



Offline handwritten Devanagari modified character recognition using convolutional neural network

MAMTA BISHT* and RICHA GUPTA

Department of Electronics and Communication Engineering, Jaypee Institute of Information Technology, Noida 201309, India
e-mail: bishtmamta29@gmail.com; richa.gupta@jiit.ac.in

MS received 25 January 2020; revised 3 November 2020; accepted 6 November 2020

Abstract. In this work, two convolutional neural network (CNN)-based models are proposed for offline handwritten modified character recognition (e.g. क्की) in Devanagari script formed when a Devanagari consonant (e.g. क) is followed by a Devanagari vowel (e.g. ई). The first model uses a single CNN architecture and the second model uses double-CNN architecture for the recognition of offline handwritten modified character. The double-CNN architecture performs better than single CNN architecture and uses a lesser number of output classes as compared with the actual existing classes of modified characters in Devanagari script. The recognition performance of these models is tested on Hindi consonants and Matras dataset with acceptable accuracy. The proposed CNN architecture yields better competitive results as compared with the traditional feature extraction (histogram of oriented gradients) and classification (support vector machine) techniques.

Keywords. Deep learning; convolutional neural network; Devanagari script; modified character recognition.

1. Introduction

Transcription of handwritten text into digital text has got increased attention among researchers due to its various challenges and complexities. The recognition system for printed text document exists with great performance [1]. However, there is a need to improve the performance of the handwritten recognition system. Handwritten text documents are divided into online and offline handwritten text document [2]. In online handwritten text documents, geometrical and temporal information is stored while writing (e.g. writing by a pressure sensitivity device on an electronic writing pad) [3, 4]. In an offline handwritten text document only a sample of the text is available after the text has been written with variations in the handwriting style of writers, which makes the offline handwriting recognition system more challenging than the online handwriting recognition system [5]. Moreover, the complex shapes of characters in some scripts like Devanagari, Bangla, etc. make the performance of the offline handwriting recognition system more difficult. Since handwriting recognition system has potential applications in the field of offline handwritten historical document digitization, bank cheque processing, postal automation, automatic data entry, etc., there is a necessity to improve the handwriting recognition system more accurately.

Indic scripts have some more challenges in handwriting recognition than Latin, Chinese, Korean and Japanese because of the presence of variations in the order of strokes or symbols, half consonant, etc., which is discussed in detail in terms of online recognition in [6]. Kaur *et al* [7] also presented a detailed review of work done in multilingual online and offline character recognition for Indic and non-Indic scripts. This work identified the deficiencies and presented an in-depth view of work done at each phase of character recognition for the printed and handwritten documents. Kumar *et al* [8] discussed major challenges for character and numeral recognition in Indic and non-Indic scripts.

This research article investigates Devanagari's modified character recognition with the help of CNN. It is observed that the performance of recognition mostly depends on the feature extractor methods. Good accuracies of feature identification and extraction like pixel value, shape, orientation, texture, position, etc. are required to solve the recognition problem accordingly. On the other hand, the recent deep learning era reduces the requirement of developing new feature extractors for every problem; e.g. convolutional neural network (CNN) learns low-level features such as edges and lines in early layers, then loops and then a high-level representation of a text image. Among all the methods presented in the literature for the recognition of characters, numerals, words, etc. it is found that the evolution of deep learning methods makes improvement in

*For correspondence
Published online: 03 February 2021

traditional feature-based methods significantly [9–12]. In this article, a deep double-stage CNN network has been used for offline handwritten modified character recognition. The superiority of proposed methods is claimed with the help of comparison with traditional feature extraction and classification method.

2. Proposed approach

In this research work, two CNN-based models and one traditional feature extraction (Histogram of Oriented Gradients—HOG) and classification (Support Vector Machine—SVM)-based methods are presented for the recognition of offline handwritten modified characters. The description of these models is presented in sections 2.1 and 2.2.

2.1 Proposed CNN architecture description

The proposed CNN network architecture is shown in figure 1, which consists of three components including pre-processing, convolutional layers and classification layer from left to right.

Preprocessing consists of resizing of images and conversion of samples from colour to greyscale. The convolution layers automatically extract features from each input image and the last convolution layer forwards the output to a fully connected layer, which is followed by the classification layer.

The architecture of this model consists of CNN layers. Input image size is set to $300 \times 300 \times 1$. For 7 CNN layers, filters (8, 16, 32, 64, 128, 256 and 256) are used of size 3×3 . These filters determine the number of feature maps. Padding is kept ‘the same’ to ensure that the spatial output size is the same as the input size. It helps in keeping information at the borders. Stride is set to 1. Batch Normalization (BN) layers used between convolution layers and non-linearity as a Rectified Linear Unit (RELU) normalize the activations and gradients propagating through the network so that network training becomes an easier optimization problem. The max-pooling layer uses pool size of [2,2] with stride equal to 2 to return the maximum value of a rectangular region of inputs. After convolution

and max-pooling (down-sampling) layers, fully connected layers are added in which the neurons connect to the preceding layer’s neurons. The size of the fully connected layer is set to the total number of unique classes (or labels) in the target data. In single CNN architecture, the fully connected layer is set to 435. In double-CNN architecture, the fully connected layer is set to 37 in Stage-1 and 13 in Stage-2. The description of these CNN-based models is presented in sections 2.1.a and 2.1.b. Softmax normalizes the output of a fully connected layer and its output produces positive numbers that sum to one. The classification layer uses the probabilities returned by the softmax activation function for each input to assign the input image to one of the mutually exclusive classes and compute the loss. Stochastic Gradient Descent with Momentum (SGDM) with an initial learning rate of 0.01 is specified in training options. Maximum epochs are set to 12 and data is shuffled in every epoch. An epoch is a complete training cycle on a training dataset where network accuracy during training is specified by validation data and validation frequency. During training, validation accuracy is calculated at regular intervals.

2.1a Single CNN architecture: This model comprises a 7-layer CNN architecture that is trained on offline handwritten modified characters with labels as the name of corresponding consonant and modifier (or matra) together, formed by rearranging the Hindi consonants with Matras dataset, e.g. sample image of ‘का’ (क(Ka) + आ(AA)) is labelled as ‘KaAA(का)’. Since the dataset formed has very few samples of a few labels, repetition of sample images has been done to increase its size. It is defined as a complete labelled dataset in table 1 and divided into unique groups of train, validation and test in 6 experiments. Validation and test accuracies are calculated 6 times during each experiment and mean validation accuracy (6-fold cross-validation accuracy) is calculated to evaluate its performance and shown in table 2.

2.1b Double-CNN architecture: In this proposed model, two CNN architectures are used. The first CNN architecture is trained on consonant labelled dataset available as Hindi consonants with Matras in CALAM in order to classify test samples into basic consonant classes, here named Stage-1. For example, Devanagari modified character ‘का’ (क(Ka) + आ(AA)) is labelled as ‘Ka(क)’. The

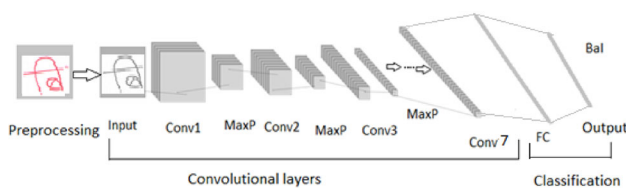


Figure 1. The network architecture.

Table 1. Description of data used for experiment evaluation.

Dataset description	Samples			Total
	Train	Validation	Test	
Complete labelled	6069	1291	1260	8620
Consonant labelled	3428	736	734	4898
Modifier labelled	2718	583	583	3884

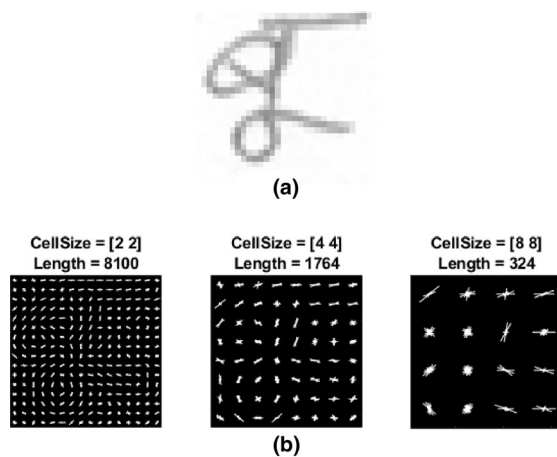
Table 2. Six-fold cross-validation and test accuracies for single CNN architecture and double-CNN architecture.

Model	Accuracy (%)	
	Validation	Test
Single CNN architecture	81.52	81.62
Double-CNN architecture Stage-1	89.80	90.12
Double-CNN architecture Stage-2	85.65	85.39

second CNN architecture is trained on modifier labelled dataset formed by rearranging sample images of consonant labelled dataset so that it can learn to classify test samples into correct modifier classes, here named Stage-2. For example, Devanagari modified character 'का' (क(Ka) + आ (AA)) is labelled as 'AA(आ)'. This model is able to predict consonant and modifier class together for a test sample after combining the results obtained from Stage-1 and Stage-2. The acquired complete predicted label is compared to the actual label to calculate accuracy on the test dataset. To check the performance of both stages, 6-fold cross-validation accuracy has been evaluated and tabulated in table 2.

2.2 HOG features and SVM classifier

The HOG is a feature descriptor technique used to count the gradient orientation in a localized portion of an image called cells. The shape information in the feature vector is varied by varying the cell size. The visualization of cell size [2 2], [4 4] and [8 8] is shown in figure 2 for [32 32] size image. It states that the maximum shape information is achieved in cell size [2 2] and cell size [8 8] encodes the least shape information. However, the dimensionality of the feature vector is increased in cell size [2 2] to cell size [8 8] from 324 to 8100. A good negotiation of cell size [4 4] is chosen here, which is able to encode a sufficient amount of

**Figure 2.** (a) Sample of modified character. (b) Visualization of HOG features on variable cell size.

spatial information in feature-length 1764. SVM is chosen as a classifier using supervised learning on HOG features and its corresponding labels.

3. Experimental results and analysis

The dataset used for experiments is randomly divided into the train (70%), validation (15%) and test samples (15%). The description of dataset distribution is tabulated in table 1. The dataset is divided into 6 number of sections (or folds), where each section is decided for train, validation and test at some point. The validation and test results of these 6 sections are calculated and the mean value is presented in table 2. It is noted for single CNN architecture that the average 6-fold cross-validation accuracy and test accuracy are, respectively, 81.52% (with 4.81 standard deviation) and 81.62% (with 5.17 standard deviation). It is also observed from table 2 that the average 6-fold cross-validation accuracy for Stage-1 and Stage-2 in double-CNN architecture is approximately 89.80% (with 3.33 standard deviation) and 85.65% (with 5.87 standard deviation), respectively. The performance of Stage-1 and Stage-2 is also calculated for random distribution of dataset used in each stage for 11 experiments and its average value is shown in table 3. It is observed from table 3 that both stages of double-CNN architecture perform well on the recognition of test data up to an average value of 90.99% (with a standard deviation of 0.01). A few examples of modified character recognition results using double-CNN architecture on test dataset are presented in table 4.

4. Comparison of results

In this research work, an attempt has been made for the recognition of offline handwritten modified characters in the Devanagari script using two CNN-based models as described in sections 2.1.a and 2.1.b. The way of recognition for a sample image of 'का' by both models is presented in figure 3. From tables 2 and 3, it is observed that the validation and test accuracies get improved for double-stage CNN model.

The recognition work discussed in this article is also evaluated by feature extraction and classification method as discussed in section 2.2. The dataset used in this work is

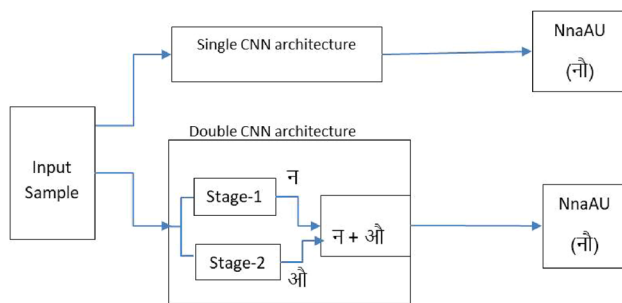
Table 3. Performance of double-CNN architecture using random distribution of dataset.

Stage-1		Stage-2		Combined on test
Validation	Test	Validation	Test	
<i>Accuracy (%)</i>				
95.92	95.56	95.74	95.97	90.99

Table 4. Recognition results of double-CNN architecture for a few test samples.

Sample image	Actual label	Predicted by double CNN architecture	Decision
बा	बा	बा	✓
बि	बि	छि	✗
चो	चो	चो	✓
ढा	ढा	ढा	✓
ण	ण	णा	✗

Indicating correct by tick (✓) and incorrect by cross (✗)

**Figure 3.** Recognition example by single CNN architecture and double-CNN architecture.**Table 5.** Comparison of proposed models with (HOG+SVM) technique.

Dataset test	Accuracy (%)	
	CNN	HOG+SVM
Complete labelled	81.62	55.62
Consonant labelled	95.56	56.59
Modifier labelled	95.97	78.81

partitioned into train and test data (7:3 ratio). The performance of the HOG features and SVM classifier is also evaluated. In this experiment the HOG features are extracted from 4×4 cell size of 32×32 size image, which leads to a 1×1764 feature set for each image as shown in figure 2. The comparison of recognition performance on the dataset described in table 1 is presented in table 5 by the HOG+SVM technique with the CNN technique.

5. Conclusion

The paper presents a novel technique for recognition of offline handwritten modified characters in the Devanagari script. Two methods using CNN models have been

discussed and it is observed that double-CNN architectures perform better than single CNN architectures. Traditional feature extraction like HOG features and a classifier like SVM are also implemented to check the performance and it is observed that deep CNN is able to recognize Devanagari modified character with more acceptable accuracy.

Acknowledgements

The authors are thankful to Malaviya National Institute of Technology, Jaipur, for providing the CALAM dataset.

References

- [1] Breuel T M 2008 The OCRopus open source OCR system. In: *Document Recognition and Retrieval XV*, p. 68150F
- [2] Plamondon R and Srihari S N 2000 Online and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 22: 63–84
- [3] Keysers D, Deselaers T, Rowley H A, Wang L L and Carbune V 2016 Multi-language online handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 39: 1180–1194
- [4] Choudhury H and Prasanna S M 2019 Representation of online handwriting using multi-component sinusoidal model. *Pattern Recognit.* 91: 200–215
- [5] Plötz T and Fink G A 2009 Markov models for offline handwriting recognition: a survey. *Int. J. Doc. Anal. Recognit.* 12: 269–298
- [6] Bharath A and Madhvanath S 2009 Online handwriting recognition for Indic scripts. In: *Guide to OCR for Indic scripts*, pp. 209–234
- [7] Kaur S, Bawa S and Kumar R 2020 A survey of mono-and multi-lingual character recognition using deep and shallow architectures: indic and non-indic scripts. *Artif. Intell. Rev.* 53: 1–60
- [8] Kumar M, Jindal M K, Sharma R K and Jindal S R 2019 Character and numeral recognition for non-Indic and Indic scripts: a survey. *Artif. Intell. Rev.* 52: 1–27
- [9] Sanchez J A, Romero V, Toselli A H and Vidal E 2016 ICFHR2016 competition on handwritten text recognition on the READ dataset. In: *Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition*, pp. 630–635
- [10] Chakraborty B, Shaw B, Aich J, Bhattacharya U and Parui S K 2018 Does deeper network lead to better accuracy: a case study on handwritten Devanagari characters. In: *Proceedings of the 13th IAPR International Workshop on Document Analysis Systems*, pp. 411–416
- [11] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning*. Cambridge: MIT Press
- [12] Bisht M and Gupta R 2020 Multiclass recognition of offline handwritten Devanagari characters using CNN. *Int. J. Math. Eng. Manage. Sci.* 5: 1429–1439