International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 02 | Feb 2022 www.irjet.net p-ISSN: 2395-0072

Devanagari Character Recognition using Image Processing & Machine Learning

Rahul S Narsing

Department of Computer Science & Engineering, Babasaheb Naik College of Engineering Pusad Pin:445204

Abstract - In terms of character recognition there are several papers reported and most of them are for English character. This paper focused on Devanagari character recognition from images. Devanagari script is used for many languages such as Sanskrit, Marathi, Nepali and Hindi. Lot of work has been done in character recognition and lot of work is to be done. Devanagari script should be given a special attention so that analysis of this language can be done effectively. This paper presents an approach for recognition of handwritten Devanagari characters, Total Fifty Eighth handwritten characters each having (vowels=220, consonant=2000, digits=2000) resulting in 94640 images are used for this experimentation. The final accuracy is around 90%. The handwritten characters are scanned and on every individual character's image transform is applied so as to get decomposed images of characters. Character recognition provides an alternative way of converting manual text into digital format and reduces the dependence of man power.

Keywords —Devanagari, Character Recognition, Feature Extraction, Machine Learning, Neural Network

1.INTRODUCTION

This Handwritten character recognition is gaining popularity for many years and attracting researchers for the purpose of potential application development. These Potential applications reduce the cost of human efforts and save the time. In the last few centuries English Character Recognition has been comprehensively studied and progressed to a level, sufficient to produce technology driven applications. Unfortunately, this is not same case for Indian languages which are complex in terms of structure and computations. Nowadays the speedily growing computational power may provide a solution for implementation of Indian Character Recognition methods. Digital document processing is achieving popularity for various application to office and library automation, bank and postal services, publishing houses and communication technology. Devanagari is composed of two Sanskrit word —Deva‖ and Nagari‖. Deva means God and Nagari means city. The Devanagari script is used for over 120 languages, including Hindi, Marathi other languages and dialects, making it one of the most used and adopted writing systems in the world. The Devanagari script is also used for classical Sanskrit texts. Image processing is a tool through which the required data from any image can be extracted very easily. Here we have used Image processing as data extractor from the database images. We have written a Python code which firstly, convert the normal images into gray image and later the pixel value of each gray images is stored into csv file. Csv file contains the Pixel values of each images which is later used as dataset. Artificial Neural Network is one of the techniques that can be applied to efficiently recognize the Devanagari characters. Artificial Neural Network is simply a network of interconnected nodes that provides classification and regression abilities to the machine. Here we have tested total 8 classifier and depending on the accuracy score we have selected one network to train our model.

II. IMPLEMENTATION

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

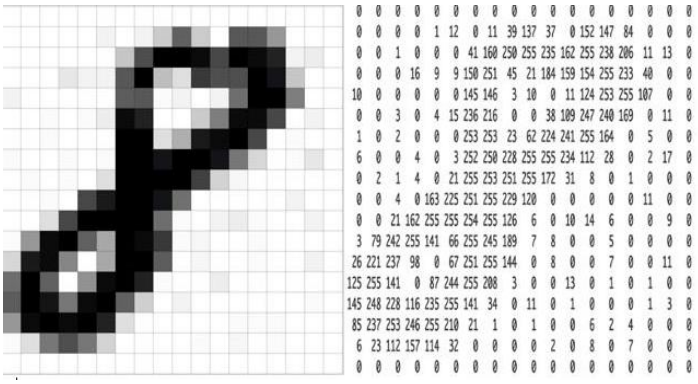

Fig.1 Image with pixel value

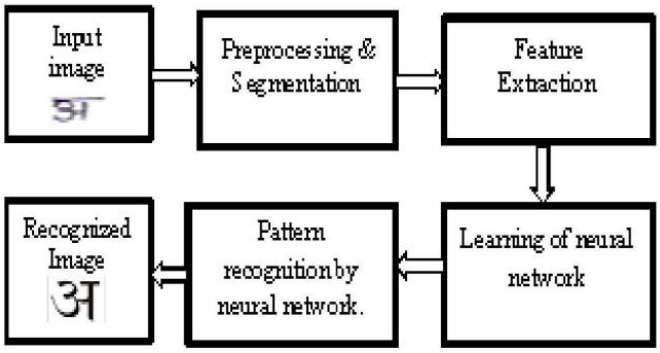

Fig.2 Block Diagram

A. PREPARATION OF DATASET

The data is purely images and we have to convert it into another form so it can be easily fed into Neural Network. Here the dataset is prepared by extracting the important feature from images. We have used a python program to extract the feature from images. First it will convert the normal image into gray level images. Now the gray images contain the pixel value between 0 to 255 i.e. for bright pixel value is 0 and for dark it is 255, depending on the intensity level the pixel value is varies between 0 and 255. Fig. 1 indicates the pixel value of only one image. Similarly, the pixel value of the all images is extracted and arranged into excel file known as CSV file. Our dataset consists of Devanagari Consonant, Vowels and Digits. There are total around 2,000 images for each consonants and digits. Therefore, total (2,000*(36+10)) =92,000 images are for digits and consonant. The vowels contain around 200 images of each. Hence the total no on images in dataset is around94,000. The dimension of each images is 32*32. So, there will be total (32*32) =1,024-pixel values of each image. Now the pixel value of each image is saved into CSV file. The CSV file contain 94,000 row and 1024 column

B. SELECTION OF ALGORITHM

```
C:\Users\Apps\lib\site-packages\sklearn\ensemble\forest.py:246: FutureWarning: The
default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
                Classifier  Test_Score  Train_Score  Fit_Time  Score_Time
0        RidgeClassifier    0.436539     0.847394  0.851513    0.050930
1           BernoulliNB     0.509990     0.555099  0.196350    0.075389
2            GaussianNB     0.407010     0.457818  0.237228    3.432099
3    ExtraTreeClassifier    0.311904     1.000000  0.180513    0.025195
4  DecisionTreeClassifier   0.382574     1.000000  3.753235    0.015675
5        NearestCentroid    0.551211     0.595506  0.094148    0.066344
6    KNeighborsClassifier   0.721809     0.839090  0.975255   30.187497
7    ExtraTreesClassifier   0.582469     1.000000  0.797307    0.052084
8  RandomForestClassifier   0.546502     0.997835  1.046088    0.051594
C:\Users\Apps\lib\site-packages\sklearn\utils\deprecation.py:125: FutureWarning: You
are accessing a training score ('train_score'), which will not be available by
default any more in 0.21. If you need training scores, please set
```

Fig.2 Score of all algorithms

Selection of proper algorithm is very important in terms of high accuracy performance of Neural Network. Here we have selected total eight algorithm and calculated the "Test Score" and "Train Score" of all the eight algorithms on the sample dataset. The algorithms are Bernoulli NB, Gaussianan NB, Ridge Classifier, Nearest Centroid, Extra Tree Classifier, K- Neighbors Classifier, Decision Tree Classifier, Random Forest Classifier. Out of these eight we selected three algorithms to find the accuracy of sample dataset depending on the better "Test" and "Train" score. The three algorithms are K- Neighbors Classifier, Random Forest Classifier and Extra Tree Classifier.

A. Accuracy of Selected Algorithm

Out of these three algorithms we have to select only one to train our model with complete dataset. The best way is to select the algorithm which has maximum accuracy. Here we did the same some sample dataset is run on these algorithms and the accuracy of each algorithm is noted. Now depending on the accuracy, the algorithm with maximum accuracy is use to train the final model

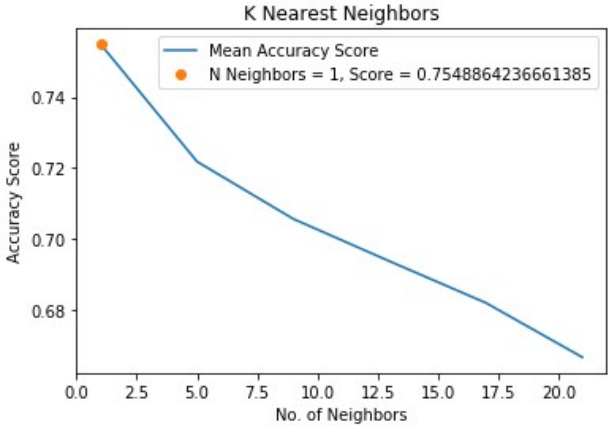

fig. 3.1 K Nearest Neighbors

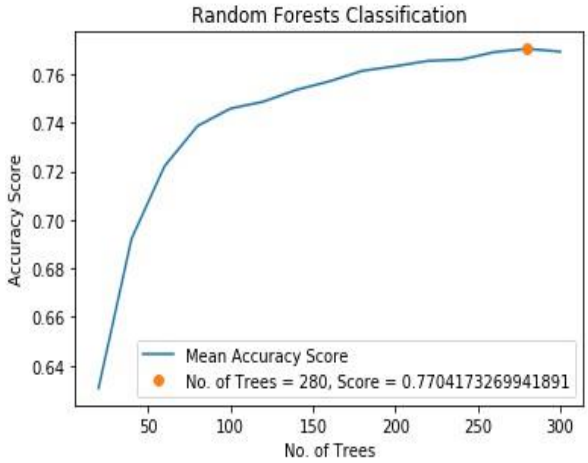

fig. 3.2 Random Forests

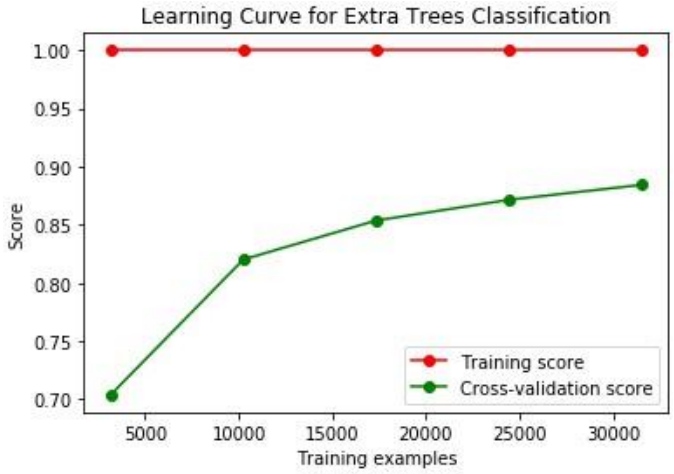

fig. 3.3 Extra Tree

Now from the graph it is clear that the accuracy of "Extra Tree Classifier" is higher than the other two algorithms. Hence, we will use "Extra Tree Classifier" to train our model to find the accuracy with the complete dataset.

III. RESULT AND DISCUSSION


```
Python 3.6.7 |Anaconda, Inc.| (default, Oct 24 2018, 09:45:24) [MSC v.1912 64 bit
(AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.0.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/mission/dcr.py', wdir='C:/mission')
Using TensorFlow backend.
tf.estimator package not installed.
C:\Users\Dinesh\Anaconda3\lib\site-packages\keras_preprocessing\image.py:492:
UserWarning: grayscale is deprecated. Please use color_mode = "grayscale"
  warnings.warn('grayscale is deprecated. Please use '
6
```

Fig 4 Output Screenshot

As illustrated in the Fig. *4* we can see that, we are truing to recognize the devanagri character of number „6'. This image is indicated by number 5549 in the prediction folder.

In the actual program we point it to the specific folder and the image number. The code then compares it with the dataset for the features and tries to match it. If a match is found it is the recognized and is indicated

IV. FUTURE SCOPE

One of our goals is to use the Camera Module as a detection unit. We intend to use the RPi 3 Camera as an image processing application to identify the sentences by taking images at particular intervals and send it to the user. Hence, a learning algorithm will be able to detect long and complex words.

V. CONCLUSION

A lot of research is to be done to handle the Challenges in Devanagari Character Recognition. There are big challenges in handwritten character recognition due to different style of writer. Recent research is not directly concern to the characters, but also words and phrases, and even the complete documents. For the character recognition, neural networks and their combinations are used as the powerful tools. Character recognition, segmentation and classification can be used in an integrated manner for the high reliability and accuracy. This paper covers methodology used for handwritten character recognition using different features and different classifiers. Literature survey tells about the past research work done in Devanagari handwritten character recognition.

References

1. Ashok Kumar Pant, Prashnna Kumar Gyawali, Shailesh Acharya, —DeepLearningBased Large Scale Handwritten Devanagari Character||, International high performance building conference at Prude, 2014.
2. Vikas J Dongre, Vijay H Mankar, —A Review of Research on Devnagari Character Recognition International Journal of Computer Applications (0975 – 8887) Volume 12– No.2, November 2010
3. Kasturi Upasani, P. V. Baviskar, —A REVIEW OF DIFFERENT TECHNIQUES FOR DEVNAGRI SCRIPT RECOGNITION USING IMAGE PROCESSING||, Global Journal of Advanced Engineering Technologies Volume 5, Issue 1- 2016 ISSN (Online): 2277-6370 & ISSN (Print):2394-0921
4. Miss. Gayatri H. Khobaragade,— Effective Techniques for the Detection,Extraction and Conversion of Devanagari Text from Traffic Panels||, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.5,May- 2015, pg. 314-323.
5. Rajni Kumari, —A Research Proposal on Recognition of Degraded Devanagari Text||,Pankaj Kale, Arti V. Bang,|| Recognition Of Handwritten Devanagari Characters Using Machine Learning Approach||, International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982 Volume-3, Issue-9,Sept.-2015.
6. Divakar Yadav,Sonia Sánchez-Cuadrado, —Optical Character Recognition for Hindi Language Using a Neural-network Approach||, J Inf Process Syst,Vol.9, No.1, March 2013
7. A.Deepika, S.Shalini, M. Sheela, —Handwritten Devanagari Numeral Recognition by Fusion of Classifiers||, J ComputEng Inf Technol Vol: 4 Issue: 2,July 06 2015.