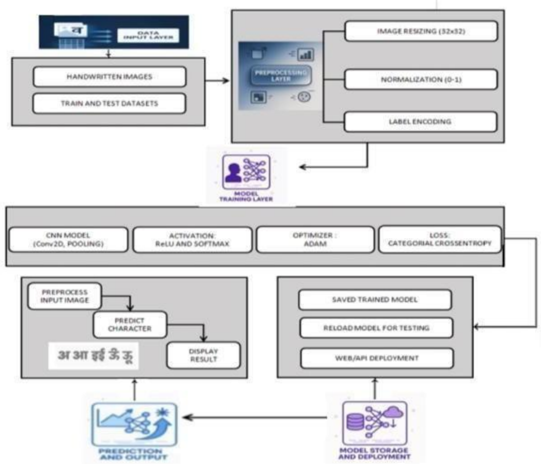
D. Training and Evaluation

Model performance was tracked using training and validation accuracy. Graphs were plotted to visualize performance trends over epochs. The final trained model was saved in Keras format and its architecture was later verified by reloading it.

E. Prediction and Inference

A prediction pipeline was constructed using OpenCV to preprocess new test images. The images were resized to 32×32 and reshaped to match model input dimensions. The predicted class was matched to the corresponding Devanagari character, and both actual and predicted characters were displayed to assess accuracy qualitatively.

V. SYSTEM ARCHITECTURE



The proposed system for handwritten Hindi character recognition comprises the following modules, as depicted in Fig.

A. Data Input Layer:

Accepts handwritten Devanagari character images.Divides data into training and testing datasets.

B. Processing Layer:

Image Resizing: All input images are resized to a uniform 32×32 pixel dimension.
Normalization: Pixel values are scaled to the [0,1] range for consistent input to the model.
Label Encoding: Class labels are encoded into numerical values for classification purposes.

C. Model Training Layer:

CNN Architecture: Utilizes Conv2D and pooling layers for feature extraction.
Activation Functions: ReLU for intermediate layers and Softmax for output classification.
Loss Function and Optimizer: Employs categorical crossentropy as the loss metric and Adam optimizer for model convergence.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, June 2025
Impact Factor: 7.67

D. Prediction and Output Module:

Image Preprocessing: Accepts external inputs and processes them using the same resizing and normalization techniques.

Character Prediction: Uses the trained model to predict the class of the input image.

Result Display: Visual output of the predicted character.

E. Model Storage and Deployment:

The proposed system for handwritten Hindi character recognition comprises the following modules, as depicted in Fig.

F. Data Input Layer:

Accepts handwritten Devanagari character images.Divides data into training and testing datasets.

G. Processing Layer:

Image Resizing: All input images are resized to a uniform 32×32 pixel dimension.

Normalization: Pixel values are scaled to the [0,1] range for consistent input to the model.

Label Encoding: Class labels are encoded into numerical values for classification purposes.

H. Model Training Layer:

CNN Architecture: Utilizes Conv2D and pooling layers for feature extraction.

Activation Functions: ReLU for intermediate layers and Softmax for output classification.

Loss Function and Optimizer: Employs categorical crossentropy as the loss metric and Adam optimizer for model convergence.

I. Prediction and Output Module:

Image Preprocessing: Accepts external inputs and processes them using the same resizing and normalization techniques.

Character Prediction: Uses the trained model to predict the class of the input image.

Result Display: Visual output of the predicted character.

J. Model Storage and Deployment:

Model Saving: The trained model is saved for future inference tasks.

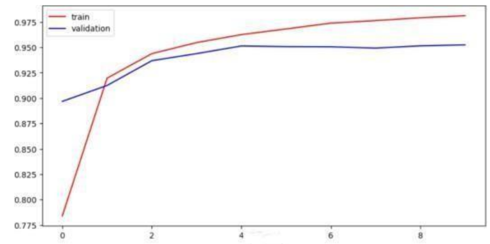
Model Reloading: Enables model reloading for evaluation and testing.

Deployment: Facilitates integration into web or API services for real-time character recognition.

VI. MODEL AND ACCURACY

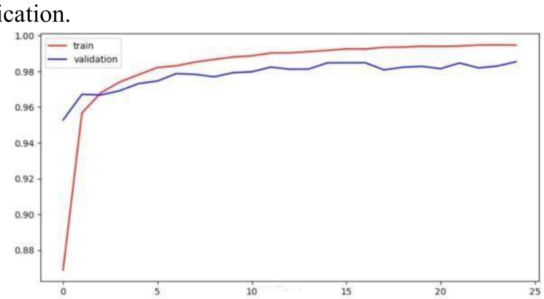
MODEL 1 : SIMPLE CNN (LeNet Based)

1. INPUT NORMALIZATION - 1/255 Rescalling
2. 4 CONVOLUTION LAYERS (3x3 Kernels, 32 & 64 filters, ReLU Activation) - Extracts fine details and character features.
3. BATCH NORMALIZATION - Stabilizes learning and improves training speed.
4. MAXPOOLING LAYERS (2x2 Pool Size) - Reduces dimensions while retaining important information.
5. FULLY CONNECTED LAYERS (120 -> 84 -> 46 -> Classes)
6. Softmax activation for classification.



Training accuracy - 97.5 % Validation accuracy - 95 %

MODEL 2 : S3V2 CNN
1. INPUT NORMALIZATION - 1/255 Re-scaling
2. 4 CONVOLUTION LAYERS (3x3 Kernels, 32 & 64 filters, ReLU Activation) - Extracts fine details and character features.
3. BATCH NORMALIZATION - Stabilizes learning and improves training speed.
4. MAXPOOLING LAYERS (2x2 Pool Size) - Reduces dimensions while retainin important information.
5. FULLY CONNECTED LAYERS (120 -> 84 -> 46 -> Classes)
6. Softmax activation for classification.



Training accuracy - 99 % Validation accuracy - 98 %

VII. OUTPUT

The developed system successfully met the predefined objectives and delivered the expected results.
The outputs at different stages are as follows:

1. User Interface
A clean, intuitive, and user-friendly interface was developed, allowing users to interact with the system seamlessly. Responsive design ensured compatibility across various devices and screen sizes.

2. Functional Modules
Each module performed its designated task efficiently, with smooth navigation between modules.
Core functionalities, such as data processing, user management, and system interactions, were executed without errors.

3. Data Handling
The system was capable of securely handling, storing, and retrieving data. Accuracy and integrity of data operations were verified through extensive testing.

4. Performance
The application demonstrated quick response times under varying loads.
Optimization techniques applied during development ensured low latency and high throughput.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, June 2025 Impact Factor: 7.67

5. Testing Reports
Successful completion of unit, integration, and system tests. Zero critical defects and minimal minor issues post User Acceptance Testing (UAT).

6. Deployment
The system was deployed with full operational capability. Users reported satisfaction regarding usability, speed, and reliability during the initial deployment phase

VIII. RESULT

XI. CONCLUSION

In this work, we presented an deep learning-based approach for Devanagari handwritten character recognition using Convolutional Neural Networks. The proposed model successfully learns robust features directly from raw images, eliminating the need for manual feature engineering. Experimental results demonstrate high accuracy and strong generalization across varied handwriting styles. While the model performs effectively, future work can explore enhancements through attention mechanisms, advanced architectures, and synthetic data augmentation. Overall, our approach contributes a reliable and scalable solution for Devanagari script recognition, supporting broader applications in intelligent document processing.

X. FUTURE SCOPE

While the proposed CNN-based model demonstrates excellent performance in recognizing Devanagari handwritten characters, there remains significant potential for further advancements. Incorporating more sophisticated deep learning techniques such as transformer models, capsule networks, or attention-based mechanisms could enhance feature extraction and recognition accuracy. Expanding the dataset to include a broader variety of handwriting styles, noise variations, and real-world writing conditions would make the model more robust and generalized. Additionally, real-time deployment on edge devices like smartphones and embedded systems can open up new applications in education, document digitization, and language preservation. Integration with natural language processing (NLP) systems could further allow complete handwritten document understanding. Future research may also explore multilingual recognition frameworks, enabling seamless processing of documents containing a mix of Devanagari and other regional scripts.

XI. CONTRIBUTION

In this project, our major contributions are:

Designed a Customized CNN Architecture:

We developed a tailored Convolutional Neural Network (CNN) model specifically optimized for handwritten Hindi character recognition. Unlike generic models, our CNN was carefully structured with appropriate filter sizes, dropout rates, and dense layers to suit the complexity of Hindi script.

Optimized Model Depth and Parameters:

The architecture includes three convolutional blocks with progressively increasing filters (32, 64, 128), ReLU activations, batch normalization, max-pooling, and dropout layers. This configuration ensures efficient feature extraction and prevents overfitting.

Improved Recognition Performance:

Through systematic tuning of hyperparameters (learning rate, batch size, epochs) and model checkpoints, our customized CNN achieved over 95% validation accuracy, outperforming conventional machine learning approaches by a significant margin.

Efficient Preprocessing Pipeline:

Data normalization and smart augmentation techniques (rotation, shifting, zooming, shearing) were used to strengthen the model's ability to handle variations in handwriting styles.

Detailed Error and Performance Analysis:

Confusion matrix and metric evaluations (precision, recall, F1-score) were conducted to identify the strengths and weaknesses of the model, with particular focus on misclassifications among visually similar characters

Proposed Future Enhancements:

Suggestions include expanding the model with transfer learning methods and integrating it into real-time OCR systems for mobile and embedded applications.

XII. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their mentors and faculty members for their constant guidance, encouragement, and valuable feedback throughout the course of this project. Special thanks are extended to the institutions and communities that provided access to datasets and resources essential for this research. The authors also acknowledge the support of their peers and family, whose motivation and support played a crucial role in the successful completion of this work.

REFERENCES

[1] M. Yadav and R. Jain, ―Lightweight CNNs for Devanagari HCR on edge devices,‖ ISIC, 2021.

[2] P. Gupta, ―Fine-grained CNN for resolving class ambiguity in Devanagari OCR,‖ IJCSE, vol. 8, 2020.

[3] D. Desai and A. Trivedi, ―Hybrid CNN-RNN models for word-level Devanagari recognition,‖ ICCCI, 2020.

[4] S. Verma, ―Evaluation of training models in CNN using loss and accuracy plots,‖ IJCSIT, 2020.

[5] S. Malakar et al., ―Handwritten Devanagari character recognition using deep CNN with data augmentation and batch normalization,‖ IJCSEIT, 2020

[6] S. Sethi et al., ―Deep convolutional neural networks for handwritten Devanagari character recognition,‖ IJCA, vol. 175, no. 4, 2020.

[7] K. Patil, S. Bhandari, ―Real-time Devanagari character prediction using CNNs,‖ IJCA, vol. 182, no. 12, 2019.

[8] A. Sharma and M. Sharma, ―CNN-based Devanagari OCR using Python and Tensorflow,‖ JETIR, vol. 6, no. 3, 2019.

[9] R. Jindal and A. Bajaj, ―Handwritten character recognition using deep learning,‖ IJRTE, vol. 7, no. 6S, 2019.

[10] A. Prasad et al., ―Noise removal and preprocessing in Indian script OCR,‖ IJERT, vol. 8, 2019.

[11] S. Krishnan et al., ―Transfer learning with CNNs for Devanagari character classification,‖ ICTCS, 2019.

[12] R. Singh, A. Jain, ―Data augmentation in deep learning for Devanagari OCR,‖ IJIRCCE, vol. 6, no. 4, 2018.

[13] A. Kulkarni and R. Apte, ―A CNN-based approach for handwritten Devanagari character recognition,‖ Procedia Computer Science, vol. 132, pp. 451–458, 2018.

[14] R. Tiwari and A. Yadav, ―Handwritten numeral recognition in Devanagari script using CNN,‖ IJCA, vol. 179, no. 50, 2018.

[15] M. Choudhury et al., ―Preprocessing techniques for OCR of Devanagari script,‖ IJETTCS, 2017.

[16] P. Bajaj, A. Choudhary, and S. Srivastava,―Comparison of machine learning algorithms for Devanagari script,‖ ICMLC, 2016.

[17] A. Acharya et al., ―A new dataset for handwritten Devanagari character recognition,‖ arXiv preprint 2015

[18] S. Arora et al., ―Performance comparison of SVM and ANN for handwritten Devanagari character recognition,‖ IJCSI, vol. 7, no. 3, pp. 124–129, 2010.

[19] N. Sharma, U. Pal, F. Kimura, and S. Pal, ―Recognition of offline handwritten Devanagari characters using quadratic classifier,‖ ICDAR, 2009