Handwritten Hindi Character Recognition using Deep Convolutional Neural Network

Jay Deep Singh
Data Science and Artificial Intelligence
211020421
jay21102@iiitnr.edu.in

Neeraj Manhar
Data Science and Artificial Intelligence
211020429
neeraj21102@iiitnr.edu.in

Srijan Oraon
Data Science and Artificial Intelligence
211020453
srijano21102@iiitnr.edu.in

Abstract: The development of automation has given rise to research fields such as optical character recognition, intelligent character recognition, etc. There has been some improvement in handwritten Devanagari text recognition due to intensive research and development. This study focuses on a survey of various techniques used for recognition of handwritten Hindi characters, covering almost all aspects of Hindi character recognition. The study discusses relevant techniques of preprocessing, feature extraction, and classification, addressing limitations of existing techniques and providing beneficial directions of research in this field.

1. Introduction

Handwritten recognition is an important research in the field of data analysis and recognition aimed at converting text to machine readable text.. In India, Hindi is one of the most commonly spoken languages, and the ability to recognize Hindi handwritten characters can have a significant impact on natural language processing and digital accessibility for Hindi speakers.

The development of a machine learning model that can accurately recognize handwritten Hindi characters is a complex task that requires a robust, scalable, and efficient model capable of learning from a large dataset of labeled handwritten characters and performing well on unseen data. The successful development of such a model will enable greater accessibility to digital content for Hindi speakers, and have significant implications for automating manual work and improving the efficiency of natural language processing.

The Hindi handwritten character recognition project aims to contribute to this research area by developing a machine learning model that can accurately recognize handwritten Hindi characters. The project will involve collecting and preprocessing a large dataset of labeled

handwritten Hindi characters, followed by training and evaluating a machine learning model using state-of-the-art techniques in deep learning.

The project will address various challenges in Hindi handwritten character recognition, including the variation in handwriting styles among individuals, the complexity of the Hindi script, and the need for a scalable and efficient model capable of handling large amounts of data and providing real-time predictions.

1.1 Background and Motivation

The widespread use of Devanagari script in India and Nepal, coupled with the increasing demand for automation, has motivated the need for developing an accurate and efficient system for recognizing handwritten Devanagari characters. This project aims to address this need and make digital content more accessible for Devanagari speakers.

1.2 Problem statement

Handwritten Hindi character recognition using machine learning is important for digitizing handwritten documents, speech recognition systems, and language translation tools for Hindi speakers. Accurate recognition of handwritten characters is essential for creating searchable digital archives and improving accessibility to digital content.This will also help children to practice a better and recognizable handwriting .A robust and efficient model will preserve and disseminate Hindi language and culture.

2. About Dataset

Context

The project focuses on developing a model that can accurately recognize handwritten Devanagari characters, consisting of 36 characters and 10 digits used in India and Nepal. Devanagari is distinguishable from other languages due to its salient features like no capitalization and aligning a horizontal bar at the top of the script. The dataset includes various examples of handwritten characters that are used to train the model. However, since these characters are handwritten, there may be some amount of error due to sloppy or illegible writing, which makes achieving 100% accuracy challenging. The main challenge of the project is to develop a robust model that can accurately recognize the characters even if they appear quite similar. The successful development of this model will have significant implications for various applications, including natural language processing and making digital content more accessible for Devanagari speakers.

Content

A total of 46,000 images are contained in our training dataset, with 2000 examples of each character. Each image is composed of 32x32 pixels and 3 color channels, which is converted into a single color channel. The test set, comprising of 9200 images (200 per character), and the training set, comprising

36800 images (800 per character), are split in an 85/15 ratio.

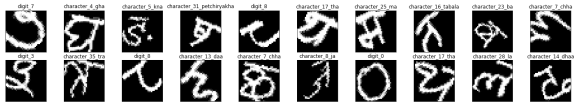

Fig1 : Some characters from the dataset

3. Related Works

In recent literature, there are major researches conducted in handwritten classification problems.

R. K. Gupta [1] presented a comprehensive survey of the techniques used for recognition of handwritten Devanagari characters. It covers different aspects of Devanagari character recognition, including preprocessing, feature extraction, and classification, and provides a critical analysis of the current state-of-the-art methods.

Singh and Lehri [2] designed an offline handwritten recognition using neural network]. The authors emphasized on the preprocessing part and then it was fed to the neural network, thereby achieving an accuracy of 93%..

Jawahar et al. [3] proposed a model using principal component analysis (APC) followed by the support vector classification. This PCA analysis reduces the dimensions and speeds up the algorithm. This model achieved an overall accuracy of 96.7%..

Acharya [4] proposed deep learning based large scale handwritten Devanagari character recognition. Jangid and Srivastava [5] developed a handwritten Devanagari character recognition system using layer-wise training of DCNN and adaptive gradient methods. ISIDCHAR database was used to evaluate the system. The results of layer-wise-trained DCNN were observed favorable..

4. Algorithm Used

CNN

CNNs and Neural Networks (NN) share learnable parameters and bias terms. CNNs have three types of layers: convolution, pooling, and fully connected layers. NNs have fully connected layers but learning so many parameters in image classification problems can result in slow computation and overfitting. CNNs solve this issue by utilizing parameter sharing and sparsity of connections, allowing for efficient and effective learning in image recognition tasks. That's why we used CNN for this project.

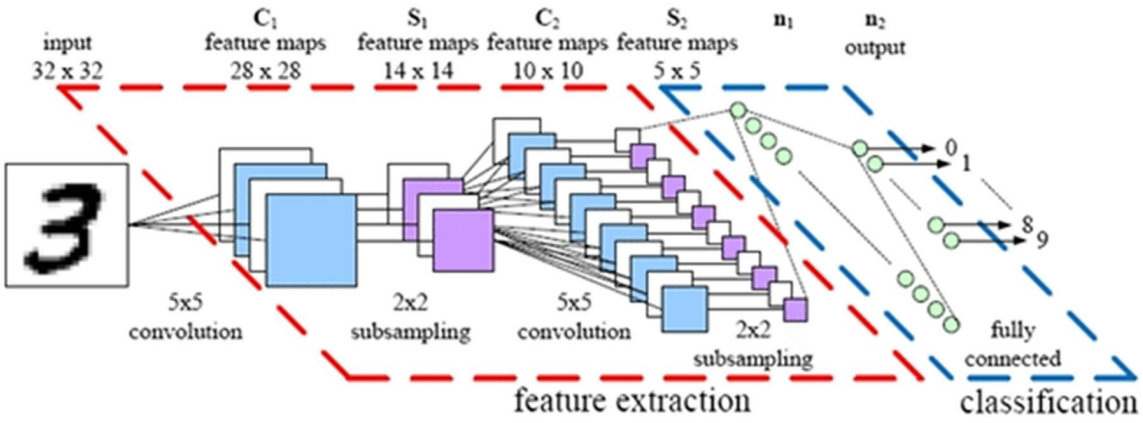

Fig 2 : Basic Architecture of Convolutional Neural Networks

5. Proposed Work

A. Model Overview and Architecture

The model architecture consists of four convolutional layers followed by three fully connected layers. The convolutional layers have 32, 32, 64, and 64 filters, respectively. Each convolutional layer is followed by batch normalization, max pooling, and a rectified linear unit (ReLU) activation function.

The fully connected layers consist of two hidden layers with 128 and 64 nodes, respectively, and a final output layer with 46 nodes (one for each Devanagari character class). The activation function used in the hidden layers is ReLU, and the output layer uses the softmax activation function.

The optimizer used in the model is the Adam optimizer, and the loss function used is categorical cross-entropy. The model is trained for 10 epochs with a batch size of 32. Initially, we were planning for 25 epochs, but due to training time we shortened it to 10 epochs.

B. Code(Model) Description

The first step is to import the necessary libraries, including NumPy, Keras, TensorFlow, and OpenCV. The DevanagariHandwrittenCharacterDataset is imported from the local system.

Next, an ImageDataGenerator object is created to apply data augmentation techniques such as rotation, shifting, scaling, and shearing to the training data. The training data is loaded using the flow_from_directory method of the ImageDataGenerator object.

After loading the training data, the code defines a list of labels for the Devanagari characters. It then loads the test data and creates an ImageDataGenerator object for it.

The model is defined as a sequential model using the Keras Sequential API. The five convolutional layers are added to the model, followed by three fully connected layers. The model is compiled using Adam optimizer.

Finally, the model is trained using the fit method of the model object, with the training and validation data provided as arguments. The training progress is saved in the res variable.

6. Results and Discussions

The details of the model are as follows:

Model:	**Sequential**	
Layer	**Output Shape**	**Param**
conv2d (Conv2D)	(None, 30, 30, 32)	320
batch normalization (BatchNormalization)	(None, 30, 30, 32)	128
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 32)	9248
batch_normalization_1 (BatchNormalization)	(None, 13, 13, 32)	128
max_pooling2d_1 (MaxPooling 2D)	(None, 7, 7, 32)	0
conv2d_2 (Conv2D)	(None, 5, 5, 64)	18496
batch_normalization_2 (BatchNormalization)	(None, 5, 5, 64)	256
max_pooling2d_2 (MaxPooling 2D)	(None, 3, 3, 64)	0
conv2d_3 (Conv2D)	(None, 1, 1, 64)	36928
batch_normalization_3 (BatchNormalization)	(None, 1, 1, 64)	256

max_pooling2d_3 (MaxPooling 2D)	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 128)	8320
batch_normalization_4 (BatchNormalization)	(None, 128)	512
dense_1 (Dense)	(None, 64)	8256
batch_normalization_5 (BatchNormalization)	(None, 64)	256
dense_2 (Dense)	(None, 46)	2990

Table 1 : Model Details

The training and testing accuracy plot:

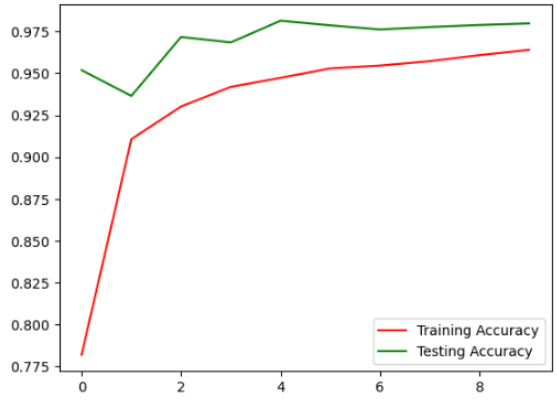

Fig 3 : Training vs Testing Accuracy

The training and validation loss per epoch:

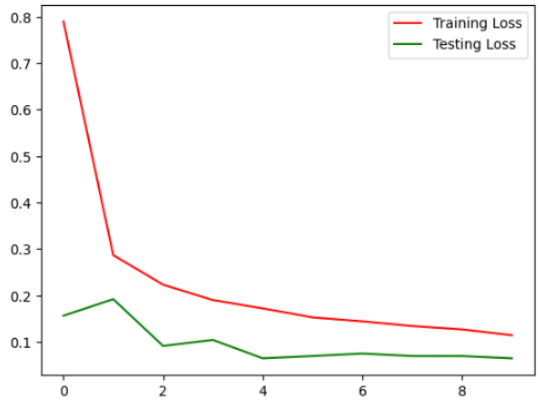

Fig 4 : Training vs Testing Loss

Conclusion

In conclusion, we implemented the CNN model for image classification using the Devanagari Handwritten Character Dataset and achieved high accuracy on both the training and validation datasets, indicating that it is a good fit for the data. This provides a promising approach for character recognition tasks in Devanagari script. Also, this project demonstrates the potential of CNNs for image classification tasks

and highlights the importance of properly preparing and augmenting the dataset to improve model performance. Additionally, there are opportunities for future work to improve the model's generalization performance as well. With further development and refinement, this model has the potential to be applied to real-world scenarios like ‘WarnaMLA’ for kids and people with disabilities.