

Devanagari Character Recognition

Shital S. Ramane¹, Moinika T. Dimble², Pooja A. Bipte³

⁴Professor.Swapnil D. Daphal(Guide), Dept. of Electronics & Telecommunication Engineering,
MIT Academy of Engineering, Maharashtra, India

Abstract - In this paper we discuss about Devnagari Handwritten Character Recognition System. To recognize the devnagari handwritten characters different algorithms are used. These algorithms are support vector machine (SVM), convolution neural network (CNN) and k - nearest neighbor (KNN). Along with this we discuss about different phases of character recognition. The main purpose is to develop a devnagari handwritten character recognition system with highest accuracy. We are going to develop this system using python programming language. To get high performance with better accuracy, selection of relevant feature extraction method is most important factor in character recognition system.

Key Words: Support Vector Machine (SVM), K - Nearest Neighbor (KNN), Convolution Neural Network (CNN), Python Programming etc.

1. INTRODUCTION

Handwritten character recognition is one of the extremely interesting research fields. There are different techniques used in handwritten character recognition. Several researches have been done to develop a system which will provide a good accuracy. Handwritten character recognition is an area of pattern recognition which defines the ability of a machine to recognize the handwritten character. Basically, the concept is to design a system which should be intelligent enough to recognize handwritten Marathi characters. It reduces the efforts of typing the Marathi letters, also it converts the hardcopy of written letters into softcopy by recognizing the letters. There are two techniques of character recognition systems which are online and offline character recognition. Here we used an offline handwritten character system. In an offline character recognition system, handwritten characters of the user are available as images [1].

Character recognition is a challenging task because there are different people who have different handwriting styles. Therefore, a large dataset is required to get more accuracy. So, in our project we used 92,000 samples of devnagari characters which includes 36 devnagari letters and 0 to 9 digits, each having 2,000 samples.

2. LITERATURE REVIEW

The literature survey on Handwritten Character Recognition System for that we refer several research papers. Anshul Gupta, Manisha Srivastava and Chitralkha Mahanta proposed an offline handwritten character recognition system using a neural network. They said that a neural network can learn the changes in the input data and also it has a parallel structure so it has a higher rate of computation than other techniques [3]. Ayush Purohit and Shardul Singh Chauhan proposed a Literature Survey on Handwritten Character Recognition System where they explained the phases of a handwritten character recognition system which are as follows [4]:

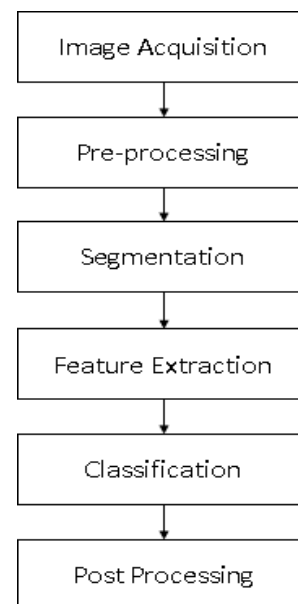


Fig-1: Block Diagram of Character Recognition

The work of each block is described below:

A. Image Acquisition

The very first step in handwritten character recognition is image acquisition. Every system needs input to provide any output. In this step, we provide input to the system as a handwritten image. It can be by any method like scanned image or a photograph.

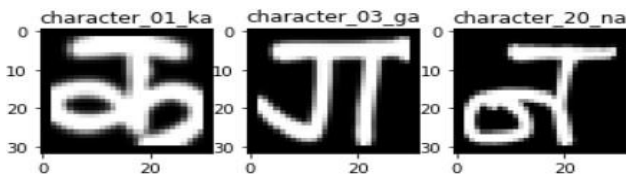


Fig-2: Sample Dataset

B. Preprocessing

This step is basically removing the noise from the image to improve detection accuracy of the image. These noises can be referred to brightening or setting contrast of image, removing distortions, missing points in the words. Preprocessing involves five common steps, size normalization and centering, interpolating missing points, smoothing, slant correction and resampling of points.

C. Segmentation

After image preprocessing the picture is proper and character need to be detected for this process of segmentation is follows. Segmentation is separating the character of an image in form of row and column.

D. Feature Extraction

The goal of feature extraction phase is to extract that pattern which is most relevant for classification. There are different feature extraction techniques like Principal component analysis (PCA), Linear Discriminant Analysis. Scale invariant feature extraction, Gradient based features, Histogram. These features are used to train the features.

E. Classification

when we provide a character image to the system as input, its features are extracted and given as input to trained classifier like AI network or SVM. The classifier compare the input feature with stored pattern and provide the best matching output.

F. Post processing

It refers to the procedure of correcting the misclassified results. It is processing of output from shape recognition. If the system failed to classify or detect the character all the needful action are done.

3. METHODOLOGY

Different algorithms are used for character recognition system. In this Section we discussed about Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Convolution Neural Network (CNN) in brief.

A. Support Vector Machine

Support vector machine (SVM) also used for character recognition system. SVM is based on supervised learning system which is used for analyzing the data. Hyper-plane is

uses for classification in SVM. Support vector machine is very robust algorithm because it tries to separate data with as large as large possible margin. SVM is called as large margin classifier and it need optimization problem [1]. This optimal hyper-plane gives the output. There are different types of kernels are available which are used in SVM. These kernels are Linear, RBF, Quadratic, Polynomial and MLP [5]. If linear kernel is used it separate the data linearly as shown in fig.

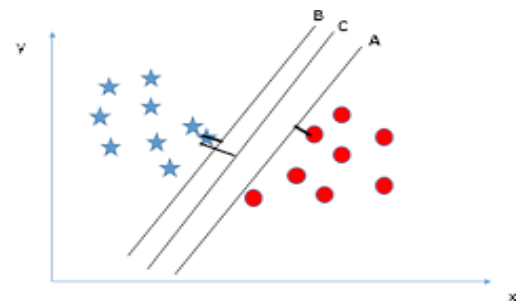


Fig-3: SVM [1]

B. K-Nearest Neighbor

K-nearest neighbor (KNN) is also used in character recognition system. In KNN algorithm we make a batch of k feature point on feature space. Whether the training point belongs to which class is dependent on majority votes. KNN comes under the level of instance base learning. Applications of KNN are YouTube, Amazon etc. KNN is also known as lazy learning algorithm.

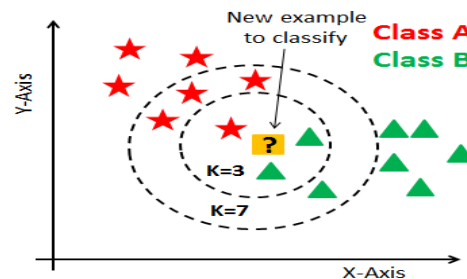
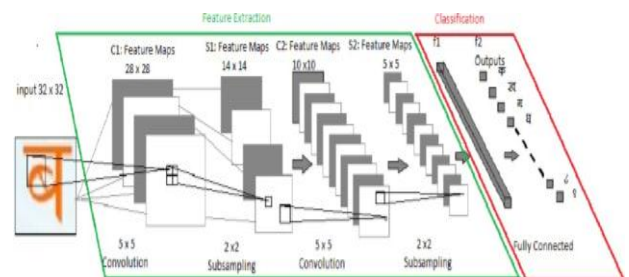


Fig-4: KNN

B. Convolution Neural Network [6]



CNN is kind of deep neural network which were designed from the biologically driven model so what research is found was how a human perceive an image into the brain is in the different layer and that's how this convolution neural network was designed and hence it has been proven

efficient for all image processing pattern recognition kind of application. as shown in above figure.

So, as we can see there are different layers which can design showing here the image is going through. If we have input image of any alphabet this input image is goes through bunch of layers. There are different layers.

1) Input Image: In the input image we have image of RGB in matrix form because computer takes image in matrix form. The CNN layer job is to reduce image size for further processing.

2) Convolution Layer: The first layer in CNN is convolution layer. It has bunch of layers of convolution filters in it. It will take patch from an input image. It will take that patch to the convolution layer and apply set of filters. Convolution layer is set of filters which are apply to given input image. It creates different activation features in that input image. So what we have input as a parameter which will width (W), height(H), depth(D) of input image. We have kernels or filters these could be any dimension e.g. $3 \times 3 \times D$ where D is depth same as input. Number of filters(N) are there which is basically activation filters which generate for a given input dimension. The depth of output of this network is same as number of kernels. Dimension is also dependent on stride .Output of this layer is $W \times H \times N$. Same input image processing through number of filters that processed filter that activated data goes into another layer which is called pooling layer.

a) Stride: To calculate new convolution output there is number of pixels which is called stride and it slide on our filter vertically or horizontally. For example if we put stride of 3 then it will jump after every 3 pixels to calculate convolution output. After the operation of stride size of image almost reduce to half.

b) Padding: Basically the information of image is more in center but there is also information in corners of image .As discussed earlier in output of convolution image is shrink so there is chances of loosing data at corner of image. So not to loose information at corners there is another layer is given at all edges of image and that layer is calling padding. For example if we have image of the size 3×3 then after padding it becomes 4×4 . In passing there is zero padding technique where layer of zeros is added outside of edges of the image.

3) Pooling layer : It is kind of nonlinear down sampling layer. Basically, image is shortened here. Output of this layer is depending on window size. For e.g. 2×2 window, then it will take every 2×2 window in convolution layer and minimize it to one. There are different ways of pooling. There are two types which are Max Pooling and Average pooling. With pooling we are able to reduce the size and reduce computation complexity and memory complexity. This kind of convolution and pooling network is continued to

the fully connected layer. Another technique is used in this layer is padding which is another extra layer to the outside of the image so that information will not missed from outside. There are different ways of pooling. There are two types 1)Max Pooling 2)Average pooling.

a)Max Pooling: Image is made up of pixels and for max pooling patches of pixels are formed. Then from one patch the max value from particular patch is taken and this is called max pooling.

b)Average Poling: As we discussed in max pooling that patches are form in a image .So in average pooling average of each pixel in patch is taken to form output of looking network. So by using pooling technique we get advantages of 1)Reduced computations 2)Required and important feature are extracted.

4)Fully Connected: Objective of fully connected layer is identified or detect the final output categories. Basically, this network is consists of multilayer perceptron and this will use SoftMax activation function. The classification of input images into the various classes is done by fully connected layer.

Table -1: Result Of Different Methodology

Sr. No.	Author Name	Paper Name	Dataset (Size)	Proposed Method	Testing Accuracy
1.	Arora S., Bhattacharjee D., Nasipuri M., L. Malik [7]	A Two Stage Classification Approach for Handwritten Devnagari Characters	50000	Neural Network	89.12%
2.	Arora S., Bhattacharjee D., Nasipuri M., L. Malik, M. Kundu, D. K. Basu [8]	Study of Different Features on handwritten Devnagari Character	1500	Multilayer Perceptron	98.16%
3.	Akanksha Gaur, Sunita Yadav [5]	Handwritten Hindi Character Recognition using K-Means Clustering and SVM	430	SVM (Linear Kernel)	95.86%
4.	Akanksha Gaur, Sunita Yadav [5]	Handwritten Hindi Character Recognition using K-Means Clustering and SVM	430	K-Means Clustering	81.7%
5.	K.V. Kale, S.V. Chavan, M.M. Kazi, Y.S. Rode [9]	Handwritten Devnagari Compound Character Recognition using Legendre moment an ANN Approach	27000	Artificial Neural Network	98.25%
6.	P.E. Ajmire, R.V. Dharaskar, V.M. Thakare [10]	handwritten Devnagari (Marathi) Character Recognition using SVM and MLP	-	Multilayer Perceptron	95.32%
7.	P.E. Ajmire, R.V. Dharaskar, V.M. Thakare [10]	handwritten Devnagari (Marathi) Character Recognition using SVM and MLP	-	SVM	95.82%
8.	Ashutosh Aggarwal, Rajneesh Rani, Reni Dhir [11]	Handwritten Devnagari Character Recognition using Gradient Features	5000	SVM	94%
9.	S. Arora [12]	Complementary Features Combined in a MLP based System to Recognize Handwritten Devanagari Characters	1500	MLP with Neural Network	98.61%
10.	R. Jayadevan, S.R. Kolhe, P.M. Patil, U. Pal [13]	Offline Handwritten Devanagari Script Recognition	More than 1000	Artificial Neural Network	85.78%
11.	A.S. Pant, P.K. Gyawali, S. Acharya [6]	Deep Learning Based Large Scale Handwritten Devanagari Character Recognition	92000	CNN	98.47
12.	Anshul Gupta, Manisha Srivastava, Chitrlekha Mahanta [14]	Offline Handwritten Character Recognition using Neural Network	74000	SVM	
13.	S.P. Ramteke, A.A. Gurjar, D.S. Deshmukh [15]	A Streamlined OCR System for Handwritten Marathi Text Document Classification and Recognition using SVM-ACS Algorithm	66000	SVM	99.36%

4. CONCLUSION

In this paper we present different algorithms used for devnagari character recognition and phases of devnagari handwritten character recognition system. Also we include Comparison between different papers which used different techniques and gives different testing accuracy. From all this we conclude that convolution neural network algorithm gives more accuracy than others.

REFERENCES

- [1] A. Gaur and S. Yadav, "Handwritten hindi character recognition using k-means clustering and svm," 01 2015, pp. 65–70.
- [2] N. Sahu and N. K. Raman, "An efficient handwritten devnagari character recognition system using neural network," in 2013 International Mutli- Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), March 2013, pp. 173–177.
- [3] A. Gupta, M. Srivastava, and C. Mahanta, "Offline handwritten character recognition using neural network," in 2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), Dec 2011, pp. 102–107.
- [4] A. Purohit and S. Chauhan, "A literature survey on handwritten character recognition," International Journal of Computer Science and Information Technology, vol. 7, pp. 1–5, 02 2016.
- [5] A. Gaur and S. Yadav, "Handwritten hindi character recognition using k-means clustering and svm," 01 2015, pp. 65–70.
- [6] S. Acharya, A. K. Pant, and P. K. Gyawali, "Deep learning based large scale handwritten devanagari character recognition," in 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Dec 2015, pp. 1–6.
- [7] S. Arora, D. Bhattacharjee, M. Nasipuri, and L. Malik, "A two stage classification approach for handwritten devnagari characters," in International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), vol. 2, Dec 2007, pp. 399–403.
- [8] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, M. Kundu, and L. Malik, "Study of different features on handwritten devnagari character," in 2009 Second International Conference on Emerging Trends in Engineering Technology, Dec 2009, pp. 929–933.
- [9] K. V. Kale, S. V. Chavan, M. M. Kazi, and Y. S. Rode, "Handwritten devanagari compound character recognition using legendre moment: An artificial neural network approach," in 2013 International Symposium on Computational and Business Intelligence, Aug 2013, pp. 274–278.
- [10] P. Ajmire, R. Dharaskar, and V. M. Thakare, "Handwritten devanagari (marathi) character recognition using svm and mlp," International Journal of Pure and Applied Research in Engineering and Technology, vol. 4, pp. 1–10, 12 2015.
- [11] R. Ashutosh Aggarwal, Rajneesh Rani, "Handwritten devnagari character recognition," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, pp. 1–6, 5 2012.
- [12] M. N. D. B. S. Arora, D. Bhattacharjee and M. Kundu, "Complementary features combined in a mlp-based system to recognize handwritten devnagari character," Journal of Information Hiding and Multimedia Signal Processing, vol. 2, no. 1, pp. 1–7, 1 2011.
- [13] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, "Offline recognition of devanagari script: A survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 41, no. 6, pp. 782–796, Nov 2011.
- [14] A. Gupta, M. Srivastava, and C. Mahanta, "Offline handwritten character recognition using neural network," in 2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), Dec 2011, pp. 102–107.
- [15] S. Ramteke, A. Gurjar, and D. Deshmukh, "A streamlined ocr system for handwritten marathi text document classification and recognition using svm-acs algorithm," International Journal of Intelligent Engineering and Systems, vol. 11, pp. 186–195, 06 2018.