Devanagari Character Recognition Using ConvNet

Akhila V U[1], Vinya Vijayan[2]
[1]Dept of Computer Applications
[2]Assistant Professor, Dept of Computer Applications
[1, 2] College of Engineering, Trivandrum, Kerala, India

Abstract- *Character recognition is one of the significant ingredients in most of the applications developed these days. In day to day life, mobile phones have become an indispensable part of our life. We rely on our mobile phones for everything. As a result, the developers started developing applications that would help us with our day to day courses. There are applications which have to recognise handwritten texts. People have developed systems which will recognize handwritten English characters. When it comes to regional languages there are only a few works, and most of them lacked accuracy. This paper proposes a system a new model for recognising handwritten Devanagari scripts. The vital component of this model is ConvNet. Like any other machine learning models, this model also makes use of a massive dataset which is the major reason for the remarkable accuracy of the system. This model gives an accuracy of about 96%.*

Keywords- Devanagari Script, Deep Learning, ConvNet, Character Recognition, Pattern Recognition.

I. INTRODUCTION

Devanagari is one of the significant writing systems in the world with 14 vowels and 33 consonants constituting 47 primary characters in total. This script has been adopted by many other languages. Our national language Hindi also shows a close connection with the Devanagari scripts. A lot of research is being done under this field. We have used DHCD dataset, which is one of the best datasets available for Devanagari character recognition till date. ANN-based recognition systems do the feature extraction by making use of character selection with vertical intensity and other properties. Adaptive gradient method is also widely used for character recognition.

In most of the systems, only a few relevant features are considered while training the system. The most prompt system would be the one which will be able to perform considerably well even in isolated and unusual situations. English character recognition has moved a step ahead by recognising even words. This paper introduces a model which uses the ConvNet algorithm as its foundation for developing a model for classification. The system can be expanded further for recognising Devanagari words as well. With an ample amount of data in the dataset, the same model can be trained for recognising words also.

II. SYSTEM OVERVIEW

Devanagari character set and number system are depicted in the figure. The major challenge is to develop a model which can recognize these characters when new handwritten characters are given as input. Data is the fuel for any machine learning solution. The system belongs to a multi-class-classification problem-based solution. It needs a lot of data as the number of classes for classification increases, otherwise, it will have an adverse impact on the accuracy of the system.

The dataset for the system is collected from the UCI machine learning repository. It contains 92000 grayscale images of 46 classes including the 10 digits class. The image size is 32x32 pixels but the actual image size is 28x28 pixels with 2 pixel padding in each direction. The image background is black(0) and the character is written in white(255).

The 36 consonants of Devanagari script are shown in Figure 1, and the ten numerical scripts are shown in Figure 2. The main challenge in the recognition system is the complexity of the script and here it is overcome by an effective deep learning model which used ConvNet algorithm.

क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट	ठ
ड	ढ	ण	त	थ	द	ध	न	प	फ	ब	भ
म	य	र	ल	व	स	ष	श	ह	क्ष	त्र	ज्ञ

Fig 1. Devanagari Consonants

०	१	२	३	४	५	६	७	८	९

Fig 2. Devanagari Numerals

III. TECHNOLOGY STACK

A. DEEP LEARNING

Deep learning is a machine learning strategy where the features are automatically selected extracted. It differs from the traditional neural networks due to the presence of multiple hidden layers. The number of hidden layers is a critical factor in deciding the optimality of the system.

The architecture of the model plays a prominent role in determining the feasibility of the system. We have developed a new model based on ConvNet which is a deep learning algorithm.

B. CONVNET

ConvNet is the widely used deep learning algorithm which works on the basis of the convolution operation. The convolution neural networks are the best way to generate an image classification model. The introduction of hidden layers to the neural network itself brought about a tremendous change. In addition to that, the convolution layers in CNN started performing automatic feature extraction. The most difficult task during image classification was to identify the important features of image processing. CNN broke all the boundaries of preconceived notions. This paper suggests an architecture for the CNN model which will classify handwritten Devanagari characters to the appropriate classes.

C. KERAS

Keras library is the most effective library system in python for deep learning model designing. It runs on the top of tensorflow which is a deep learning framework in python. Keras library provides inbuilt functions to perform convolution and all other ConvNet processes. Here we are using the sequence to sequence model from keras library.

D. BOTTLE

The bottle is a python web-framework used for developing web interfaces. It is a better way of building UI for machine learning codes. All the code fragments are made into methods for easy invoke in the web interface. The bottle is compatible with other web languages including CSS.

IV. SYSTEM DESIGN

A. PREPROCESSING

As the preprocessing step in each iteration, few steps are carried out to evade the issue of overfitting. Overfitting is a situation in which the system gets overtrained with the training data. In such cases the noise in the dataset will affect the efficiency of the system. The predictions made with such models will fail to generalise to the future data. It will also respond with less accuracy for the testing datasets and unknown inputs. The following are the steps for preventing overfitting of the data:

- Rotation_range is a degree value at which the image is rotated.
- width_shift_range & height_shift_range: The shift range is the value of the fraction which is used to randomly translate a picture is transformed vertically or horizontally.
- Rescaling is done to reduce the complexity of processing. Original images will be having pixel values between 0-255, so to make it suitable to better processing, the values are converted into some fraction between 0 and 1 by scaling with 1/255 factor.
- Shearing range is used for performing shearing transformations which are used to slant the shape of images.
- The zoom range is the value at which the zooming inside the image is done.

B. ARCHITECTURE

In the convolutional neural network, the first layer will be the convolutional layer. It has got 5 nodes which serve as the input nodes.

The first one is the number of filters which is a vector of weights used to convolve the input. Here we are using 32 filters. The second input is the size of the filter used and here it is 3x3. The third input is strides which is a numerical value which decides the movement of the selection area for the convolution.

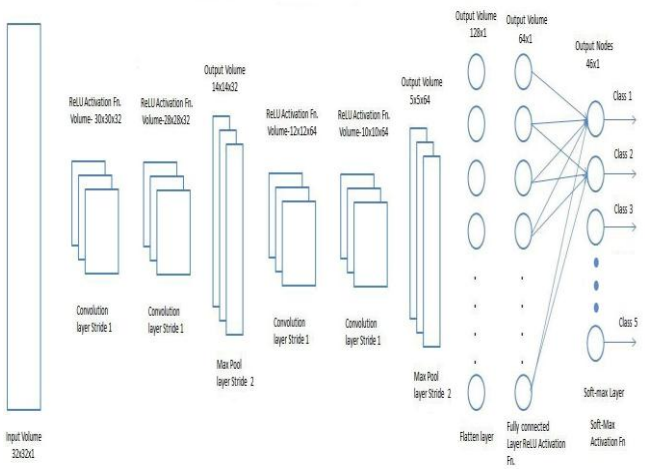

Fig 3. ConvNet Architecture

In the initial layer, we are applying thirty-two filters each of which has a size of 3 X 3 and the striding the matrices by a value 1. The activation function is the soul of the artificial neural network. It is very important to choose the activation function wisely as the wrong choice will adversely affect the functioning of the whole system. The activation function that would better meet our requirements is Relu. The Relu activation function replaces all negative values with zeros and leaves positive values as intact. We also have to mention the type of images used for training i.e. whether they are grayscale or RGB image. The convolution operation between image matrix and filter matrix results in the convolved feature map which will be the input of the next layer and it will be having 30x30 pixel.

The second layer is also a convolutional layer which will be performing the same operation as the first layer. Here the output feature map will be having 28x28 pixels. The third layer is the pooling layer where the pooling of the feature map is performed. Here we are performing 2x2 max pooling with stride 2. The pooling operation eliminates low character features and takes only the necessary features from the image. This helps to reduce the complexity of the processing.

The fourth layer is again a convolutional layer which performs all the convolution operation, but here the number of filters is 64. Which will be having a feature map of 12x12 as output. The fourth convolution will be performing the same operation as that of the third one. The result will be a 10x10 feature map.

The fifth operation is max-pooling which will reduce the size of the last input to 5x5. The next operation is flattening which converts the matrix into a vector. At last, we will use a dropout value to eliminate some of the connections amongst the layers. A fully connected network will result in overfitting. In order to avoid overfitting, we have removed 20% of the connections.

The last layer is the output layer which uses a softmax function as its activation function to get a prediction with a probability value. The softmax activation function is an apt choice here as this is a multi-class classification scenario.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 30, 30, 32)	320
conv2d_2 (Conv2D)	(None, 28, 28, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 14, 14, 32)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	18496
conv2d_4 (Conv2D)	(None, 10, 10, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 5, 5, 64)	0
dropout_1 (Dropout)	(None, 5, 5, 64)	0
flatten_1 (Flatten)	(None, 1600)	0
dense_1 (Dense)	(None, 128)	204928
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 46)	2990

Total params: 281,166
Trainable params: 281,166
Non-trainable params: 0

Fig 4. Summary of ConvNet Architecture

The whole network architecture is illustrated in the Figure 3 where the classes indicate each Devanagari characters.

C. TRAINING

When the image dataset is given the above model triggers the data and learn from the data by automatically extracting features. 72000 Grayscale images are used for training and the rest 20000 images are used as the test and validation data. After training a model can be saved for further classification.

D. PREDICTION

The system has a user interface which is designed using bottle framework. The user will be able to browse the image and after uploading the image the system will show the predicted character on the web interface itself. On the command line, we will be able to see the actual Devanagari character. Few screenshots of the application is shown below:

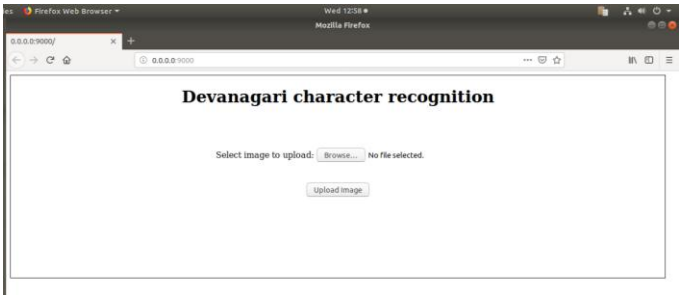

Fig 5. Interface in Bottle framework

Fig 6. Prediction

E. RESULTS

The classification has got an accuracy of 96% and most of the input handwritten Devanagari characters are accurately classified. The summary of the ConvNet operation is shown on Figure 4.

V. CONCLUSION

Digitalisation is ruling each stream and optical character recognition has found its place in almost all areas. Devanagari character recognition is a challenge as compared to other scripts due to its complexity. Building an efficient model will help to incorporate it to other OCR models. In the project, I have developed a convolutional neural network based Devanagari character recognition model which has attained a fair accuracy of 96%. Few characters look almost similar and that is a challenge as the handwritten Devanagari character will be different in style from person to person. This model recognises almost all the newly supplied handwritten characters.

As an extension, vowel dataset can be created and which will help to recognise the complete Devanagari character recognition. Currently, vowel data set is not available for research or project purpose. Vowel recognition will be more challenging and it will be a good research area too.

VI. ACKNOWLEDGMENT

We are grateful to the Almighty for the good health and well being that were necessary to complete this work. We would also like to express our sincere gratitude to Prof. Jose T Joseph, Head of department and Prof. Baby Syla L, Department of Computer Applications, College of Engineering Trivandrum imparting their knowledge with us and for all the help and support they have given us from the start of the work till its completion.

REFERENCES

[1] R. M. K. Sinha, H. Mahabala, "Machine recognition of Devnagari script", *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 8, pp. 435-441, Aug. 1979.

[2] S. Palit, B. B. Chaudhuri, "A feature-based scheme for the machine recognition of printed Devanagari script" in Pattern Recognition Image Processing and Computer Vision, India, New Delhi:Narosa Publishing House, pp. 163-168, 1995.

[3] H. Ma, D. Doermann, "Adaptive Hindi OCR using generalized Hausdorff image comparison", *ACM Trans. Asian Lang. Inf. Process.*, vol. 2, no. 3, pp. 193-218, 2003.

[4] V. J. Dongre, V. H. Mankar, "A Review of Research on Devnagari Character Recognition", *International Journal of Computer Applications*, vol. 12, no. 2, pp. 0975-8887, November 2010.

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, "Handwritten digit recognition with a backpropagation network", *Advances in Neural Information Processing Systems 2 (NIPS 1989) Denver CO*, 1990.

[6] K. Oyebade, O. Ebenezeri, A. Khashman, "Deep Learning in Character Recognition Considering Pattern Invariance Constraints", *International Journal of Intelligent Systems and Applications*, vol. 7, no. 7, pp. 1, 2015.

[7] L. B. Saldanha, Ch. Bobda, "An embedded system for handwritten digit recognition", *J. Syst. Archit.*, vol. 61, no. 10, pp. 693-699, Nov. 2015.

[8] A. Gupta, M. Srivastava, C. Mahanta, "Offline handwritten character recognition using neural network", *2011 IEEE Int. Conf. on Computer Applications and Industrial Electronics (ICCAIE)*, 2011.

[9] D. C. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, "Convolutional Neural Network Committees for Handwritten Character Classification", *2011 International Conference on Document Analysis and Recognition*, pp. 1135-1139, 2011.