2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)
Pune, India. Oct 29-30, 2021

2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) | 978-1-6654-2503-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMARTGENCON51891.2021.9645786

Devanagari Handwritten Character Recognition using CNN as Feature Extractor

Shilpa Kaur Manocha
Dept. of Electronics and Communication Engineering
Delhi Technological University
Delhi, India
shilpamanocha03@gmail.com

Piyush Tewari
Dept. of Electronics and Communication Engineering
Delhi Technological University
Delhi, India
piyushtewari@dtu.ac.in

***Abstract*— This paper proposes the use of Convolutional Neural Networks as feature extractor for extraction of features from handwritten Devanagari characters. For classification, classifiers employed are SVM (Linear, Polynomial and RBF), KNN, RF, DT, MLP and XGB. Use of CNN model for feature extraction eliminates the need of handcrafted features by traditional pattern recognition methods. XGB Classifier, which is a very recent technique, has also been explored (which has not been done previously to the best of our knowledge). The experiments with these eight techniques have been done on the DHCD dataset proposed in year 2015. Use of CNN proved to be very effective for Devanagari characters recognition as all the models achieved recognition accuracy of over 92% and total training time including feature extraction and classification did not exceed a total of 12.16 minutes.**

Keywords*— *CNN, SVM, KNN, DT, RF, OCR, HCR, XGB, MLP

I. INTRODUCTION

Devanagari handwriting recognition has been gaining attention by researchers in recent years. The script serves as a base for many popular scripts like Hindi, Marathi, Sanskrit etc. Unfortunately, there are no robust commercial systems available for recognition of Devanagari handwritten characters. Owing to difficulties like the cursive nature of script, many characters differing only in minor structural details, different handwriting styles and non-availability of standard labelled datasets, the recognition task becomes tedious. Development of robust OCR[1] systems for recognition of handwritten characters of Devanagari script[2] will enable us to preserve old manuscripts by converting the physical files to digital formats.

For conversion of handwritten characters from physical documents and images to text, the first step is image pre-processing [3] that includes:- binarization, skew/slant/rotation removal, image thresholding and noise removal. The next step is feature extraction i.e. instead of feeding all raw pixels as input to the classifier for classification, several methods are employed to detect important features from the images. For textual data, relevant features are extracted using horizontal and vertical projections, crossings, distances, shape number, Histogram of gradient etc. Selection of features that can best describe the dataset and their extraction is a crucial step and affects the final recognition accuracy. Finally in the last step, a classifier takes the combination of these extracted features as inputs and classifies the characters into appropriate labels.

In this paper, a novel method for of the recognition of handwritten Devanagari characters has been presented. Feature extraction using Convolutional Neural Network model has been utilised that eliminates the need of explicit handcrafted feature selection and extraction. Manual selection of best features and extraction using various methods is a tedious task and doesn't always result in best accuracy. Main contributions of this paper include: -

1) Hybrid recognition method for automatic feature selection and classification of handwritten Devanagari characters. CNN model is used as feature extractor instead of traditional pattern recognition techniques and use of SVM, KNN, DT, RF, MLP, XGB as classifiers.
2) Characters classification with the use of XGB classifier: both with the input as raw pixels and with CNN as feature extractor.

The paper is arranged in following sections: - Section 1 gives Introduction, Section 2 is dedicated to literature review, Section 3 gives brief explanation of classification techniques, Section 4 describes about the proposed system architecture, Section 5 is dedicated to Results and Discussion and Section 6 provides conclusion of the experiments performed.

II. LITERATURE REVIEW

In [4], the authors made use of discrete cosine transform for feature extraction and Neural network based classification for recognition of Devanagari characters. The highest recognition accuracy obtained was 94.89%. In [5], the authors classified 1000 Hindi characters by calculating the histogram of projections and then classifying the characters with artificial neural network. The classification accuracy obtained was 90%. In [6], the authors have recognised Devanagari script characters by finding the gradient and curvature of characters and then classifying them using two classifiers namely MQDF and SVM. The highest recognition accuracy obtained was 95.13%. In [7], the authors employed GLAC feature extraction method and SVM classifier for recognition of Devanagari handwritten characters. Accuracies reported on the datasets: V2DMDCHAR and ISIDCHAR are 95.21% and 93.21% respectively. In [8], the authors classified Devanagari characters by combining wavelet based feature extraction and BPNN classifier. The highest accuracy achieved was 70%. In [9], the author proposed a deep convolutional NN for recognition of Devanagari handwritten alphabets and digits. The model obtained an accuracy of 98.47%. In [10], the authors developed a recognition system for Devanagari script by evaluating various structural and geometrical features and performing classification using MLP classifier.

Maximum classification accuracy achieved was 82.7%. In [11], the authors developed a recognition system for Devanagari script by evaluating various structural features and performing classification using FFBPN, CFBPN and EBPN classifiers. Accuracy rates obtained were 97.20%, 97.46% and 98.10% respectively. In [12], the authors use of AlexNet model that has been pertained on the ImageNet dataset, for building a recognition system for Devanagari script characters. The results demonstrated that in 3 epochs only, the accuracy rate went above 90%. The highest validation accuracy and test accuracy attained were 94.49 % and 95.46% respectively. In [13], the authors have proposed a deep CNN model for classification of Devanagari handwritten characters. The model was able to identify the characters with 93.73% accuracy on a small database of size 5484.

III. Related Theory

A. Support Vector Machine(SVM)

Support Vector Machine [14] is a supervised machine learning technique used for classification and regression tasks. SVM is a popular and widely used classifier for image classification tasks. It distinguishes the sample points by identifying a hyperplane that separates the classes. The sample points that are nearest to the hyperplane are defined as “support vectors”. The classifier aims to maximise the margin (difference between support vectors) on each side of hyperplane. The classifier takes 2 inputs: one array of shape (number_of_samples, number_of_features) and the other array containing class labels, either as numerical or string format, of shape (number_of_samples). A kernel function must be specified which can be linear, polynomial, radial basis function or any other custom function.

1) Linear Kernel: - {x, x’}
2) Polynomial (Poly): - {γ (x,x’) + r) ^ d , where γ- , d is degree, r- coefficient.
3) Radial Basis Function (RBF): (−γ‖x−x′‖2), where γ is gamma (> 0).

The classifier has following advantages :- powerful in high dimensional spaces, viable in cases where number of dimensions are more than the number of samples, is memory effective as it only focuses on a subset of training points i.e. the support vectors and is versatile i.e. different kernel functions can be used.

The disadvantages of the classifier are : overfitting may occur in cases where number of feature elements are larger than the number of sample points, selection of kernels is difficult and calculation of scores & probabilities is done using expensive k-fold cross-validation.

B. K-Nearest Neighbours (KNN)

KNN [15] is a supervised learning method. The idea behind this algorithm is that similar things occur in close proximity. Closeness between points is measured using distance between the points .e.g. by using Euclidean distance. It finds distance between the query and all data points , and selects a specified number of samples (k) closest data points from query. The query is classified to the class label that is the most repeating in the chosen k points group. The advantages of using KNN are:- easy and simple to implement, no need of tuning parameters, can be employed for classification as well as regression tasks. The disadvantage of algorithm lies in the slow classification time for large number of samples/classes. KNN algorithm is a popular choice in recommender systems, as they work on principle of close data points.

C. Decision Tree(DT)

Decision Tree [16] is a powerful classification and prediction tool that learns using an inductive approach. It has a tree structure similar to that of a flowchart. In a decision tree, there are several nodes ,that present a decision condition, starting from parent node at the top to the terminal node as the final class label. At each node, the data is classified based on a test on the attribute and each branch represents the result of the test/decision. A DT is built by recursively splitting the data samples into subsets based on the attribute tests at each node. The splitting is stopped when we have the final predicted classes at the final nodes. An optimal DT is one that is able to represent all data points with least number of node levels. Pruning methods are used to drop nodes that are not critical for classification ,hence reducing size of DT. Use of DT has following advantages:- they can handle both categorical and numerical values, no prior domain knowledge requirement, can be visualised , results can be interpreted and analysed to understand critical decision rules. DT have following limitations:- may lead to overfitting on data by creating complex tree, problem of variance i.e. change in tree with even slight changes in data points, no guarantee of optimal tree by use of greedy algorithms and it results in a biased tree if some classes in the dataset dominate.

D. Random Forest (RF)

Random Forest Classifier [17] is a supervised learning algorithm that combines decision tree classifier with the bagging method. A forest is a set of decision trees, that collectively vote for the correct label. The model builds multiple trees by randomly a subset of features for each tree. Thus multiple trees of reduced depth are built and overfitting is reduced. The results from subtrees are averaged to reach to final class label. The classifier also makes it possible to measure the importance of features on a scale of 0-1. Important features are those having higher importance value. The classifier doesn’t require much hyper-parameter tuning, is simple and easy to build, reduces chances of overfitting and mostly results in good accuracy for both classification as well as regression tasks. The training of model doesn’t need much time but prediction takes some time. In case of large number of trees, model gives higher accuracy but the computations become slow.

E. Multi Layer Perceptron (MLP)

Multi-layer perceptron classifier [18] is a deep, artificial neural network (ANN) .It has 3 types of layers- 1 input layer, multiple hidden layers & one output layer. The input layers are connected with output layers as directed graph. The classifier trains by method of back-propagation. It can be used as a classifier or a regressor. The model minimises cross entropy loss function for classification. For classification into multiple classes, ‘SoftMax’ function is used for adding non-linearity. The advantages of using MLP are:- it can understand non-linear models and can learn models in real time. The weaknesses include: requirement of fine tuning of hyper parameters like number of hidden units, number of layers and iterations, difference in validation accuracy with different random weight initialisations due to non-convex loss function of hidden layers. The model is also sensitive to feature scaling.

F. *Extreme Boost Gradient (XGB)*

XGB [19] Algorithm can be used for both classification and regression tasks. Boosting is an iterative approach to building ensemble of trees. In XGB classifier, gradient boosting is done by building trees iteratively. The trees try to reduce/correct residual errors from previous trees. In other ensemble methods, the models end up learning same mistakes. The building of large number of sequential trees creates a complicated model which may lead to overfilling. But the model provides parameter tuning methods to prevent it.

G. *Convolutional Neural Network (CNN)*

Convolutional NNs [20] are popularly used for image processing & computer vision tasks. They consist of combination of convolution and pooling layers followed by fully connected layers. The convolutional layers consist of filters and are convolved with input images to extract relevant features. The pooling layer reduces the dimensions of feature maps by performing average or maximum pooling over each area in the image. The fully connected layers activate the neurons and classifies the input features into appropriate classes.

IV. Proposed System Architecture

A. *Framework*

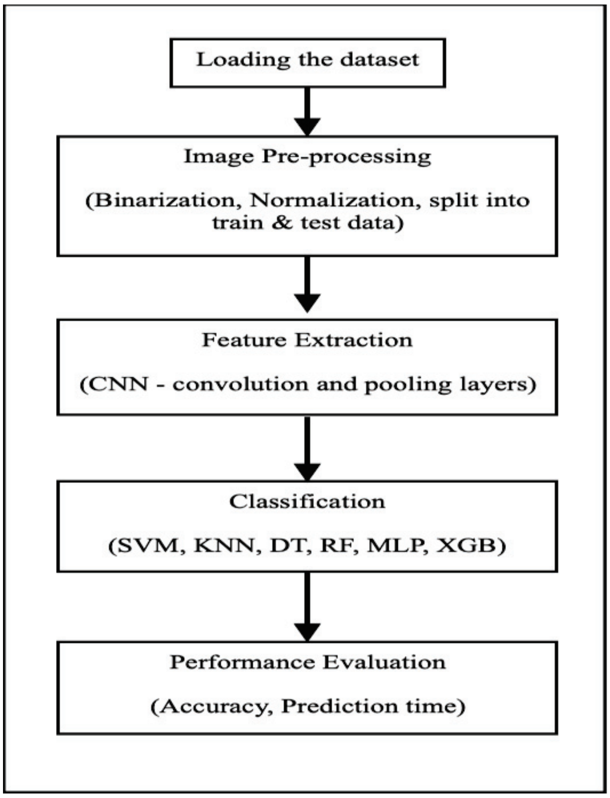

Fig. 1. Flowchart of proposed recognition method

Step1) Loading the dataset: The dataset consists of isolated characters of each character class in separate folders. The raw pixel information is available in a csv file. Each character is of size 32*32 i.e., having 1024-pixel values. The character labels have also been mentioned for all characters. Input features (1024) are read into one array and character labels in another array.

Step2) Image Pre-processing: The pixels contain information of the gray scaled images. The images are converted to binary. Normalisation is performed for all features by dividing them by 255(highest pixel value in black and white image). Thus, we get features normalised to a scale of 1. The features and character label arrays are split into training and testing set.

Step 3) Feature Extraction using CNN: CNN automatically extracts relevant features from the data fed. Convolution layers are used to create feature maps from training samples and pooling layers are used for reducing the size of feature maps generated. Instead, or connected the last fully connected dense layer for classification, the features are obtained in an array and fed to different classifiers as input. The model is trained for 3 epochs to learn the features from the training samples. Figure 2 shows the CNN model architecture used for the purpose of feature extraction.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 30, 30, 32)	320
conv2d_5 (Conv2D)	(None, 28, 28, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 14, 14, 64)	0
conv2d_6 (Conv2D)	(None, 12, 12, 64)	36928
conv2d_7 (Conv2D)	(None, 10, 10, 64)	36928
max_pooling2d_3 (MaxPooling2	(None, 5, 5, 64)	0
dropout_1 (Dropout)	(None, 5, 5, 64)	0
flatten_3 (Flatten)	(None, 1600)	0
dense_3 (Dense)	(None, 128)	204928
dense_4 (Dense)	(None, 64)	8256
dense_5 (Dense)	(None, 46)	2990

Fig. 2. CNN Model Architecture used for feature extraction

Step 4) Classification: Features extracted from last layer of CNN model are fed to various classifiers and test samples are applied for prediction. Classifiers are fed with two arrays, X - input features obtained from last/Dense layer of CNN and y-class labels for training and then tested on test samples. Total number of character classes are 46. Default parameter values have been used for each classifier.

Step 5) Performance Evaluation: For performance analysis , classification accuracy and prediction time are noted for all classifier models.

V. Results And Discussion

A. *Dataset Used*

For Experimental Analysis, we have made use of the DHCD [21] dataset, proposed by Acharya et. al, in 2015. The dataset includes 46 handwritten Devanagari characters - 36 alphabets and 10 numerals. Each character label has 2k image samples. i.e., a total of 92k samples. The dataset is divided into two parts: training set -to train and extract the features from CNN and testing set - fed to classifiers for testing the performance of classifier. We have considered 3 cases for dataset splitting. Case 1) 70:30, Case 2) 75:25, Case 3) 80:20.

Fig. 3. Handwritten characters from the DHCD dataset

B. Evaluation Parameters

1) Accuracy: It is the ratio of correct predictions made by the classifier to the total number of test samples fed. High accuracy percentage is the desired outcome.
2) Prediction Time: The time required by the classifier to segregate the test samples into different classes, after they have been fed with input features to be classified. Lowest prediction time for correctly identifying the classes is the desired outcome.

TABLE 3. Recognition Results of different classifiers on DHCD dataset.

Model / System	**Case 1: Dataset Split - 70:30**		**Case 2: Dataset Split - 75:25**		**Case 3: Dataset Split - 80:20**	
	Accuracy(in %)	**Prediction Time (in seconds)**	**Accuracy(in %)**	**Prediction Time (in seconds)**	**Accuracy(in %)**	**Prediction Time (in seconds)**
CNN + KNN	98.73	21.35	98.91	30.84	98.99	25.60
CNN + SVM Linear	98.41	**10.78**	**98.40**	**10.80**	98.87	12.03
CNN + SVM Poly	98.36	20.82	98.63	20.79	98.74	22.20
CNN + SVM RBF	**98.88**	23.78	99.01	23.21	99.13	26.60
CNN + MLP	98.55	22.84	98.69	17.73	98.86	30.03
CNN + DT	93.54	**2.95**	**93.85**	**2.79**	94.41	**3.43**
CNN + RF	98.57	20.21	98.84	20.20	98.7	23.88
CNN + XGB	98.15	169.92	98.37	160.59	98.42	198.51
Feature Extraction time taken by CNN (in seconds)	497.36		467.68		532.48	

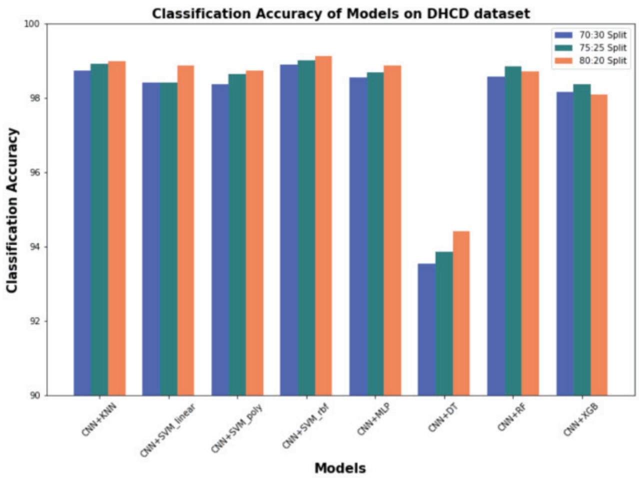

Fig. 4. Comparison of Classification accuracy of models on the DHCD [21] Dataset

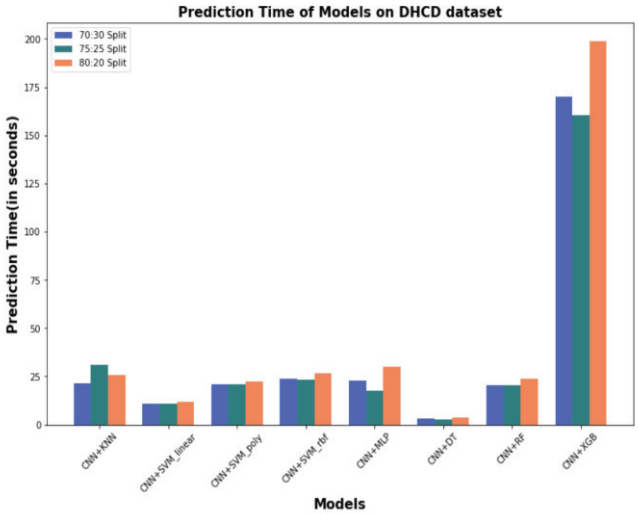

Fig. 5. Comparison of prediction time required by various models on the DHCD [21] Dataset

VI. CONCLUSION

It has been observed that CNN is able to identify relevant features from images and thus results in computational cost saving as compared to traditional pattern recognition techniques for feature extraction. It also saves the hassle and struggle of analysing and identifying best features for the image dataset and then applying various techniques or combination of feature extraction techniques to get appropriate feature maps. Use of CNN as feature extractor reduced the training and prediction time significantly for all classifiers(in the range of few seconds), along with achieving accuracy rates as high as 99% (CNN + SVM_RBF). The Devanagari characters have also been classified using XGB classifier which gives approx. 90% accuracy when trained with raw pixel values which is unusually remarkable. Use of CNN as feature extraction and XGB as classifier further improved the accuracy to 98% with average prediction time being only 176.34 seconds.

REFERENCES

[1] Mithe, R., Indalkar, S., & Divekar, N., “Optical character recognition”, International journal of recent technology and engineering (IJRTE), 2(1), 72-75,2013. OCR

[2] Bright, W.,. “The devanagari script.”, The world’s writing systems, pp.384-390,1996. Script

[3] Kumar, G., Bhatia, P.K. and Banger, I. “Analytical review of preprocessing techniques for offline handwritten character recognition.”, International Journal of Advances in Engineering Sciences, 3(3), pp.14-22, 2013.

[4] Kubatur, S., Sid-Ahmed, M., & Ahmadi, M. (2012). A neural network approach to online devanagari handwritten character recognition. In IEEE International Conference on High Performance Computing and Simulation (pp. 209–214). Madrid: IEEE Press.

[5] Yadav,D.,Sanchez-Cuadrado,S.,&Morato,J.(2013).Optical character recognition for Hindi language using a neural-network approach. Journal of Information Processing Systems (JIPS), 9(1), 117–140.

[6] Pal,U.,Chanda,S.,Wakabayashi,T.,&Kimura,F.(2008).Accuracy Improvement of Devnagari character recognition combining SVM and MQDF. In 11th International Conference on Frontiers in Handwriting Recognition (pp. 367–372). Canada: L. Lam Publications.

[7] Jangid,M.,&Srivastava,S.(2014).Gradient localauto-correlation for handwritten Devanagari character recognition. In IEEE International Conference on High Performance Computing and Applications (pp. 1–5). Bhubaneswar: IEEE Press.

[8] Dixit,A.,Navghane,A.,&Dandawate,Y.(2014).Handwritten devanagari character recognition using wavelet based feature extraction and classification scheme. In: Annual IEEE India Conference (pp. 1–4). Pune: IEEE Press.

[9] Acharya,S.,Pant,A.K.,&Gyawali,P.K.(2015).Deep learing based large scale handwritten devanagari character recognition. 9th International Conference on Software, Knowledge, Information Management and Applications (pp. 1–6). Nepal: IEEE Press.

[10] Dongre,V.J.,&Mankar,V.H(2015).Devanagari offline handwritten numeral and character recognition using multiple features and neural network classifier. In 2nd International Conference on Computing for Sustainable Global Development (pp. 425–431). New Delhi: IEEE Press.

[11] Shelke, S., & Apte, S. (2016). Performance optimization and comparative analysis of neural networks for handwritten Devanagari character recognition. In IEEE International Conference on Signal and Information Processing (pp. 1–5). Vishnupuri: IEEE Press. 106 S. Singh and N. K. Garg

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

[13] Narang, S.R., Kumar, M. & Jindal, M.K." DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition.” Multimed Tools Appl 80, 20671–20686, 2021.

[14] Ramya, S., & Shama, K. Comparison of SVM Kernel effect on online handwriting recognition: A case study with Kannada script. In Data engineering and intelligent computing (pp. 75-82), 2018. Springer, Singapore.

[15] Zhang, M. L., & Zhou, Z. H. ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition, 40(7), 2038-2048, 2007.

[16] Song, Y. Y., & Ying, L. U. Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130, 2015.

[17] Bernard, S., Adam, S., & Heutte, L. Using random forests for handwritten digit recognition. In Ninth international conference on document analysis and recognition (ICDAR 2007) (Vol. 2, pp. 1043-1047), 2007. IEEE.

[18] Satange, D. D., Ajmire, D. P., & Khandwani, F. I. Offline Handwritten Gujrati Numeral Recognition Using MLP Classifier. International Journal of Novel Research and Development, 3(8), 2018.

[19] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., & Cho, H. Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4), 1-4, 2015.

[20] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," International Conference on Engineering and Technology (ICET), pp. 1-6, 2017.

[21] S. Acharya, A. K. Pant and P. K. Gyawali, "Deep learning based large scale handwritten Devanagari character recognition," 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 1-6,2015.