Developing Real-time Digitization System using Deep Learning for Handwritten Documents

1st Atharv Kailas Joshi
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
atharvjosh@gmail.com

2nd Siddhesh Puranik
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
puraniksiddesh@gmail.com

3rd Niranjan Ram
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
niranjanram3657@gmail.com

4thSonal Balpande
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
ssabalpande@apsit.edu.in

5thJayashree Jha
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
jpjha@apsit.edu.in

***Abstract*—The purpose of this research is to automate the existing manual system for digitization of the Devanagari script with the help of an automated approach which reduces time and workload. The automation of digitization enables easy access, manipulation, and longer storage of this data. Unlike Western languages such as English written in the Roman script, Devanagari does not have common or easy-to-use digitization tools. Thus, the proposed approach aims to fill this gap and make the digitization process more efficient for the Devanagari script. This work will focus on two languages in particular: Hindi and Marathi. Both of these use virtually the same script with no major differences in spelling or pronunciation. This work is to serve as a proof of concept for the digitization of the Devanagari script. The goal of this work is to improve the accuracy and efficiency of Devanagari handwritten text recognition by utilizing the most effective techniques and configuring a Convolutional Neural Network (CNN). By leveraging state-of-the-art approaches and customizing them for the unique characteristics of the Devanagari script, this study aims to significantly enhance the recognition rate and automate the digitization process, allowing for easier access, manipulation, and storage of this important data. This approach will use the SHABD dataset (Complete Hindi characters) [12] which is a cutting-edge and enormous open dataset with 384 unique classes of Devanagari characters. The complete dataset contains 304,150 grayscale images making it an enormous dataset. This work also plans on developing a real-time app to convert handwritten text into editable files. This will also help provide a better user experience and accessibility to digitising such an important but overlooked script.**

***Index Terms*—CNN, Devanagari, Digitization, Handwritten character recognition, Deep neural network, Convolutional neural network**

I. Introduction

Devanagari script is the most used script in India [14]. With the growth in the digital and computer world as well as certain good initiatives by the government like “Make in India” and “Digital India”, documents also need to be digital. In the Information-age or as some call it the New Media Age one needs to work in the digital constructs. However, it is not always possible, especially in the more rural areas or even with the people whom some might consider old-school. There is an abundance of sources and mediums to digitize the languages that are in Roman script. Indic scripts and especially the Devanagari script lacks behind.

Digitizing old and sometimes obsolete documents written in the Devanagari script can indeed be a challenging task, especially when an automated system for doing so is not available. The growth in technology and the demand for digitization at a rapid rate has led to an increased need for automated systems that can recognize and convert Devanagari script into digital format. This urges one to have an automated system that would help the overall process by making it considerably cheaper and faster. The Indian civil offices use the Devanagari script more than most other scripts. National Informatics Centre (NIC), courtrooms, registrar offices and small-scale vendors maintain various documents which are available in either handwritten or printed form in Devanagari script [1]. Furthermore, some important documents in Maharashtra like the 7/12 extract are exclusively in Marathi in most cases. The data that is stored is huge both in volume and in terms of the physical space it consumes. This is very inefficient and the process of making it digital is laborious.

The complexity of handwritten characters is greater than that of printed characters due to several reasons:

i. Each individual has a diversified writing style causing substantial deviation in character strokes and overall handwriting resulting in a number of variations.
ii. There exist many letters in Devanagari that are similar in both Form and shape.
iii. The inclusion of ”matra” and consonant clusters (”jod-shabda”) in letters adds complexity to the task of classifying handwritten characters. Therefore, using a general classifier to recognize handwritten characters written by multiple writers may not always be feasible. [1]

II. Related Work

Pande, S.D., et al, in [1], explain how they have used CNN with 4 convolutional, 2 max-pooling, 3 drop out, a flatten and 2 dense layers. This has helped them avoid over-fitting and achieve results more accurately and with less training time. The proposed approach is to implement a DHTR (Devanagari Handwritten Text Recognition) system using the following steps:

1) Segment the input handwritten Devanagari text query image into individual characters.
2) Isolate each character and prepare it for comparison.
3) Assess the resemblance of each isolated character with the characters in the DHTR system's repository.
4) Compare the characters within an anticipated range of resemblance to improve the accuracy of the recognition.

By implementing these steps, the proposed approach aims to accurately recognize handwritten Devanagari text. This approach uses the dataset DHCD [7] and has only 46 classes which do not include the characters that have 'matra' in them. In this work, a proposed system is presented that achieves better performance in handwritten character recognition. The system uses statistical features computed from isolated characters in the input query image, and a character class repository of limited diversity is utilized, which includes very few homogeneous patterns. The proposed approach outperforms other handwritten character recognition systems in terms of accuracy and performance. Additionally, a conflict resolution step is incorporated which reduces ambiguity in the recognition of words. Overall, this work presents a promising approach to improving the accuracy and efficiency of handwritten character recognition systems.

Deore, S.P., Pravin, A. in [2], achieve higher recognition rates with the Devanagari Handwritten Dataset. This work purely focuses on the recognition of the characters and improving the efficiency of doing it. The results of this study demonstrate the effectiveness of incorporating aggressive runtime data augmentation and regularization techniques, such as Dropout and Batch Normalization, to improve the performance of the DHCRS model. The authors were able to achieve the best classification accuracy of 96.55% using a fine-tuned VGG16 architecture on their trivial database. Furthermore, the proposed approach was shown to perform better on two Indic scripts, indicating the potential for broader application and improved accuracy in character recognition tasks.

K. Dutta, P. Krishnan, M. Mathew and C. V. Jawahar in [3], use a hybrid CNN-RNN architecture to find faster recognition of the Devanagari Handwritten Character Datasets. In this study, the authors not only proposed a new dataset called the IIIT-HW-Dev dataset but also benchmarked it using a CNN-RNN hybrid network. They demonstrated the effectiveness of fine-tuning the network using both synthetic and real handwritten data, resulting in improved word recognition performance. By applying their fine-tuning strategy on the CNN-RNN hybrid network, the authors were able to achieve state-of-the-art results on the RoyDB dataset. This suggests that incorporating additional training data and fine-tuning techniques can significantly enhance the performance of handwriting recognition systems.

In [4] by Patil, Jyoti A., and Dr S. R. Patil, an off-line handwritten Devanagari character recognition system with Neural Network has been described. The paper describes a system that can be applied to recognize handwritten names, read documents, and convert any handwritten document into machine-readable text. The system has achieved a high accuracy level of 98%, making it effective in accurately recognizing and digitizing handwritten text.

I. Kissos and N. Dershowitz, in [5], discussed the utilization of machine learning techniques to enhance OCR (Optical Character Recognition) accuracy. The approach taken by the researchers in this study involved the combination of features from a language model, OCR model, and document context to improve the accuracy of OCR word recognition. By correcting misspelled OCR words using these combined features, they were able to create a reliable spelling model that can be used for different languages and domains. This method allowed for greater accuracy in OCR word recognition, which is an essential component of many applications involving text analysis and processing.

P. K. Sonawane and S. Shelke presented an approach in [6] that involved fine-tuning a CNN using transfer learning. Their method achieved a validation accuracy of 94.49% and a test accuracy of 95.46%. The training curves demonstrated that the training accuracy exceeded 90% in just three epochs. The results indicated that transfer learning is a more effective option for faster and better training with a limited number of training samples.

Seba Susan* and Jatin Malhotra in [15] have used basic 2 layer CNN to recognise characters in Devanagari script. They used MNIST dataset which has 46 classes and got an accuracy of 97%. They suggested using deep learning techniques to analyze image quadrants by utilizing the hidden state activations obtained from convolutional neural networks trained on five different image quadrants.

III. Problem Definition

In India, clerical work is primarily done in English, but this is mostly true for urban areas. In rural areas, government offices still predominantly use local languages, mainly Hindi and Marathi, and many of these documents are handwritten. Digitizing this handwritten text poses a challenge as it requires significant human effort and time, and there is a higher risk of errors due to the complexity and variability of the Devanagari script. Therefore, the problem statement can be formulated as follows: "The user needs to convert a piece of document written in the Devanagari script into a digital format."Digitization

means converting the handwritten or typed physical document into an editable digital format.
This is also a problem for native writers who are old-school and want to keep on working with pen and paper. The digitization of their work requires time and money. Having an automated way to digitize the document gives the user more time to work and costs less as compared to human labour which in turn has a chance of having errors.
That said, there has never been a ready-to-use and easily accessible real-time application on the market. Our work hopes to fix that problem by making a cross-platform real-time app that will digitize handwritten Devanagari text into an editable file. The issues mostly come in the segmentation process as well as the recognition process. The reconstruction of the words is also a big task for making a robust system.

Fig. 1. Character segmentation

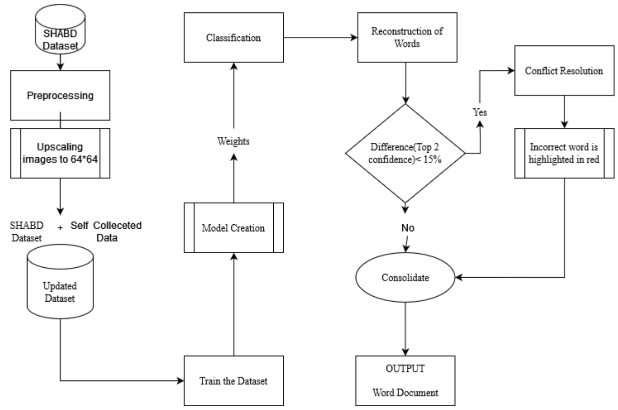

Fig. 2. Classification

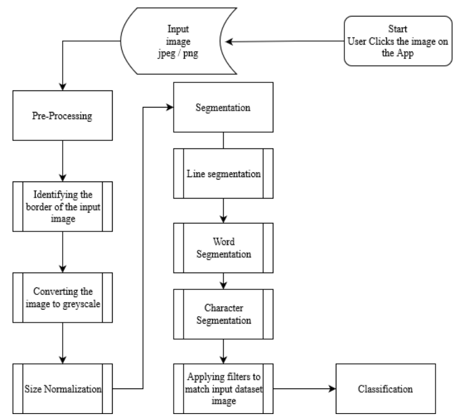

Fig. 3. Segmentation

IV. Proposed System Architecture

The proposed workflow suggests the following steps :
The user uploads the image in real-time and that is taken as the input for the model. The pre-processing of data is followed by the segmentation of characters from the words [8]. The Pre-processing goes as follows, the first step is to identify the borders of the input image. Then convert this image to greyscale to have better visibility. The size of the image will be normalized to a specific size. The noise is removed and skews are corrected. This is to be followed by classification using CNN [2]. First, the document is segmented into lines followed by word segmentation. The proposed approach uses a vertical histogram projection technique to perform word segmentation. This method involves counting the number of foreground pixels along each column of the image, resulting in an array of values equal to the number of columns in the image. [13]. Following this, filters are applied to match the input from [12] which are images that serve as the dataset to train the data. This is then followed by Classification. In this process, the weights from the model are then applied and the characters are recognized. After identifying the characters, the reconstruction of words is done. If the difference between the confidence of the top 2 results is greater than 15%, it is forwarded to conflict resolution which will highlight the text in red colour which can be fixed by the user on their own leaving us with the desired output text, which will be consolidated in the final step and converted to an editable word file. The approach mentioned configures a CNN (Convolutional Neural Network) in a manner that allows for the easy addition of new classes of Devanagari characters, including those containing consonant clusters. This can be done without significant changes to the existing model, thereby expanding its scope.

V. Results

The proposed system has yielded favourable results on our machines. The accuracy obtained is on average 84% and validation accuracy of about 97%. Despite having to train a large number of classes, the results were favourable. The model is trained on 384 classes as opposed to 46 classes in [1] and is trained in an average of 4 hours. The model was trained on 3 different machines 2 of which had integrated graphics and one having a dedicated graphics card.

This table shows the training time of our model done on different machines with different specifications. All of the machines had Windows 11 as their OS.

PC	Specifications	Accuracy	Training Time(minutes)
PC 1	i7 9th gen 16 GB RAM	0.9742	130
PC 2	i3 11th gen 8 GB RAM	0.9818	230
PC 3	i5 10th gen 8 GB RAM	0.9761	226

During the training process, the model's accuracy improves as the number of epochs increases. In our proposed system, a significant increase in accuracy during the first few epochs was observed, after which the accuracy growth plateaued. The validation accuracy also increased steadily during the initial epochs before stabilizing at around 97%. These observations suggest that the model was able to effectively learn the features of the Devanagari characters and achieve high accuracy with relatively fewer epochs.

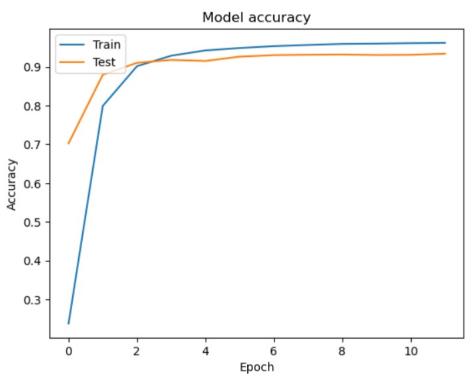

Fig. 4. Epoch accuracy

The currently trained model has high accuracy. The summary of the CNN can be seen in the below table:

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 32)	896
conv2d_1 (Conv2D)	(None, 64, 64, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
conv2d_3 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_1 (Dropout)	(None, 16, 16, 64)	0
conv2d_4 (Conv2D)	(None, 16, 16, 128)	73856
conv2d_5 (Conv2D)	(None, 16, 16, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_2 (Dropout)	(None, 8, 8, 128)	0
conv2d_6 (Conv2D)	(None, 8, 8, 256)	131328
conv2d_7 (Conv2D)	(None, 8, 8, 256)	262400
conv2d_8 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 256)	0
dropout_3 (Dropout)	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 512)	2097664
dropout_4 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_5 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 459)	117963

Total params: 3,617,771
Trainable params: 3,617,771
Non-trainable params: 0

Fig. 5. CNN Summary

VI. Conclusion

There are a lot of handwritten Devanagari texts written in India now and thousands are currently being written. Most of them are in government offices or in academic settings. The process of digitization is mostly done manually with human effort which is prone to errors and is slow. There are a lot of causes that can wipe out a big chunk of information in the form of handwritten data like natural disasters such as floods, tsunamis, fires, pests and earthquakes or even human errors. Over time, information can be forgotten and lost forever if there are no digital copies available. Therefore, it is crucial to digitize handwritten texts. Our approach, along with previous methods, has demonstrated that digitizing documents can be done at a low cost with high accuracy and minimal errors. Our proposed method performs better than other handwritten character recognition systems in terms of accuracy and performance. Our approach is one step closer towards a perfect system with characters having 'matra' being recognized and the next foreseeable step is to have a model that recognizes 'jodshabda' or consonant clusters as well.

VII. Future Scope

The present research demonstrates the feasibility of achieving high accuracy in digitizing Devanagari's handwritten text, covering a larger set of characters than previously explored in related works [1], [2]. This work explores 384 classes, surpassing previous studies in this area. Further improvements could involve expanding the dataset to include characters with consonant clusters or 'jodshabda,' which are connected characters. Another potential application for this work is to integrate it into text-to-speech software, providing blind individuals with the ability to read handwritten Devanagari text.

In addition, the proposed approach can be extended to recognize handwritten text in other languages, as the basic architecture and techniques used are not specific to the Devanagari script. The use of data augmentation and transfer learning can further improve the performance of the model with limited training data. Moreover, the model can be integrated into mobile applications for on-the-go digitization, enabling users to capture and digitize handwritten text in real time. These potential applications demonstrate the practicality and usefulness of the proposed approach in various domains.

References

[1] Pande, S. D., Jadhav, P. P., Joshi, R., Sawant, A. D., Muddebihalkar, V., Rathod, S., Gurav, M. N., & Das, S. (2022). "Digitization of handwritten Devanagari text using CNN transfer learning – A better customer service support". Neuroscience Informatics, 2(3), 100016. https://doi.org/10.1016/j.neuri.2021.100016

[2] Deore, S.P., Pravin, A. "Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset". Sādhanā 45, 243 (2020). https://doi.org/10.1007/s12046-020-01484-1

[3] K. Dutta, P. Krishnan, M. Mathew and C. V. Jawahar, "Offline Handwriting Recognition on Devanagari Using a New Benchmark Dataset," 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), 2018, pp. 25-30, doi: 10.1109/DAS.2018.69.

[4] Patil, Jyoti A., and Dr S. R. Patil. "Optical Handwritten Devnagari Character Recognition Using Artificial Neural Network Approach." International Journal of Innovations in Engineering Research and Technology, vol. 5, no. 3, 2018, pp. 1-5.

[5] I. Kissos and N. Dershowitz, "OCR Error Correction Using Character Correction and Feature-Based Word Classification," 2016 12th IAPR Workshop on Document Analysis Systems (DAS), 2016, pp. 198-203, doi: 10.1109/DAS.2016.44.

[6] P. K. Sonawane and S. Shelke, "Handwritten Devanagari Character Classification using Deep Learning," 2018 International Conference on Information, Communication, Engineering and Technology (ICICET), 2018, pp. 1-4, doi: 10.1109/ICICET.2018.8533703.

[7] https://archive.ics.uci.edu/ml/datasets /Devanagari+Handwritten+Character+Dataset

[8] https://nanonets.com/blog/ocr-with-tesseract/

[9] https://likegeeks.com/python-image-processing/

[10] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. TensorFlow: a system for Large-Scale machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).

[11] F. Chollet & others, Keras, Available at: https://github .com /fchollet /keras, 2015.

[12] Dataset used: https://www.kaggle.com/datasets/sushantshetty/shabd-complete-hindi-characters-dataset

[13] https://towardsdatascience.com/segmentation-in-ocr-10de176cf373

[14] https://en.wikipedia.org/wiki/Devanagari

[15] Susan, S., Malhotra, J. (2020). Recognising Devanagari Script by Deep Structure Learning of Image Quadrants. DESIDOC Journal of Library Information Technology, 40(05), 268-271. https://doi.org/10.14429/djlit.40.05.16336