Proceedings of the International Conference on Intelligent Computing and Communication for Smart World (I2C2SW-2018)
14-15 December 2018

Isolated Offline Tamil Handwritten Character Recognition Using Deep Convolutional Neural Network

A. Arun Prakash
Department of ECE
Kongu Engineering College
Perundurai, India
arunprakash.ece @kongu.ac.in

S. Preethi
Department of ECE
Kongu Engineering College
Perundurai, India
Preethi.s@kongu.ac.in

***Abstract*—In this paper, a Convolutional Neural Network architecture (ConvNet) for offline isolated Tamil character recognition is proposed. A first-ever attempt has been made to recognize all 247 characters in the Tamil text using 124 unique symbols. The proposed architecture contains two Convolutional layers and Two Fully Connected (FC) Layers with ReLu activation function. Softmax function is used in the final layer to compute the probability of the classes. The 9.6 million parameters of the network are randomly initialized using He initialization and fine-tuned using Nesterov Accelerated batch gradient descent optimization algorithm. Dropout regularization method has been used to avoid over-fitting of the network to the training data. A total of 98,992 image samples from IWFHR database are divided into 69% for training set (68,488), 20% for validation set (20,584) and 11% for test set (9920). Cross entropy loss has been used during the training phase to measure the loss and thereby update the parameters of the network. The network has achieved 88.2% training accuracy and 71.1% testing accuracy. The reason for reduction in the test accuracy is analysed. The source code and the dataset have been published for a quicker reproducibility of the result.**

***Keywords*—ConvNets, Tamil Character Recognition**

I. INTRODUCTION

Tamil is one of the classical languages spoken by about 72 million people in and around the world [1]. The Tamil alphabets consist of 12 vowels, 18 consonants and one special character (ஃ) which is neither a vowel nor a consonant. The combination of vowels and consonants produces 216 compound alphabets. Therefore, a total of 247 characters are obtained by summing up vowels, consonants, compound characters and the special character as shown in the Table I.

A simple observation reveals that a unique set of 124 symbols are sufficient to represent all 246 characters in the Tamil scripting. For example, consider the set of characters in a second column and a third column in Table I; all of them are structurally similar except that the characters in the third column differ from the characters in the second column only by a spatially separated adjacent symbol ா . Therefore, it is sufficient to build a classifier to recognize these 124 symbols which could represent all 247 characters. The class numbers are from 0 to 123 associated with each class symbols for the classification task. The special

TABLE I. TAMIL ALPHABETS (VOWELS: TOP MOST ROW, CONSONANTS: LEFT MOST COLUMN)

	அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ
க்	க	கா	கி	கீ	கு	கூ	கெ	கே	கை	கொ	கோ	கௌ
ங்	ங	ஙா	ஙி	ஙீ	ஙு	ஙூ	ஙெ	ஙே	ஙை	ஙொ	ஙோ	ஙௌ
ச்	ச	சா	சி	சீ	சு	சூ	செ	சே	சை	சொ	சோ	சௌ
ஞ்	ஞ	ஞா	ஞி	ஞீ	ஞு	ஞூ	ஞெ	ஞே	ஞை	ஞொ	ஞோ	ஞௌ
ட்	ட	டா	டி	டீ	டு	டூ	டெ	டே	டை	டொ	டோ	டௌ
ண்	ண	ணா	ணி	ணீ	ணு	ணூ	ணெ	ணே	ணை	ணொ	ணோ	ணௌ
த்	த	தா	தி	தீ	து	தூ	தெ	தே	தை	தொ	தோ	தௌ
ந்	ந	நா	நி	நீ	நு	நூ	நெ	நே	நை	நொ	நோ	நௌ
ப்	ப	பா	பி	பீ	பு	பூ	பெ	பே	பை	பொ	போ	பௌ
ம்	ம	மா	மி	மீ	மு	மூ	மெ	மே	மை	மொ	மோ	மௌ
ய்	ய	யா	யி	யீ	யு	யூ	யெ	யே	யை	யொ	யோ	யௌ
ர்	ர	ரா	ரி	ரீ	ரு	ர	ரெ	ரே	ரை	ரொ	ரோ	ரௌ
ல்	ல	லா	லி	லீ	லு	லூ	லெ	லே	லை	லொ	லோ	லௌ
வ்	வ	வா	வி	வீ	வு	வூ	வெ	வே	வை	வொ	வோ	வௌ
ழ்	ழ	ழா	ழி	ழீ	ழு	ழூ	ழெ	ழே	ழை	ழொ	ழோ	ழௌ
ள்	ள	ளா	ளி	ளீ	ளு	ளூ	ளெ	ளே	ளை	ளொ	ளோ	ளௌ
ற்	ற	றா	றி	றீ	று	றூ	றெ	றே	றை	றொ	றோ	றௌ
ன்	ன	னா	னி	னீ	னு	னூ	னெ	னே	னை	னொ	னோ	னௌ

character (ஃ) is also considered as class for classification.

The classification can be carried out either in online mode or in offline mode. Offline Tamil handwritten character recognition (OTHR) is the process of converting a handwritten text in a scanned document into its respective Unicode Transformation Format (UTF-8). Offline recognition is considered more difficult than that of the online mode because of the higher dimensionality of the input and the addition of noise during the scanning process. However, OTHR has potential applications in reducing memory requirement for storage of scanned version of handwritten scripts and could also be used as an input to Text to Speech synthesiser system for Tamil language.

II. RELATED WORK

There has been a minimal but continuing effort in making a robust isolated offline handwritten recognition system for Tamil characters. Irrespective of that it is still been a challenge to reduce the rate of classification error as the number of classes considered for classification task are more, the existence of highly similar characters, and great variation in writing style.

Many authors in the past haveappliedtheir own dataset in their research work and reported the results. Hence the result produced by them is not reproducible as their dataset was not made publically available. Most of the techniques proposed in the past relied much on the traditional approach in which certain features are extracted from the input image based on the domain expertise; the extracted features are given to classification algorithms such as Support Vector Machines (SVM) for recognition. The quick summary of such approaches can be found here [2].

It is well known that the CNN performs well on image classification tasks [3] in spite of its network and computational complexity. Therefore, it is reasonable to apply it for OHR tasks. One such attempt was made in [4].They have used Convolutional Neural Network (Convnet) for offline OTHR task and have achieved recognition accuracy of 94%. However, the authors have considered a dataset of size 18000 samples for a subset of 35 symbols out of 124 symbols. The model they have trained was directly evaluated on the test set and then fine-tuned the parameters to achieve the higher test accuracy. This might have over fitted the model for both training and testing data.

III. THE DATASET DESCRIPTION

In order to set up a benchmark for the offline OTHR, the dataset from International Workshop on Frontier in Handwriting Recognition (IWFHR) have been used for the performance evaluation of the proposed system. The samples from the dataset were augmented in order to develop a more robust system. The details of the modified dataset are as follows:

TABLE II. DETAILS OF THE DATASET

Total samples per class	Total number of classes	Total number of samples
798	124	98,952

A total of 98,952 samples were randomly divided into training data, validation data and test data. Out of 98952, 68448 (69% of the total) were considered for training, 20584 (20% of the total) were considered for validation and the rest 9920 (10% of the total) were considered for testing the model. The image samples in the dataset have variable-resolutions and hence all the images are resized to a fixed dimensionality before feeding it to the network on both training and testing.

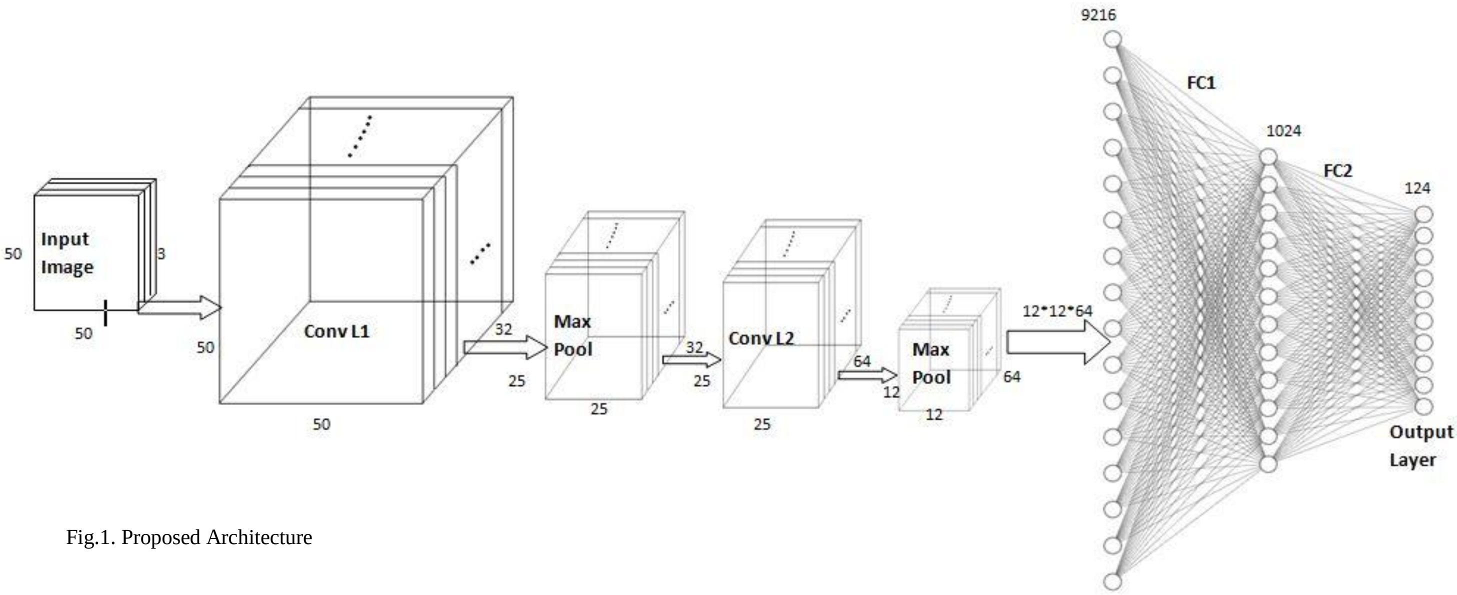


Fig.1. Proposed Architecture

IV. THE PROPOSED CNN ARCHITECTURE

The proposed CNN architecture is summarized in the Fig.1. The network contains two convolution layer (Conv) followed by Max pooling layer and two fully connected (FC) layer followed by an output layer. The dimensionality of each layers are displayed along side of the respective layers.

A.Convolution Layer and Max Pool Layer

If f(x,y) and g(x,y) denotes input and the kernel function respectively at some local region, then the convolution between the two can be computed using eq.(1)

$$y(x, y) = \sum_{i=1}^{n} \sum_{j=1}^{n} f(i, j) g(i, j) \qquad (1)$$

The above equation assumes the dimension of both the input and the kernel are n x n x 1.The convolution layer takes an input image of size 50 x 50. The first convolution layer uses 32 kernels of size 5 x 5 with a Padding value set in such a way that the output spatial dimension is same as input spatial dimension. Therefore, convL1 produces the output blob of size 50 x 50 x 32. This output dimension is reduced further to 25 x 25 x 32 by skipping the redundant information after passing it through a Max pooling layer 1 with the kernel size of 2 x 2 and stride 2. The second convolution layer takes the output from the max pooling layer1 and increases its depth to 64 by using 64 kernels with kernel size of 5x5x32 and padding value set appropriately to preserve the spatial dimension. This helps in finding more abstract features that are important for classification. Therefore, the output dimension of the ConvL2 is 25 x 25 x 64 and it is reduced further again to 12 x 12 x 64 by passing it through Max Pooling Layer2.

B. Rectified Linear Unit (Relu)

The choice of activation function plays a crucial role in terms of training a network to converge faster [5]. Activation units such as sigmoid and tanh could be saturated and may lead to vanishing gradient problems. Considering the importance of above said statements, Relu activation function as in eq.(2) proposed in [4] was used throughout the proposed architecture.

$$\mathrm{Re}Lu(x, y) = \max(0, y(x, y)) \qquad (2)$$

C. Fully Connected Layer

The output from the last convolution layer is flattened out to a single vector of dimension 1 x 9216. The first FC Layer takes the 1 x 9216 vector as input and produces the output of dimension 1 x 1024 .The second FC layer further reduces the dimension to 1 x 124 by computing more abstract representation of the input passed to this layer. The final layer is a Softmax Layer which computes the output probability for all the classes.

V. RESULT AND DISCUSSION

A. Experimental Setup

The proposed architecture was developed using python deep learning framework: Tensorflow 1.9. The model was trained and tested on Nvidia's Tesla K80 GPU. All image samples for training and testing the model were taken from IWFHR database. The total available training samples were divided into Mini-batches of size 256 for training the network. The dataset contains images of varying dimension; hence they were resized to 50 x 50 before feeding it into the network during the training phase and the testing phase.

The network parameters are initialized using He Initialization and the cross entropy loss was considered to measure misclassification error of the network. The Nesterov Accelerated batch gradient descent optimization algorithm was used to fine tune the network parameters with the learning rate of 0.001. In order to prevent the network from overfitting to the training data, dropout regularization was used with 0.4 dropout probability.

B.Experimental Result and analysis

After training the network for thirty thousand iterations with the above parameter setting, the network achieved 88.2% accuracy on the training data and 71.1 % accuracy on the testing data. The trend of the loss vs iteration is shown in the Fig.2. and the image representation of the confusion matrix is shown in Fig.3. The bright white pixels in the diagonal elements of the confusion matrix represent that most of the classes were correctly classified with a higher accuracy and gray pixels in the diagonal elements indicates that a few classes were misclassified. An analysis reveals that the misclassifications arise due to highly similar nature of a few classes.

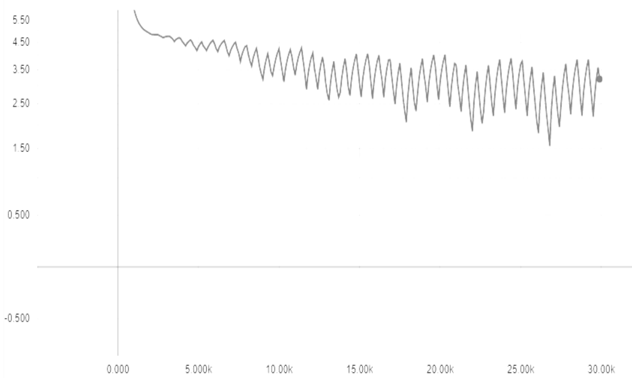


Fig.2. Trend of Iteration Vs Loss

TABLE III. LISTS OF TOP 3 MISCLASSIFIED SYMBOLS

S.No	True Class	Misclassified as	Max
1	ரி	ஈ,த,ஞி,தி,நி,பி,மி,லி ,றி,ய்,ல்,ற்	ஈ(10)
2	டூ	ஞீ,ரீ,டு,மு,மூ,ளு	டு(10)
3	ஙீ	கி,ஞி,டி,நி,பி,யி,லி, றி,ழீ	பி(15)

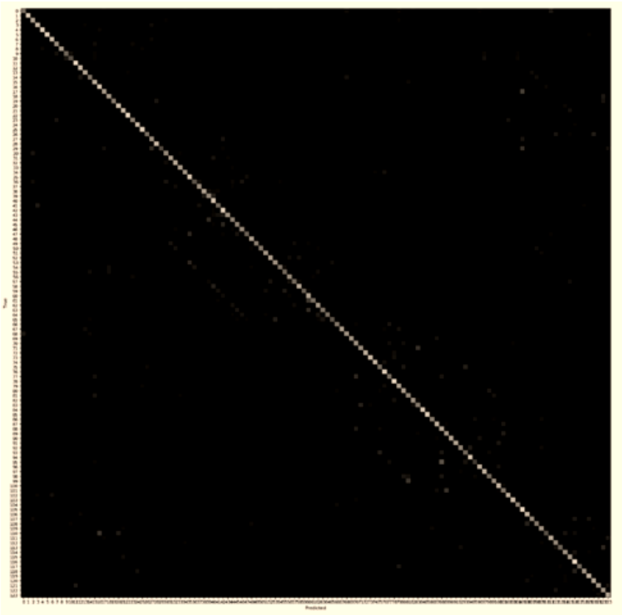
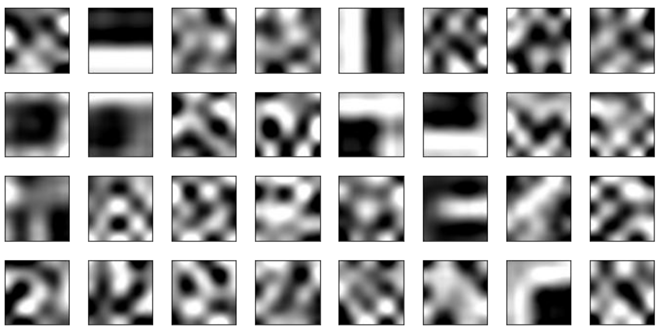
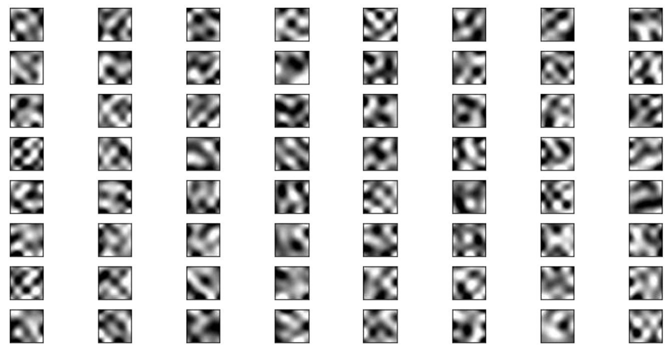


Fig.3.Image representation of Confusion Matrix

The Table III lists a few classes that were highly misclassified by the network.From the Table III it can be inferred that the misclassification occurs among the symbols having high similarity in their structure. For example, the symbol wasmost often (10 times) misclassified as .This is because the, kernels in the convolutional layers learned to look for complex shapes such as lines and curvesas shown in the Fig.4, in the given symbol. Since the two symbols are highly correlated, the same set of neuron fires for both symbols and led to misclassification.



(a)



(b)

Fig.4. Visualization of Kernel Functions inConvolutional Layer 1(a) and Layer 2 (b)

VI. CONCLUSION

The proposed deep Convolutional Neural Networkarchitecture performed significantly well on Tamil Hand written character recognition task. Totally 124 classes were considered for classification. The network has achieved 88.2% training accuracy and 71.1% testing accuracy. It was observed that the highly similar symbols were misclassified often and hence led to reduction in testing accuracy.

REFERENCES

[1] https://en.wikipedia.org/wiki/Tamil_population_by_nation

[2] Umapada, et al.,**"H**andwriting recognition in Indian regional scripts: asurvey of Offline techniques" ACM Trans. Asian Lang. Inform. Process. 11, 1, Article 1 (March 2012)

[3] Alex Krizhevsky , Ilya Sutskever , Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks", Proceedings of the 25th International Conference on Neural Information Processing Systems, p.1097-1105, December 03-06, 2012, Lake Tahoe, Nevada .

[4] Misha Sra and PrashanthVijayaraghavan, "Handwritten Tamil Recognition using a Convolutional Neural Network", MIT Media Lab, 2014.