Proceedings of 4th International Conference on Computers and Management (ICCM) 2018

# A Deep Learning Approach to Recognize Handwritten Telugu Character Using Convolution Neural Networks

*Dr. Anupama Angadi [a], Dr. Valli Kumari Vatsavayi [b], Dr. Satya Keerthi Gorripati [c]*

[a]*IT Department, GMR Institute of Technology, Rajam, Andhra Pradesh, India, anupama.a@gmrit.org*
[b]*Department of CSSE, Andhra University, Visakhapatnam, Andhra Pradesh, India, vallikumari@gmail.com*
[c]*IT Department, GMR Institute of Technology, Rajam, Andhra Pradesh, India, satyakeerthi.g@gmrit.org*

Abstract:
Automated character recognition is one of the most vital components which enable a data processor to distinguish letters and digits possibly using contextual data. Various attempts at resolving this problem using different selections of classifiers and features have been established and still the problem remains challenging. In the proposed work, we suggest and evaluate a classic Convolutional Neural Network (CNN) for the identification of online Telugu characters. The network comprises of 4 layers, one with 5 X 5 and remaining with 3 X 3 kernels and ReLU, softmax activation functions, followed by max pooling and two dense layers. The final layer has 168 outputs, matching to the classes considered: vowels, consonants. To train and estimate the CNN, we pre-owned 45,133 images written by many telugu writers. A qualified analysis proved the efficiency of the proposed CNN against previous methods in an interesting dataset. On test dataset, the classification technique provided 92.4% accuracy. The conclusion is improved than some recently proposed literature used for the identification of online handwritten Telugu characters.
Keywords: Character Recognition, Convolution Neural Network, Hand Writing, Telugu Characters

## 1. Introduction

Neural Networks are vastly used in the field of Pattern Recognition. Handwriting differs from person to person; hence, it is a tedious task to recognize the handwriting characters. In pattern recognition field, Handwriting Recognition (HR) has become a recent research area of interest due to the exponential usage of the resources such as paper documents, photographs, smartphone, iPad, and so on (Cireşan, Meier, U., Gambardella, & Schmidhuber, 2010; Das,Redyy et.al,2012; Das, Sarkar et.al, 2012). HR can be categorised into online or offline. Offline is acknowledged after completion of writing. Offline HR converts image from input into binary image, where pixel values are only either 0 or 1. Whereas, Online HR captures digital pen-tip movements in a dynamic format. This information contains list of coordinates, direction and speed, used as response for the classification system.

Earlier works used shallow learning with hand-designed features on both online and offline datasets. (Pujari et.al,2004; Cireşan& Meier, 2015; Lakshmi,Jain,Patvardhan,2006) offered a complete OCR (Optical Character Recognition) system, which is font, shape independent and using proper selection of Wavelet scaling function the signatures are calculated. (Rani & Vasudev, 2016) Applied multi-layer perceptron networks for identification of Telugu characters. During training MLP (Multi-Layer Perceptron) back propagation method so that recognition can be done efficiently and accurately. ( Koppula, & Negi,2011) Projected a new frill map method, in which every binary pixel value of an image is connected with a frill number that labels the distance to the adjacent black pixel. These frill numbers are used to fragment text lines. (Rajkumar et.al,2012) Presented two schemes for online recognition linking multi-classifier frame works. (Lakshmi et.al, 2006 ) Used histograms of edges for knowing features of basic symbols. Where a symbol is a basic unit of recognition in Telugu script. (Prasad & Kanduri,2016) Projected a multiple zone based feature extraction which is an arrangement of two methods. The first method is a Genetic algorithm and a second method uses distance and density based features.

An enormous literature has been reported for HR in English and Asian languages (Chaudhuri, 2006) such as Chinese Japanese, etc., and very few efforts on Indian languages (Pal& Chaudhuri,2004) like Sanskrit, Tamil, Telugu and Kannada (Sen, et.al,2018; Chaudhuri,2006). Researchers have recently introduced CNN based approaches for the offline recognition of English characters (Maitra et.al,2015; Yuan et.al,2015; Bouchain,2007; Bai, 2014). Encouraged by this fact, in the present framework, a HR algorithm for Telugu with high accuracy and with minimum training, classification time is proposed.

### *1.1 Data Exploration*

This dataset is available in website HP Labs India [dataset]. The dataset comprises of 270 trials of each of 138 Telugu “characters” written by many Telugu writers to get variability in writing styles. Telugu script has 18 vowels and 36 consonants, of which 13 vowels and 35 consonants are in common usage and made available in TIFF files shown in Fig. 1.

Telugu handwriting style is in non-cursive and therefore pen-up typically divides the basic graphic symbols although not always. Hence, the graphic symbols i.e., vowels, consonants, consonant modifiers and diacritical signs are included in the symbol set. Some consonant-vowels are also included which dissembling be easily subdivide. Additionally, the symbol set also comprises certain symbols which do not have a dialectal interpretation but have an unchanging outline across writers and help lessen the total number of symbols to be collected. So totally 166 symbols exist which are assigned to Unicode characters. Some characters of vowels and consonants in Telugu are represented in Fig. 1.

**Fig. 1(a). Typical "Telugu characters" Images Vowels**

**Fig. 1 (b) Consonants**

## 2. Proposed Work

In the proposed work, we have used Convolutional Neural Network (CNN), a deep learning construction for recognition of hand written Telugu Characters (TCR) which holds an input, convolutional, rectified linear unit, pooling layer and fully connected layer continued by an output layer as shown in Fig. 2.

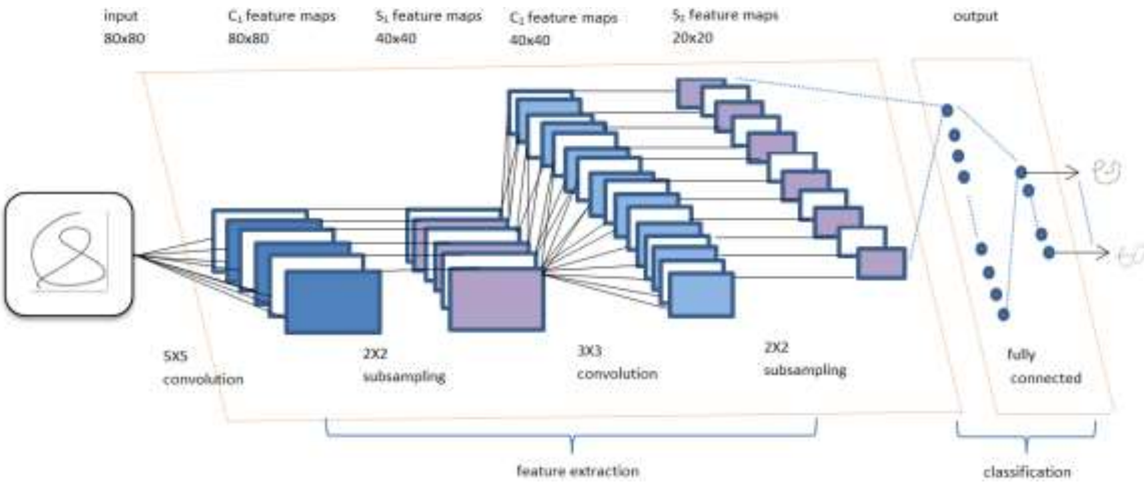


**Fig. 2 Visual features in CNN**

The early step required to initiate the process is to select a hand written character image for the classification. The input layer will hold the raw pixel values of the selected image of width 80, height 80, which passes the input image to convolution layer. The responsibility of this layer is to involve random number of filters to proceed along the height and width of the image to yield a feature map. A filter is a sequence of numbers where the numbers are called weights or parameters. A sample learned weights of the 1st layer of the proposed model is shown in Fig. 3(b).

**Fig. 3 (a) Sample handwritten Telugu characters**

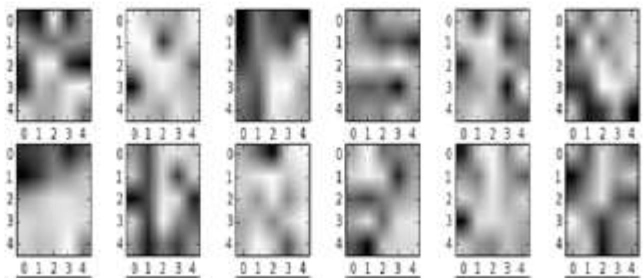

**Fig. 3 (b) Learned weights of 1st layer**

By sliding each filter across the height and width of an image and compute the dot products between the input volume and the filter during the forward pass, we have achieved an 80 X 80 sequence of numbers, which is termed as a feature map. The output of the first convolutional layer creates 32 such feature maps and transforms it to the next layer through a differentiable function. Lastly the output is of 3D volume (80 X 80 X 32), is transformed to first pooling layer where the image is down-sampled along the spatial dimensions, resulting in an output volume of (40 X 40 X 32). Formally, it can be mathematically expressed in Eq. 1

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} k_{ij}^l + b_j^l) \tag{1}$$

Where $x_j^l$ the output of the existing layer, $x_i^{l-1}$ is the previous layers' output, $k_{ij}^l$ is kernel for existing layer and $b_j^l$ is the bias. Where $i$ selects input map from $M_j$. Though, the input feature maps will be convolved with random kernels to create the respective output maps.

Proceeding in the similar fashion, second convolutional procedure creates 32 different feature maps. A size of 2 X 2 filters results a feature map size of 40 X 40 into 20 X 20. Further down-sampling in the pooling layers produces resizing feature maps of size 5 X 5. This subsample layer performed on the input feature maps. Due to this down-sampling, the size of the output feature maps will be decreased based on the size of the mask. In this approach, a 2 X 2 mask is used. This can be conveyed using the following Eq.2

$$x_j^l = f(\beta_j^l down(x_j^{l-1}) + b_j^l) \tag{2}$$

Where $down(.)$ signifies a max-pool function through local averaging, multiplicative coefficient and bias respectively. The above function adds up all n X n blocks of the feature maps from preceding layers and selects either highest or average values.

The final feature map from the last convention layer is changed into a single dimensional feature vector matrix is taken as 3200 (=128 X 5 X 5) random nodes which are functionally connected to 138 output class labels. CNN, minimizes the error using the following Eq.3

$$E = \frac{1}{2}\frac{1}{PO}\sum_{p=1}^{P}\sum_{o=1}^{O}(d_o(p) - y_o(p))^2 \tag{3}$$

Where $P, O$ are patterns (i.e., 256 in the proposed work), output nodes (i.e., 138) and $d_o(p), y_o(p)$ are predicted output and exact output of node respectively.

It is noted that the proposed work used zero- padding in order to retain the image size from the previous layer. Each convolutional layer uses this hyper parameter around the border of an image to control the spatial size. In the proposed work, two activations like RELU, Softmax have been employed for the convolution and pooling layers during organization in the output layers. The Softmax activation is used for multiple class logistic regression whereas the RELU function has output zero if the input is less than 0, and 1 otherwise. The mathematical notations for both functions are mentioned in the following Eq.4 and 5

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{4}$$

$$f(x) = \max(x, 0) \tag{5}$$

Both functions use log loss function for error correction and described as shown in Eq.6

$$E(p,o) = -\sum_j p_j \log o_j \tag{6}$$

Where $p, o$ are predicted, output of neuron $j$ and $o_j$ represents the result of the activation function respectively. The addition is carried on every neuron on the output layer.

## 3. Results and Discussion

TCR model was experimented using Amazon EC2 server instance for use with t2.xlarge instance type on Bitfusion Ubuntu 14 TensorFlow setup.

The proposed work considered 166 different character classes, where each consists of 270 samples. After resizing the images, the dataset splits into 34325 images to use for training, 5617 images for test and 5191 character images for validation. We have analysed the learnable filters of CNN such as input image shapes, pooling strategies and optimizer functions on the model. Table 1 recapitulates the results for all likely groupings of those distinctions on the test set. From this table, we have detected that recognition accuracy has reached to maximum when the model contains four convolutional layers, with zero-padding and max-pooling by two fully connected layers.

In addition, we observed the impact of filter size by (7 X 7, 5 X 5 and 3 X 3) if filter size becomes higher the model fails to observe tiny details of structurally related character patterns. However, if the size of the filter is too small it may produce duplicate information which, in turn, would decrease the model accuracy. Hence, filter size of 5 X 5 is chosen for the first convolutional layer and 3 X 3 for the remaining is confirmed to be optimal filter size. Our model also observed the accuracy with the influence of image size by (32 X 32, 64 X 64 and 80 X 80) and with different optimizers (adam and sgd). Finally our model has reached the 92.4 accuracy with sgd optimizer.

**Table 1: Model accuracy by CNN with varying structure, image size and optimizers for classification**

| Exp | Model | Structure | Image size | Training Epochs | Loss | Optimizer | Accuracy |
|---|---|---|---|---|---|---|---|
| 1st | CNN | 3 Conv layers, with zero-padding and max-pooling by 2 fully connected layers | 32 X 32 | 20 | Categorical crosssentropy | sgd | 70.8 |
| 2nd | | 3 Conv layers, with zero-padding and max-pooling by 2 fully connected layers | 32 X 32 | 20 | Categorical crosssentropy | adam | 73.5 |
| 3rd | | 3 Conv layers, with zero-padding and max-pooling by 2 fully connected layers | 64 X 64 | 20 | Categorical crosssentropy | sgd | 82.3 |
| 4th | | 4 Conv layers, with zero-padding and max-pooling by 2 fully connected layers | 64 X 64 | 20 | Categorical crosssentropy | adam | 82.9 |
| 5th | | 4 Conv layers, with zero-padding and max-pooling by 2 fully connected layers | 80X80 | 20 | Categorical crosssentropy | sgd | 92.4 |

Additionally, we tried out the model with generalization techniques like Dropout and Data augmentation functions. Data augmentation consists of applying transformations to our training set, in order to increase the dataset size and variation. Even though we reached a high performance in model accuracy without data augmentation. By increasing the size of our dataset, data augmentation helps prevent over fitting. However, it is not fool-proof, since the augmented images will be highly-correlated. Finally, for parameter updates, we settled the model optimizer to sgd with learning rate 0.1.

The accompanying Fig. 4 shows images corresponding to actual and predicted of the first fifty out of which four are misclassified images which show the model is predicting almost correct labels. The validation curves for the model accuracy, loss for the train set and test set are shown in Fig.5

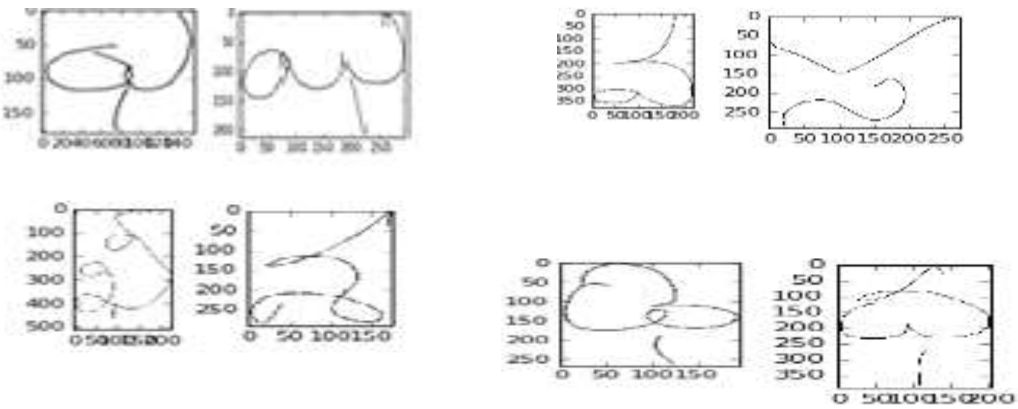

**Fig. 4 Misclassified images recognized by the model**

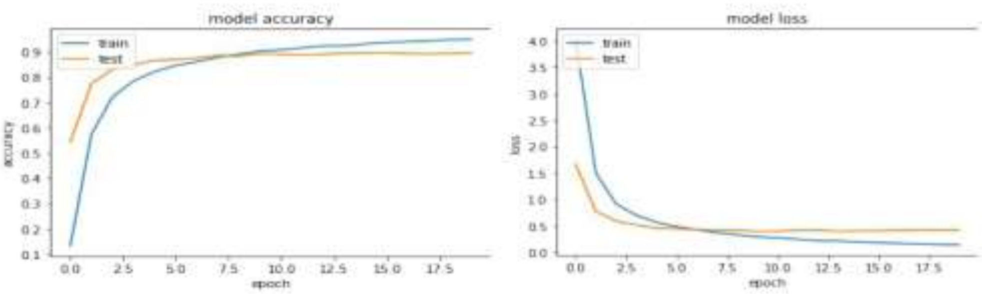


**Fig. 5 Model accuracy for the training and test sets**

## 4. Conclusion

In this proposed work, we projected a deep convolutional neural network character image into 166 classes contained in 270 samples. A new architecture was proposed that captures the low-level textual features of the hand written characters. The network comprises of 4 conventional layers, the first layer with 5 X 5 and remaining with 3 X 3 kernels and ReLU activation functions, followed by two Dense layers with Softmax activation function.

The training was carried out by decreasing the categorical cross-entropy with the SGD optimizer. The outcome of the proposed method is quite impressive when compared against the state of the art algorithms. The method can be further extended to multiple classes (i.e., vattu, gunintham) and can improve recognition accuracy even on a bigger dataset.

## REFERENCES

Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. Neural computation, 22(12), 3207-3220.

Das, N., Reddy, J. M., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., & Basu, D. K. (2012). A statistical–topological feature combination for recognition of handwritten numerals. Applied Soft Computing, 12(8), 2486-2495.

Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., & Basu, D. K. (2012). A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. Applied Soft Computing, 12(5), 1592-1606.

Pujari, A. K., Naidu, C. D., Rao, M. S., & Jinaga, B. C. (2004). An intelligent character recognizer for Telugu scripts using multiresolution analysis and associative memory. Image and Vision Computing, 22(14), 1221-1227.

Cireşan, D., & Meier, U. (2015, July). Multi-column deep neural networks for offline handwritten Chinese character classification. In Neural Networks (IJCNN), 2015 International Joint Conference on (pp. 1-6). IEEE.

Lakshmi, C. V., Jain, R., & Patvardhan, C. (2006). OCR of printed Telugu text with high recognition accuracies. In Computer Vision, Graphics and Image Processing (pp. 786-795). Springer, Berlin, Heidelberg.

Rani, N. S., & Vasudev, T. (2016). A Comparative Study on Efficiency of Classification Techniques with Zone Level Gabor Features towards Handwritten Telugu Character Recognition. *International Journal of Computer Applications*, *148*(1).

Koppula, V. K., & Negi, A. (2011, September). Fringe map based text line segmentation of printed Telugu document images. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 1294-1298). IEEE.

Rajkumar, J., Mariraja, K., Kanakapriya, K., Nishanthini, S., & Chakravarthy, V. S. (2012, September). Two schemas for online character recognition of telugu script based on support vector machines. In Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on (pp. 565-570). IEEE.

Lakshmi, C. V., Jain, R., & Patvardhan, C. (2006). OCR of printed Telugu text with high recognition accuracies. In Computer Vision, Graphics and Image Processing (pp. 786-795). Springer, Berlin, Heidelberg.

Prasad, S. D., & Kanduri, Y. (2016, September). Telugu handwritten character recognition using adaptive and static zoning methods. In Technology Symposium (TechSym), 2016 IEEE Students' (pp. 299-304). IEEE.

Chaudhuri, B. B. (2006, October). A complete handwritten numeral database of Bangla–a major Indic script. In Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft.

Pal, U., & Chaudhuri, B. B. (2004). Indian script character recognition: a survey. pattern Recognition, 37(9), 1887-1899.

Sen, S., Shaoo, D., Paul, S., Sarkar, R., & Roy, K. (2018). Online Handwritten Bangla Character Recognition Using CNN: A Deep Learning Approach. In Intelligent Engineering Informatics (pp. 413-420). Springer, Singapore.

Chaudhuri, B. B. (2006, October). A complete handwritten numeral database of Bangla–a major Indic script. In Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft.

Maitra, D. S., Bhattacharya, U., & Parui, S. K. (2015, August). CNN based common approach to handwritten character recognition of multiple scripts. In Document Analysis and Recognition (ICDAR), 2015 13th International Conference on(pp. 1021-1025). IEEE.

Yuan, A., Bai, G., Jiao, L., & Liu, Y. (2012, March). Offline handwritten English character recognition based on convolutional neural network. In Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on (pp. 125-129). IEEE.

Bouchain, D, "Character recognition using convolutional neural networks". Inst.Neural Inf. Process. (2007).

Bai, J., Chen, Z., Feng, B., & Xu, B. (2014, October). Image character recognition using deep convolutional neural network learned from different languages. In Image Processing (ICIP), 2014 IEEE International Conference on (pp. 2560-2564). IEEE.

Dataset http://lipitk.sourceforge.net/datasets/teluguchardata.html