International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

DEVANAGARI DIGITIZATION: A MACHINE LEARNING PARADIGM FOR ACCURATE WORD CONVERSION

Shilpa Tyagi[1], Chiranjit Dutta[2], Manu Singh[3]

[1]Department of Computer Applications, SRM Institute of Science and Technology,
Delhi-NCR Campus, Uttar Pradesh, 201204, India
2Department of Computer Science and Engineering, SRM Institute of Science and Technology,
Delhi-NCR Campus, Uttar Pradesh,201204, India
[3]Department of Computer Science, ABES Engineering College,
Ghaziabad, Uttar Pradesh, India

Abstract

The digitization and conversion of handwritten or printed documents in the Devanagari script into editable Word format have become increasingly essential in the digital age. This article presents an innovative approach that utilizes machine learning techniques to perform the conversion seamlessly and accurately. The proposed system is designed to handle a wide range of Devanagari scripts from various sources and formats. It provides a combination of Optical Character Recognition (OCR) and machine learning algorithms to recognize the characters, words, and structure of the script. The model is trained on a diverse dataset comprising handwritten and printed Devanagari text, covering different writing styles and qualities. The character recognition module employs convolutional neural networks (CNN) and recurrent neural networks (RNN) to accurately identify Devanagari characters. Word segmentation techniques enable the system to split sentences into meaningful units, considering the specific rules and characteristics of the Devanagari script. In addition to transcription, the system recognizes contextual information and maintains the logical structure of the text. The proposed model demonstrates the effectiveness and accuracy of the proposed system in converting Devanagari script into an editable Word format, surpassing traditional OCR methods. It has immense potential in preserving and promoting the cultural and linguistic heritage encapsulated in Devanagari documents and opens new possibilities for research, translation, and content accessibility in a digital context.

Keywords—Devanagari Script, Optical Character Recognition, Convolutional Neural Network, Recurrent Neural Network, Machine Learning.

Introduction

The Devanagari script, also known as Nagari, is a writing system and script used primarily for several languages in the Indian subcontinent. It is one of the most widely used scripts in South Asia and is particularly known for its association with languages such as Sanskrit, Hindi,

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Marathi, Nepali, Konkani, and several others [1] [2]. There are several key aspects and characteristics about the Devanagari script:

Origin and History: The Devanagari script has a rich history dating back to the 7th century CE. It evolved from the Brahmi script and was primarily developed for writing Sanskrit, one of the classical languages of India. Over time, it has been adapted to write various modern Indian languages.

Characteristics: Devanagari is an abugida script, which means that each character represents a consonant with an inherent vowel sound that can be modified with diacritic marks to indicate other vowel sounds. It is written from left to right.

Alphabet: The Devanagari script consists of a set of consonants and vowels. It has 36 basic consonants and a range of diacritics that modify these characters to represent other consonants or vowel sounds. Additionally, there are characters for numerals and various punctuation marks.

Complex Characters: Devanagari features complex characters known as "ligatures" and "conjunct characters" that result from the combination of two or more basic characters. This complexity is one of the unique features of the script.

Languages: While it is most commonly associated with Sanskrit and Hindi, Devanagari is used for a wide range of languages across India and Nepal. Each language may have some variations in its script based on its phonetic requirements.

Unicode: The Devanagari script is part of the Unicode standard, which ensures its compatibility with modern computer systems and digital platforms.

Usage: Devanagari script is used in various contexts, including literature, religious texts, official documents, education, and digital communication. It has a significant cultural and historical importance in South Asia.

Calligraphy: Devanagari calligraphy is a highly regarded art form in India, and different styles of writing have developed over the centuries.

Variations: There are regional variations of the Devanagari script, each with its own unique features. For example, the script used for Marathi differs in some characters and ligatures from the one used for Hindi.

Figure. 1. Devnagari Script

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Figure1 shows the example of Devnagari Script and its characters. The Devanagari script is not only a means of communication but also an important part of the cultural and linguistic identity of the Indian subcontinent. Its adaptability and the ability to write a diverse range of languages make it a crucial script in South Asia. The conversion of Devanagari script into an editable Word format involves distinct methods such as Optical Character Recognition (OCR) techniques, machine learning, and specific software tools.

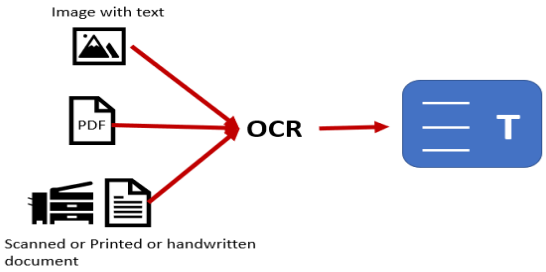

Figure. 2. OCR Working

Optical Character Recognition (OCR) Software: Figure 2 indicates the working of OCR that tries to convert distinct files or images into editable text. OCR software is designed to recognize and convert printed or handwritten text into digital text that can be edited in word processing software like Microsoft Word [3]. There are OCR software options available that support Devanagari script. Popular OCR software such as ABBYY FineReader, Adobe Acrobat, and Google OCR can be used for this purpose. Distinct online OCR services are also available for free or as paid options. Websites like i2OCR, Online OCR, and Google Drive's built-in OCR feature can be used to convert Devanagari text into an editable format. The document can be upload as an image or scanned document containing Devanagari text, and the online OCR service will process the image and provide an editable text output.

Machine Learning-Based Models: Developing or utilizing machine learning models specifically trained for Devanagari script recognition can be an effective method. Train a custom machine learning model using a dataset of Devanagari script samples [4]. The use of deep learning techniques, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), for character recognition. These models can be integrated into applications or scripts to convert Devanagari text into an editable Word format.

Custom Software Development: Building a custom software application for Devanagari script conversion is an option if we have specific requirements. Such software can be designed to recognize Devanagari characters, handle ligatures and conjunct characters, and format the text correctly before exporting it to Word format.

Transcription Services: In cases where handwritten or complex Devanagari script needs to be transcribed, it can utilize professional transcription services. These services often employ trained individuals who manually convert the text into an editable format.

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Document Scanners with OCR: Some document scanners come with built-in OCR capabilities. If the end-user has access to such a scanner, then it can scan the Devanagari document directly and have it converted into an editable format.

Translation and Word Processing Software: Translation software and word processors like Google Docs or Microsoft Word often provide built-in OCR features. These can be used for converting Devanagari script into an editable format.

Additionally, post-processing may be required to ensure the converted text is correctly formatted and free from errors. Traditional conversion methods of Devanagari script into editable Word format often face several challenges and limitations due to the complexity and uniqueness of the script. Some of the common problems associated with these methods include:

Character Recognition Errors: Devanagari script has a wide range of characters, including consonants, vowels, ligatures, and conjunct characters. Traditional methods may struggle to accurately recognize and differentiate these characters, leading to recognition errors.

Ligatures and Conjunct Characters: Devanagari script frequently uses ligatures and conjunct characters, where multiple characters are combined into a single glyph. Traditional methods may not handle these complex combinations well, resulting in broken or incorrectly recognized text.

Font and Writing Style Variations: Devanagari script is written in various fonts and styles. Traditional methods may not be robust enough to handle these variations, leading to recognition errors and formatting issues.

Handwritten Text Challenges: Handwritten Devanagari text can be highly variable in terms of quality and style. Traditional methods may struggle to accurately recognize characters in such documents, especially when they include cursive or irregular handwriting.

Language Variations: Devanagari script is used for multiple languages in India and Nepal. Different languages may have variations in character usage and pronunciation, making recognition more challenging for traditional methods.

Non-standard Layouts: Devanagari documents may not always follow standard formatting rules, leading to problems with text layout, paragraph breaks, and line breaks during conversion.

Contextual Analysis: Understanding the context and meaning of the text is crucial in Devanagari, as it affects the choice of characters and ligatures. Traditional methods may not always capture the correct context, leading to incorrect translations.

Complex Script Rules: Devanagari script has complex rules for character combination, diacritics, and syllable formation. Traditional methods may not fully understand or apply these rules, resulting in incorrect text conversion.

Special Characters and Punctuation: Devanagari script includes special characters and punctuation marks that need to be accurately recognized and formatted. Traditional methods may struggle with these elements.

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Editing and Post-processing: After conversion, the text often requires manual editing and post-processing to correct errors and ensure proper formatting. This can be time-consuming and complex.

Limited Language Support: Some traditional OCR and conversion tools may have limited support for Devanagari and its variations, making them unsuitable for certain languages or dialects.

Loss of Formatting: Traditional methods may not always preserve the original formatting, which is especially important in documents with complex structures, such as poetry or technical manuals.

To address these problems, advanced methods that incorporate machine learning and deep learning techniques are increasingly being used. These methods have the potential to improve accuracy, handle variations in fonts and writing styles, and provide better support for ligatures and conjunct characters, making them more suitable for converting Devanagari script into an editable Word format. The major contribution of the current research article are as follows:

- To Propose an integrated approach for the conversion of Devnagari script into editable word format.
- To implement a custom machine learning model for character recognition, word segmentation, and language understanding.
- To improve the efficient and accuracy of the system by considering context and semantics of the text.

The rest of the article is organized as in the following section such as, Section II represents the literature review related to the machine learning model used for the conversion of Devnagari Script into editable word format. Section III explores the dataset with its description and exploratory data analysis. Section IV defines the proposed mechanism utilized to build the system more effective. Section V discuss about the result section and comparative analysis of the proposed system with existing system. Section VI concludes the article with some future scope key points.

Literature Review

A literature review on the conversion of Devanagari script into editable Word format using Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) reveals the state of the art and recent advancements in this field. This review focuses on key research papers and studies that have explored this topic.

Agnihotri [5] presented a CNN-based approach for recognizing offline handwritten Devanagari script. The study demonstrates the effectiveness of CNNs in recognizing individual characters, which is a crucial step in converting Devanagari script into an editable format. For better outcomes, more than 1000 samples are collected for training and testing purpose. The Supervised Layer Wise training of a Deep Convolutional Neural Network (SL-DCNN) architecture was created by Roy et al. [6]. On the challenging CMATERdb 3.1.3.3 handwritten

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Bangla isolated compound character dataset, it had established a new benchmark with an error rate of 9.67%. The model achieved an outstanding result in Bangla compound character recognition, cutting the error rate by around 10% compared to the prior benchmarks on the same.

Alom et al. [7] evaluated the effectiveness of CNN and DBN with the incorporation of Deep Learning using a mix of dropout and multiple filters on a dataset CMATERdb 3.1.1. After conducting numerous tests, they find that, when compared to other methods, CNN with Gabor feature and dropout achieves superior accuracy for BANGLA digit recognition. Convolution neural network (CNN) handwritten character recognition for ARABIC characters was employed by El-sawy et al. [8]. The results were encouraging, showing a 94.9% classification accuracy rate on testing picture scripts. In offline mode, Savitha Attigeri [9] demonstrated a neural network-based method for offline character recognition from handwritten scripts without the need for feature extraction from scanned images. They obtained accuracy for 90.19/5 from 100 neurons by using layers.

According to Adnan et al.'s analysis [10] of several deep convolution neural networks (DCNNs), DenseNet can produce better outcomes for BANGLA character recognition. Furthermore, mentioned is the possibility of achieving up to 98% accuracy in the recognition of numbers, letters, and special characters. Deep convolution neural network (DCNN) and adaptive gradient techniques were employed by Jangid et al. [11] for DEVENAGRI character recognition. Additionally, they have employed the RAMSProp optimizer approach and Network Architecture – 6. Using the ISIDCHAR and V2DMCHAR datasets, they were able to attain a recognition accuracy of almost 98%. The model and staying informed about the latest security threats and solutions is crucial for maintaining a secure cloud environment.

Bati et al. [12] Utilized distinct models of CNN classifier such as AlexNet, ZFNet, and LeNET for the recognition of characters and improve the accuracy. The experimental results achieved the testing accuracy 99.73% with better precision, and recall factors. The Ranjana dataset was utilized for training and testing purpose. However, recognition of compound characters can be done in the proposed work. Dingari et al. [13] focused on the printed characters recognition by using Meetei-Mayek script. They have applied segmentation process on images to channelize the images into white and black channels. Then classification of images can be done by using distinct CNN models such as ResNet, VGG-19, and VGG-16. Although, the system achieved the highest accuracy (99.27%) with VGG-16 model, however, the extraction of local languages is a major concern with the current approach. These papers collectively demonstrate that deep learning techniques, including CNNs and RNNs, have shown promise in the recognition and conversion of Devanagari script into an editable Word format. While challenges remain, ongoing research in this field is focused on improving accuracy, handling variations, and advancing the state of the art in Devanagari script recognition and conversion.

I. DATASET DESCRIPTION

The Devnagari Handwritten character dataset (DHCD) [14] dataset consist of more than 92k images with 46 distinct classes of characters. A well-known dataset for handwritten Devanagari character detection and classification is the DHCD, which is utilised in computer vision and

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

machine learning research. Many South Asian languages, including Hindi, Marathi, Sanskrit, and Nepali, are written in the Devanagari script. The DHCD dataset is useful for character recognition, optical character recognition (OCR), and handwriting analysis algorithm development and testing. Here is a brief description of the DHCD dataset:

Character Set: The DHCD dataset contains a comprehensive collection of characters from the Devanagari script. This includes consonants, vowels, digits, and special characters commonly used in languages like Hindi and Nepali. The characters encompass the full range of Devanagari script, making it suitable for a wide array of applications.

Handwriting Variations: One of the key features of the DHCD dataset is its diversity in terms of handwriting styles. It includes samples of Devanagari characters written by various individuals, leading to variations in size, style, and quality. This variability is essential for training and testing robust recognition models.

Image Format: The characters in the DHCD dataset are typically provided as images. These images are scanned or captured from handwritten text and are often in grayscale. Each character is usually isolated within its image.

Training and Testing Sets: Usually, there are two primary subsets of the dataset: a training set and a testing set. Machine learning models are trained on the training set, and their performance is assessed on the testing set.

Usage in Machine Learning: Researchers and practitioners use the DHCD dataset to develop and assess machine learning models for Devanagari character recognition. Convolutional Neural Networks (CNNs) and other deep learning techniques are commonly employed for this task.

Research Contributions: The DHCD dataset has contributed to the development of handwriting recognition systems for various South Asian languages that use the Devanagari script. It has been used in academic research to advance the field of character recognition.

Figure. 3. Sample Image of DHCD dataset

Figure 3 indicates the sample image of the DHCD dataset where the dataset contains more than 92k image records with 46 characters and digits (size 32 X 32 px) which belongs to distinct set of languages. The dimension of the csv file is 92000 * 1025 than represents the 1024 input features and grayscale values between 0 to 255. The dataset is arranged in training and testing folders with 46 subfolders.

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

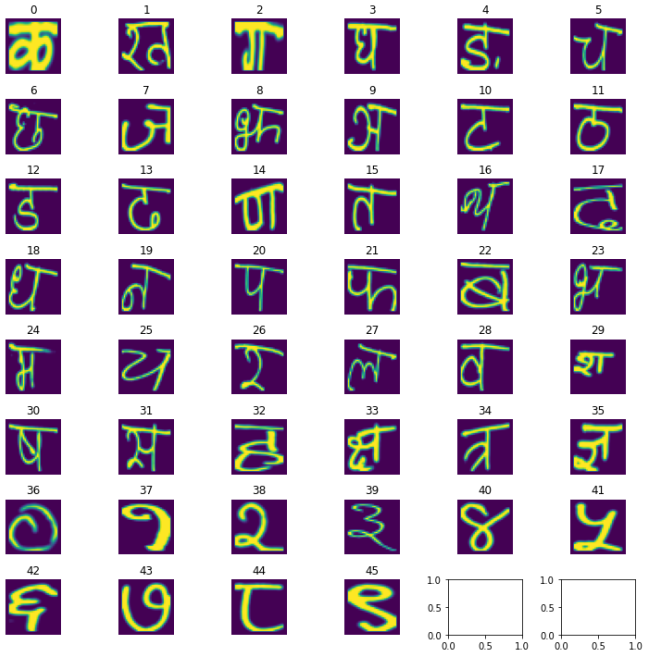

Figure. 4. Train images with Hot Encoding

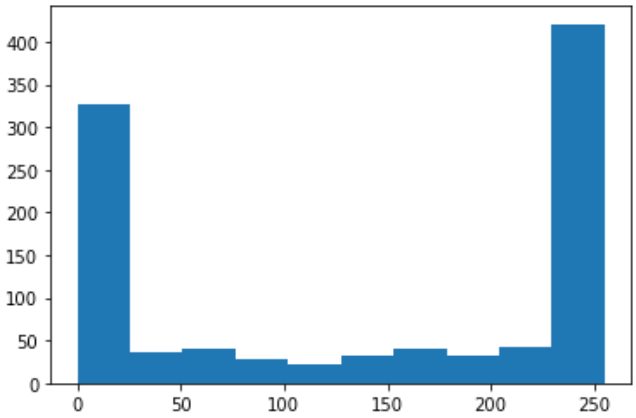

Figure. 5. Data histogram

One-hot encoding is commonly used to represent categorical variables, such as labels in image classification tasks. Figure 4 represents the train image sample with hot encoding of the images. Figure 5 represents the data histogram for the image data that indicates the graphical representation of the data points which are organized with-in a range.

II. PROPOSED METHODOLOGY

To convert Devanagari script into an editable Word format using CNN and RNN, we build a custom machine learning model for character recognition, word segmentation, and language understanding. There are several steps (Figure 6) to perform the task such as:

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

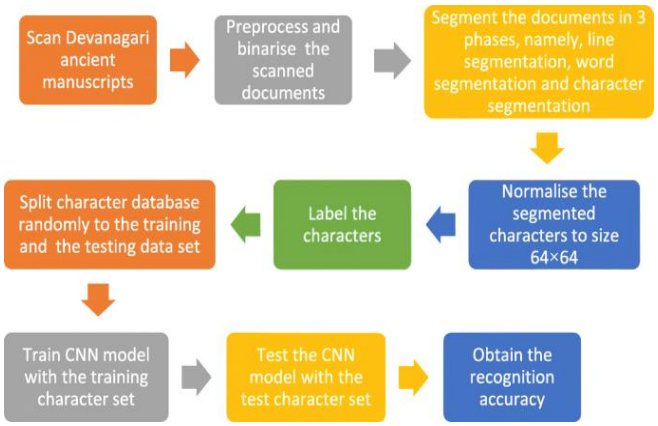

Figure. 6. Process for the proposed approach

Step 1: Data Collection and Preparation: Gather a substantial dataset of Devanagari script samples, which includes both printed and handwritten text. Organize the dataset into individual characters and words, ensuring proper labeling for supervised learning.

Step 2: Character Recognition with CNN: Train a CNN model for character recognition. The model can use popular deep learning frameworks like TensorFlow or PyTorch. The CNN should take an image of a Devanagari character as input and output the recognized character. Augment the dataset with variations in fonts, styles, and quality to improve model robustness. Implement techniques for handling ligatures and conjunct characters, as they are unique to the Devanagari script. Figure 7 indicates the character recognition after label 0. There are some steps which may improve the overall performance of the system.

- Split the dataset into training and validation sets.
- Design a CNN architecture suitable for character recognition.
- Train the CNN on the training data, using the Devanagari characters as labels.
- Use categorical cross-entropy as the loss function.
- Optimize the model using an algorithm like SGD or Adam.

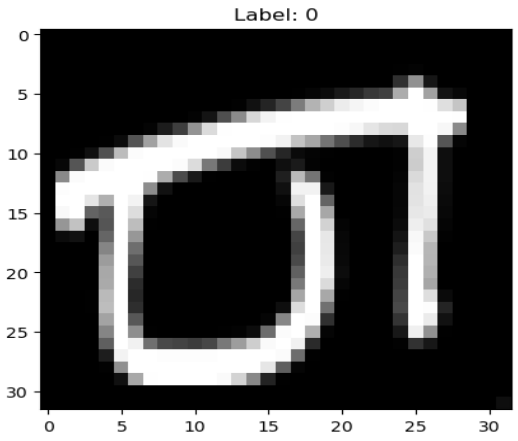

Figure. 7. Character recognition after label 0

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Step 3: Word Segmentation: Develop a word segmentation module to split sentences into words. This module should be aware of Devanagari script-specific rules for word boundaries. Apply the trained character recognition model to recognize characters in a given sentence. Steps included in this process are:

- Use the trained CNN to recognize characters in a given sentence.
- Develop an RNN model that identifies word boundaries based on character sequence.
- Train the RNN for word segmentation using labeled data, if available.

Step 4: Language Understanding with RNN: Train an RNN model for language understanding. The RNN model can be based on Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) architectures. The language understanding model should consider the context and semantics of the text, which is particularly important for Devanagari script due to the influence of contextual factors on character and word choices.

Step 5: Text Formatting and Export: After character recognition and language understanding, format the recognized text properly. This involves applying punctuation and formatting rules specific to Devanagari. Once the text is correctly formatted, export it into an editable Word document. Python libraries can be utilized for docx to create Word documents programmatically.

- Create a programmatic interface to Microsoft Word or use a library like python-docx to create Word documents programmatically.
- Insert the formatted text into the Word document.
- Export the Word document with editable Devanagari text.

Step 6: Post-processing and Error Correction: Implement post-processing steps to correct errors that may have occurred during character recognition or language understanding. Offer a user-friendly interface for manual corrections and editing, especially when dealing with handwritten text or complex scripts.

- After the initial conversion, implement post-processing steps to correct errors and ensure the text is correctly formatted.
- Offer a user-friendly interface for manual corrections and editing, especially when dealing with handwritten text or complex scripts.

Step 7: Evaluation and Improvement: Evaluate the performance of the proposed model by using a validation dataset or real-world data. Fine-tune the model to improve recognition accuracy and language understanding. Continuously update the model with additional data to expand its recognition capabilities.

- Evaluate the overall system's performance using a diverse dataset of Devanagari script samples.
- Fine-tune the system based on feedback and performance metrics.

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Step 8: Deployment: Deploy the Devanagari script recognition and conversion system as a standalone software application, a web service, or as part of an OCR tool. It's important to note that creating a robust Devanagari script conversion system using CNN and RNN is a complex task that may require expertise in machine learning, deep learning, and natural language processing. Additionally, ongoing maintenance and updates are necessary to keep the system accurate and adaptable to various Devanagari script variations and writing styles.

III. Results and Discussion

The obtained results of converting Devanagari script into an editable Word format using CNN and RNN would depend on various factors such as specific implementation, the quality of the training data, the architecture of the neural networks, and several other parameters like accuracy, error correction, generalization, user experience, and word layouts. Figure 8 shows the distribution of characters of output classes.

Figure. 8. Visualization of distribution of characters

Figure. 9. Character visualization after pre-processing

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The pre-processing of the dataset may include distinct steps for character image. These steps can vary depending on the quality and source of the image that can includes resizing that ensure the character image is of a consistent size, grayscale conversion which convert the image to grayscale to simplify processing, reduce memory usage, and improve contrast, noise reduction that uses Gaussian blur or median filtering to remove noise and artifacts from the image. Figure 9 indicate the character visualization after pre-processing of the dataset. Figure 10 represents the labelled output class encoding for distinct character classes. One-hot encoding is typically used for categorical variables with no inherent order or hierarchy. In this encoding, each unique class label is represented as a binary vector. Each binary vector has a length equal to the number of unique classes, with a 1 in the position corresponding to the class and 0s in all other positions.

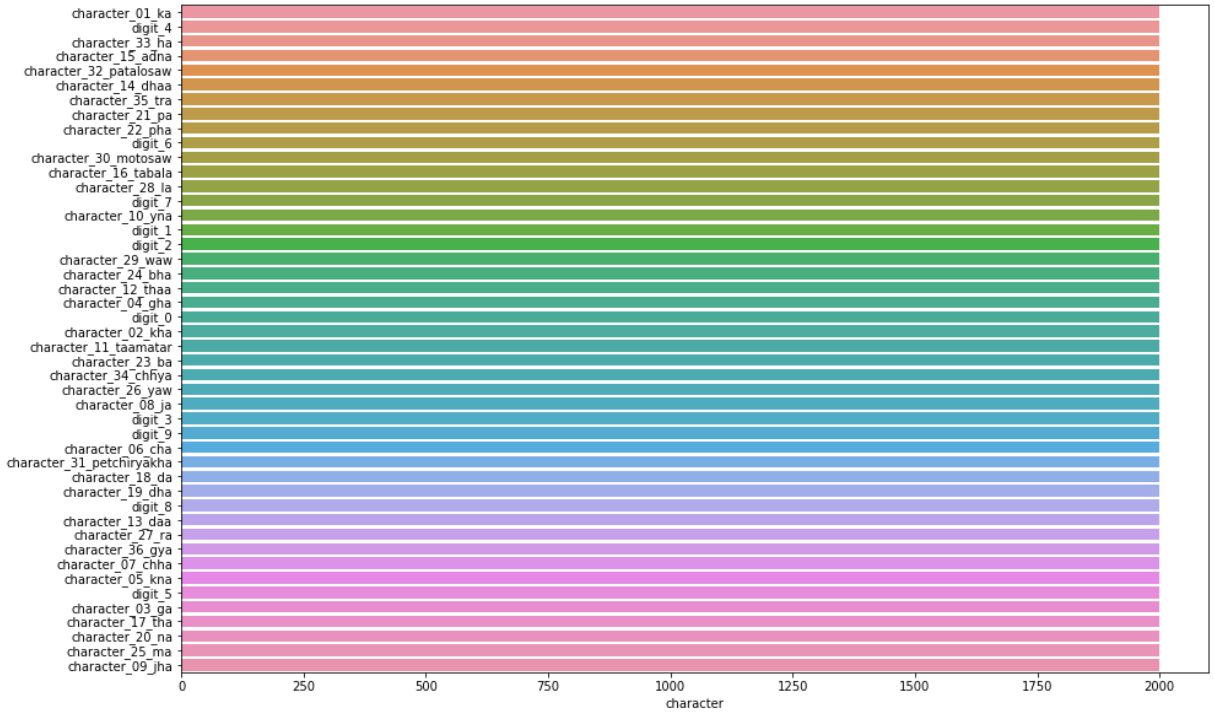

Figure. 10. Encoding of the Labeled output classes

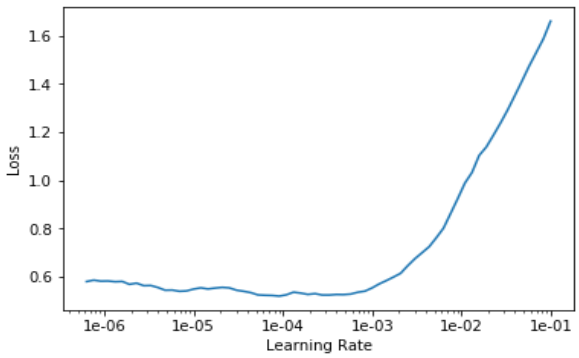

Figure. 11. Learning rate Vs Loss function

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Fig. 12. Training and validation losses using CNN

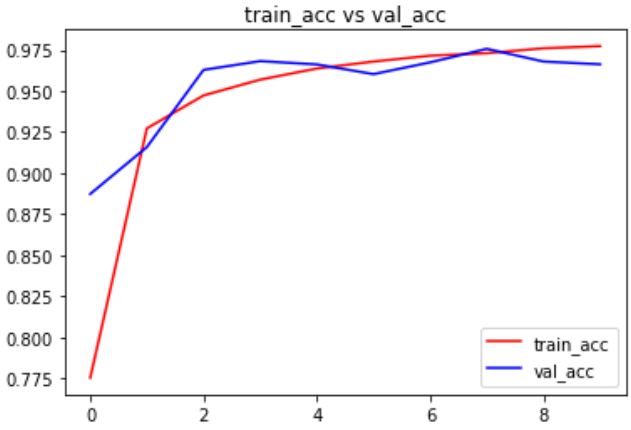

Figure. 13. Training and validation accuracy using CNN

Figure 11 shows the learning rate Vs loss function. The learning rate is a hyperparameter that determines the step size at which a model's parameters are updated during training. The loss function, also known as the cost function or objective function, quantifies how well the model is performing on the training data. Figure 12 represents the training and validation losses using CNN. The training loss is a measure of how well a machine learning model is performing on the training data. It is computed during each training iteration (or batch) and represents how close the model's predictions are to the actual target values (ground truth) for the training examples in that batch. The validation loss is a measure of how well the model generalizes to data that it has not seen during training. It is computed using a separate validation dataset that is distinct from the training data. Training and validation accuracy are important metrics used to assess the performance and generalization of machine learning and deep learning models. These metrics provide insights into how well a model is learning and how effectively it can make predictions on both the training data and unseen data.

Received: July 29, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

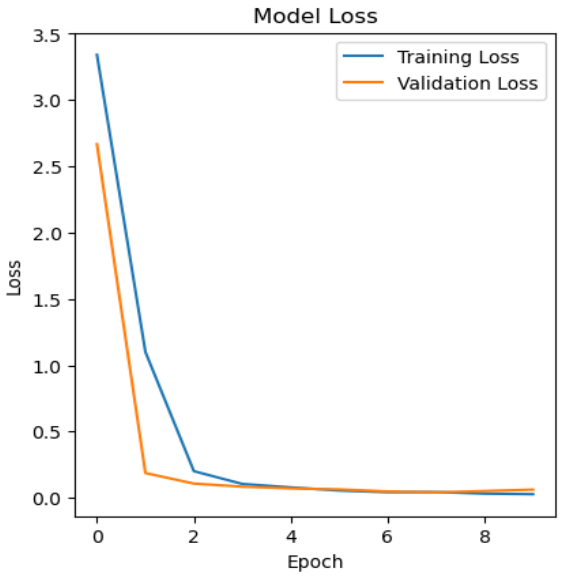

Figure. 14. Training and validation losses using RNN

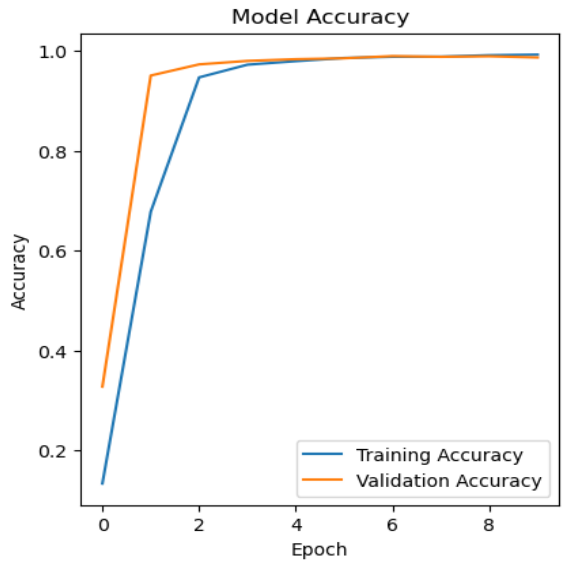

Figure. 15. Training and validation accuracy using RNN

Figure 14 and Figure 15 shows the training and validation losses and accuracy by using RNN algorithm. The model shows the better validation accuracy and less training losses.

CONCLUSION

The conversion of Devanagari script into an editable Word format using CNN and RNN represents an exciting field of research and application. While it comes with its challenges, it holds the promise of digitizing and preserving valuable Devanagari content, making it more accessible, editable, and adaptable for various purposes, including research, translation, and content dissemination in the digital age. As technology continues to advance, the accuracy and usability of such systems are likely to improve, contributing to the broader field of character recognition and document digitization.

Received: July 29, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

REFERENCES

[1] M. Bisht and R. Gupta, "Offline handwritten Devanagari modified character recognition using convolutional neural network," Sādhanā, vol. 46, no. 1. Springer Science and Business Media LLC, Feb. 03, 2021. doi: 10.1007/s12046-020-01532-w.

[2] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Off-Line Handwritten Character Recognition of Devnagari Script," Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). IEEE, Sep. 2007. doi: 10.1109/icdar.2007.4378759.

[3] Ms. S. Tantarpale and C. N. Deshmukh, "An Approach to Pattern Recognition for Identification of Devnagari Script Based on Fingertips and Palm," Journal of Physics: Conference Series, vol. 2327, no. 1. IOP Publishing, p. 012032, Aug. 01, 2022. doi: 10.1088/1742-6596/2327/1/012032.

[4] S. M. Pande and B. K. Jha, "Character Recognition System for Devanagari Script Using Machine Learning Approach," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). IEEE, Apr. 08, 2021. doi: 10.1109/iccmc51019.2021.9418028.

[5] V. P. Agnihotri, "Offline Handwritten Devanagari Script Recognition," International Journal of Information Technology and Computer Science, vol. 4, no. 8. MECS Publisher, pp. 37–42, Jul. 16, 2012. doi: 10.5815/ijitcs.2012.08.04.

[6] S. Roy, N. Das, M. Kundu, and M. Nasipuri, "Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach," Pattern Recognition Letters, vol. 90. Elsevier BV, pp. 15–21, Apr. 2017. doi: 10.1016/j.patrec.2017.03.004.

[7] M. Z. Alom, P. Sidike, T. M. Taha, and V. K. Asari, "Handwritten Bangla Digit Recognition Using Deep Learning." arXiv, 2017. doi: 10.48550/ARXIV.1705.02680.

[8] A. El-Sawy, H. EL-Bakry, and M. Loey, "CNN for Handwritten Arabic Digits Recognition Based on LeNet-5," Advances in Intelligent Systems and Computing. Springer International Publishing, pp. 566–575, Oct. 18, 2016. doi: 10.1007/978-3-319-48308-5_54.

[9] Attigeri, S., "Neural Network based Handwritten Character Recognition system", International Journal of Engineering and Computer Science, pp. 23761–23768, Volume 7 (08), 2018.

[10] Adnan, M., Rahman, F., Imrul, M., Al, N., & Shabnam, S., "Handwritten Bangla character recognition using inception convolutional neural network", Int. J. Comput. Appl, 181(17), 48-59, 2018.

[11] M. Jangid and S. Srivastava, "Handwritten Devanagari Character Recognition Using Layer-Wise Training of Deep Convolutional Neural Networks and Adaptive Gradient

Received: July 29, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Methods," Journal of Imaging, vol. 4, no. 2. MDPI AG, p. 41, Feb. 13, 2018. doi: 10.3390/jimaging4020041.

[12] J. Bati and P. Raj Dawadi, "Ranjana Script Handwritten Character Recognition using CNN," JOIV : International Journal on Informatics Visualization, vol. 7, no. 3. Politeknik Negeri Padang, p. 984, Sep. 10, 2023. doi: 10.30630/joiv.7.3.1725.

[13] V. V. S. Dingari, G. Kosanam, D. S. S. Chavatapalli, and C. N. Devi, "A Printed Character Recognition System for Meetei-Mayek Script Using Transfer Learning," Lecture Notes in Electrical Engineering. Springer Nature Singapore, pp. 519–528, Oct. 03, 2023. doi: 10.1007/978-981-99-4713-3_50.

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

Received: July 29, 2025