Handwritten Tamil character recognition with deep neural network

Pradeepsurya Rajendran
E-mail: pradeepsurya@live.com

Abstract—Tamil handwritten character recognition (offline) is a quite challenging problem, due to the complex curvy nature of the Tamil characters and variants in the writing style of the different users. Recently, deep convolutional neural network (CNN) has exhibited a great potential in image classification problems by learning discriminative features automatically. In this work, classification of 156 different Tamil characters using a single CNN, trained end-to-end from isolated Tamil character images, has been demonstrated. A CNN architecture that achieved the state-of-the-art performance in Chinese handwritten character recognition was used in this work and trained end-to-end using isolated character images of IWFHR-10 dataset. Upon optimizing different hyper-parameters and incorporating data augmentation technique, the CNN model achieved a training accuracy of 97.996% and a testing accuracy of 94.897%. This result is the state-of-the-art accuracy in recognizing all 156 character classes in IWFHR-10 dataset. Equipped with this deep CNN, Tamil OCR system can be used to effectively digitize rare Tamil literary works and thereby preserving the Tamil culture.

Keywords: Tamil Handwritten Character Recognition, Optical Character Recognition (OCR), Convolutional Neural Network, Pattern Recognition, Tamil

I. INTRODUCTION

Handwriting recognition is the capability of the computing system to recognize and interpret handwritten inputs from different sources like documents and photographs. Some of the general use cases of this technology are (i) digitizing documents for preservation, which includes literary works and historical documents/newspapers (ii) data entry automation (iii) assisting blind and visually impaired persons. It is one of the challenging problems in pattern recognition, due to a considerable variation in individual writing style. This domain has an active research community working on different areas of handwriting recognition, such as online recognition and offline recognition. Online recognition involves the conversion of digital pen-tip movement into a list of coordinates that can be used as the input for the recognizing system. On the other hand, offline recognition uses images as input and extracts features. In this paper, offline recognition has been demonstrated, as it is more challenging than the online recognition. Efficient Handwritten Character Recognition (HCR) systems are widely available for languages such as English, Chinese, Arabic, and Japanese [1- 3]. However, for Indian languages, especially for Tamil, the research contributions are minimal, despite the high demand for an efficient Tamil HCR system.

A. *Tamil*

Tamil, ‘Semmozhi’, is one of the oldest languages in the world with several million speakers in the state Tamilnadu, India and also in several other countries such as Srilanka, Malaysia, and Singapore. It is a classical language with vast and rich literature. The literary works exist not only in handwritten books but also in palm leaves and stone inscriptions. Palm leaf manuscript and stone inscriptions contain valuable information on several subjects such as medicine, philosophy, astrology, and Tamil culture. It is imperative that these rare literary works have to be preserved by digitization. For that, Tamil HCR system should be robust and achieve performance on par with humans.

Tamil script has 12 vowels, 18 consonants, and one aytham. Vowels in combination with consonants form 216 composite consonants, and thus, the language contains a total of 247 characters. Since the consonant is represented as the separate character while forming a composite component, it can be segmented into separate symbols. Therefore, the total number of characters to recognize will be 156. The writing style in Tamil script is from left to right, and the concept of the upper/lower case does not exist. Comparing to Latin character recognition, offline Tamil Handwritten Character Recognition (THCR) is a harder problem because of the highly complex characters with curvy nature, the variants in the writing style of different user and potential confusion due to the similarity between handwritten characters. Some of the research works on offline THCR [4, 5] involved only a few character classes, which is due to the difficulty in differentiating all 156 characters.

B. *Dataset*

Isolated Tamil character images are required, to implement offline Tamil HCR. HP Labs, India created the isolated handwritten Tamil character dataset [6], both online and offline, for the Tamil Handwritten Character Recognition competition organized in the context of the 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR-10). The offline dataset contains approximately 500 samples for each of the 156 characters in

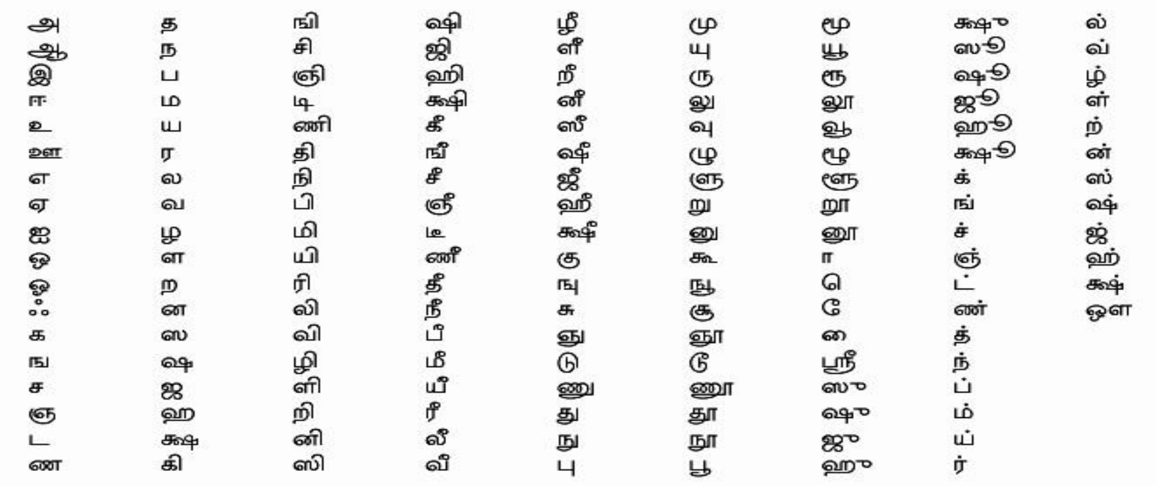


Fig. 1. 156 character classes in IWFHR-10 dataset

Tamil. These characters were written by the native Tamil writers, including adults, university graduates, and school children. All 156 character classes in the dataset are shown in figure 1.

C. Related works

Bhattacharya et al. [7] proposed a two-stage recognition scheme using HP Labs dataset for offline Tamil character recognition system. In the first stage, a small number of groups of character classes were created using unsupervised k-means clustering. Then, a supervised classification technique – Multi-Layer Perceptron – was used to classify characters in each of the smaller groups. Using all samples of 156 character classes, an accuracy of 92.77% and 89.66% were obtained on the training and testing dataset respectively. Vijayaragavan et al. [4] used a convolutional neural network to recognize 35 character classes of HP Labs dataset for offline Tamil character recognition. Different network architectures were tried, and also different generalization techniques such as stochastic pooling, probabilistic weighting, local response normalization, and local contrast normalization were experimented with the network. As a result, local response normalization outperformed all other configuration with a training accuracy of 99% and a testing accuracy of 94.4%. Deepa et al. [8] used Modified GA, which combined Levenshtein distance and Genetic algorithm for classification, with line strokes as features. Using all 156 character classes from HP Labs dataset, the proposed model obtained a recognition accuracy of 89.5%. Unlike machine learning approaches for handwritten character recognition, Deepa et al. [9] used nearest interest point classifier to recognize all 156 character classes in HP Labs dataset. The classifier achieved an accuracy of 90.2%, which is the state of the art result in recognition of all 156 characters classes.

One of the significant reasons for minimal research contributions in THCR, comparing to other languages, is the scarcity of standard benchmark dataset.

D. Algorithm and Technique

In recent years, Deep learning algorithms have shown to exceed human-level performance in visual tasks such as playing Atari games [10], skin cancer classification [11], and object recognition [12]. Convolutional Neural Network (CNN), a class of deep neural network, has shown excellent performance in image recognition tasks [2, 4, 11] because it has the potential to work well with data that has a spatial relationship. Also, it is good at developing an internal representation of an image, thus allowing the model to learn the position and scale invariant structures in the data. Therefore, CNN would be an appropriate choice to recognize sophisticated Tamil characters images that belong to 156 classes.

For high performance, CNN requires a large amount of labeled training data, which may not be available all the time. Moreover, collecting more data and labeling will be an expensive effort. Also, the dataset used for this project contains only 500 samples, approximately, for each class. Therefore, in order to handle such a situation, the CNN model trained and developed for different problem domain can be applied to the current recognition problem [13]. This process is known as transfer learning.

Transfer learning [14] is a machine learning technique, where a model trained on one task will be used as a starting point on a second related/unrelated task. This technique has shown great success in the visual task such as skin cancer classification [11], in which GoogleNet Inception v3 CNN architecture that was pre-trained on approximately 1.28 million images (1,000 object categories) from the 2014 ImageNet Large Scale Visual Recognition Challenge was used. In this work, two different CNN architectures were experimented with transfer learning technique. For implementation, an open source Keras library with tensorflow backend was used.

E. Evaluation metrics

In the earlier works [4, 7, 8, 9], accuracy, which is the ratio of the number of correct predictions to the total number of predictions, was used as an evaluation metric to determine the model performance. In addition to accuracy, in this work, precision[1], recall[2] , and F1-score (harmonic mean of precision and recall) are computed for every class comparing to all other classes. Since THCR is a multiclass classification problem and also the dataset used in training, validation and testing are

[1] True Positive / (True Positive + False Positive)

[2] True Positive / (True Positive + False Negative)

balanced, these evaluation metrics would be more appropriate and provide better insight on the model's performance.

II. MATERIALS AND METHODS

A. Data Preparation

The IWFHR-10 dataset contains images of Tamil characters written by 214 people. All these images are in either '.png' or '.tiff' format with unequal height and width. The total number of samples for each character are not equal, and thus the dataset is imbalanced. However, specific tactics have been used to combat the imbalanced dataset. The number of samples for each class has been shown in figure 2, in which x-axis represents the class and y-axis represents the number of samples in that class. From figure 2, it can be witnessed that the samples count for some of the classes is only around 275, which is almost half the count of many other classes.

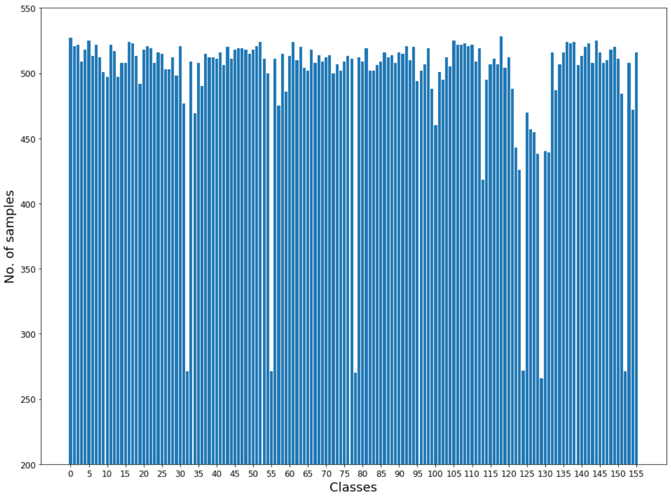


Fig. 2. Samples count for each class

Since the dataset contains images of the characters written by 214 people, there are 214 folders in the dataset with each folder containing the images of all 156 characters written by the respective writer. Thus, there are a total of approximately 82,500 randomly sized image samples. Image samples of Tamil vowels written by one of the users have been shown in figure 3.

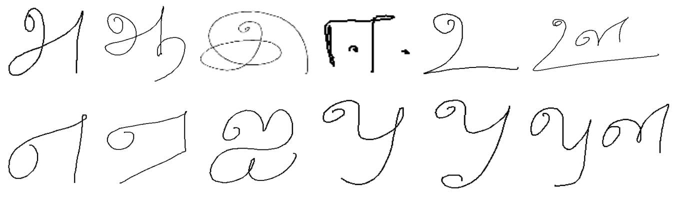


Fig. 3. Tamil vowels, IWFHR-10 dataset

As the first step in data preparation, the image samples are split into training, validation, and testing datasets in a ratio (60 %: 20 %: 20%). Then, all these image samples are converted to '.jpg' format and stored in structured directories, which are organized in such a way that it is compatible to use with Keras' flow_from_directory (). Initially, the dimensions of the images are not modified because it can be handled using the 'target_size' parameter in flow_from_directory (). All the above mentioned pre-processing steps are carried out in python script.

Secondly, some of the images in the dataset are incorrectly labeled. In order to ensure whether the labels for all images were proper, every image was scrutinized, and the incorrect one was then labeled correctly. Like shown in figure 2, the dataset is imbalanced, which could create a problem in the prediction due to the bias towards majority classes. However, this problem can be obviated by adopting a simple tactic, which is 'Resampling the dataset' [15]. Resampling consists of two methods, under-sampling, and over-sampling. Both of these methods have been applied in the dataset to make it balanced.

Under-sampling involves removal of samples from the majority classes. For this, classes that contain samples higher than 500 (majority class) are under-sampled by manually deleting some images. In general, the drawback of this technique is the loss of potentially useful information that would be significant for the model's training. However, in this dataset, the total number of image deletion per majority class amounts to less than 5%, which will not affect the model's performance.

On the contrary, over-sampling involves the addition of images in the minority classes. One potential solution for over-sampling the image dataset is data augmentation. For this, using Keras' ImageDataGenerator, different augmentation techniques such as zooming, rotation, and translation are implemented on minority classes. Eventually, all classes are balanced, with each containing 500 image samples.

B. Architecture

Designing a CNN architecture from scratch might not be needed for 90% or more of applications, as adopting a pre-trained model that works best on ImageNet would most likely work well on some other dataset with some hyper-parameter tweaks [16]. This process is known as Transfer learning. By using transfer learning technique, the time and the money on computing resources to learn many features can be saved. In this work, two CNN architectures were trained and tested on IWFHR-10 dataset. First, the Xception architecture that was trained on ImageNet dataset was used. Then, a CNN architecture that achieved the best performance in Handwritten Chinese Character Recognition (HCCR) competition was employed. It resulted in better performance than Xception in classifying 156 characters of the IWFHR-10 dataset.

1) Xception

Xception [17], stands for an extreme version of Inception, was developed by the author of Keras library. This architecture is made up of a linear stack of depthwise separable convolution layers with residual connections.

Initially, in this work, the Xception model with ImageNet weights was trained on the dataset. For this, the final layer of the model was removed and then a layer of average pooling and a fully connected layer with 156 nodes and softmax activation function were included. The input images were resized to 299 x 299 x 3 and are pre-processed by dividing each pixel value by

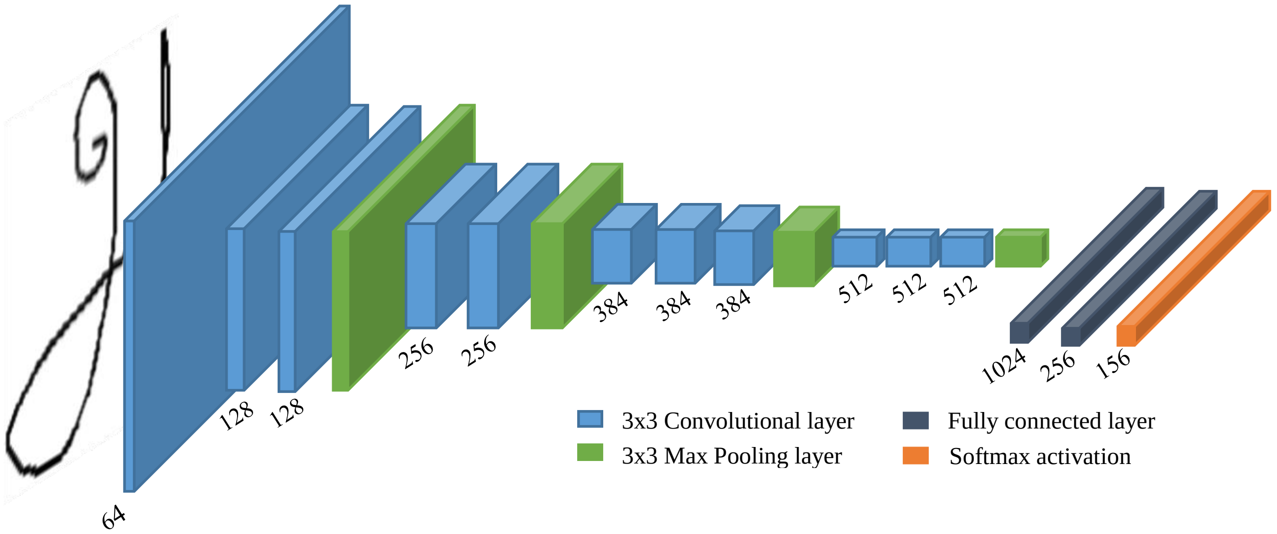


Fig. 4. The figure displays the architecture of the proposed model. All convolutional layers are depicted in blue, all max pooling layers in green and the fully connected layers in black. The final orange box depicts the softmax activation layer. The number of filters for each convolutional layer is shown underneath as are the number of neurons for the fully connected layers. The number of neurons in the last layer is equal to the total number of characters.

255. With Adam [18] optimizer and a batch size of 128, the model was trained for 10 epochs. Unfortunately, the results are not as expected and are very poor. The model achieved a testing accuracy of less than 1%. Since the IWFHR-10 dataset is not related to ImageNet dataset, the weights learned on ImageNet dataset performs poorly in the current dataset and failed to differentiate the features in the characters.

Next, without using the pre-trained ImageNet weights, the Xception architecture was trained end-to-end on the IWFHR-10 dataset with random initialization of weights. For this as well, the final layer of the model was removed, and then a layer of average pooling, a dropout layer with 50% probability and a fully connected layer with 156 nodes and softmax activation function were included. The input images were resized to 96 x 96 x 3. With Adam optimizer and a batch size of 128, the model was trained for 10 epochs. As a result, the model achieved a testing accuracy of 77.78%, which is quite good but still worse than the state of the art accuracy of 90.2%. The hyper-parameters, as mentioned above, such as image size, optimizer, batch size, activation function, and the probability of a dropout layer were chosen after experimenting with different values.

2) *Proposed model*

The CNN architecture [19] that achieved the state of the art accuracy of 97.68% in HCCR was trained on the current dataset. Since Chinese characters are as complex as Tamil characters, a model that performs well on Chinese characters dataset is highly likely to perform well on the Tamil characters dataset, and so this CNN model was chosen to experiment with the current dataset. It was inspired by VGG model and performed better than the ResNet and DenseNet in Chinese HCR.

The architecture has been shown in figure 4. It contains a total of 11 convolutional layers, each with a kernel size of 3x3. All convolutional layers are followed by leakyReLU [20] activation, batch normalization [21] and preceded by zero padding, which is an addition of rows and columns of zeros at the top, bottom, left and right side of an image. Adding a layer of batch normalization can address the problem of vanishing gradient by standardizing the output of the previous layer. Also, it speeds up the training by reducing the number of required iterations and thus, it facilitates the training of deeper neural networks. Third, fifth, eighth, and eleventh convolutional layers are followed by the max-pooling layer with a pool size of 3 x 3, a stride of 2 and the same padding. A stack of convolutional layers is followed by three fully connected layers. The third one is the soft-max layer that performs classification and thus it contains 156 channels, one for each class. Each hidden layer is followed by leakyReLU activation and batch normalization. The input to this CNN is a fixed-size 96 x 96 x 3 image. The only pre-processing performed was dividing each pixel of an image by 255.

Initially, with Adam optimizer and a batch size of 128, the model was trained for 10 epochs. As a result, the testing accuracy of 91% was obtained, which is far better than 77.78% of Xception. Also, this value is higher than the state-of-the result. Moreover, this testing accuracy can be improved even further by optimizing several hyper-parameters and incorporating data augmentation technique.

CNN requires a large amount of training data for high performance. However, the number of training samples for each class is limited to only 500. Also, the model should be robust to make predictions that are invariant to variations of the same class of patterns [22]. However, with this limited data, the model may over-fit and affect the classification robustness. The easiest and simple solution to this problem is the data augmentation, in which the training samples are artificially augmented using label preserving transformations. Data augmentation is extensively used in image recognition tasks and has improved recognition accuracy significantly [22].

In order to improve the accuracy of the model further, different hyper-parameter combinations and different data augmentation techniques were tried. The model refinement strategies that were tried tested and the impacts on the model's performance have been shown in table 1. All the implementations, mentioned in this section were carried out on

Tesla K80 GPU with 12 GB memory, in Floydhub (A cloud service provider for deep learning applications).

Table 1. Model refinement strategies

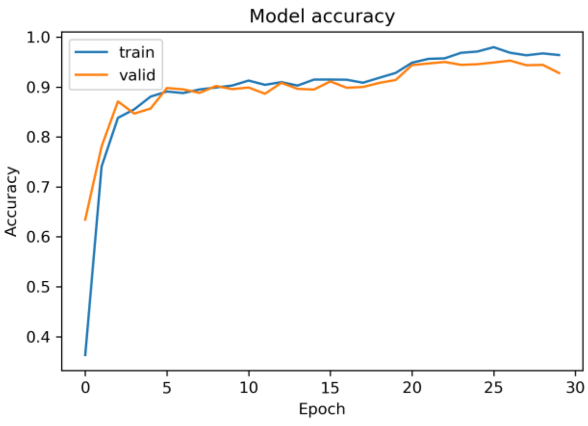
Refinement Strategies	Impact on the model's performance
The number of epochs increased from 10 to 30.	Testing accuracy improved from 91% to 94.154%.
Different batch sizes like 32, 64, 100, and 128 were tried.	All batch sizes other than 128 reduced the testing accuracy from 94.154%.
Different image sizes such as 96 x 96, 128 x 128, and 224 x 224 were tested.	96 x 96 resulted in better accuracy and also lesser computation time.
Various optimizers such as Adam, RMSprop, and Stochastic Gradient Descent (SGD) were tried and tested.	Out of all these, Adam performed better.
For data augmentation, using Keras' ImageDataGenerator, the following transformations were tried • Rotation • Width shift • Height shift • Shear • Zoom • ZCA whitening	(i) Width shift, height shift, and shear did not improve or diminish the model's performance. (ii) Zoom declined the model's performance. (iii) Whereas, ZCA whitening along with rotation (value = 2) exhibited a marginal improvement in the testing accuracy from 94.154% to 94.897%.

III. Results and Discussion

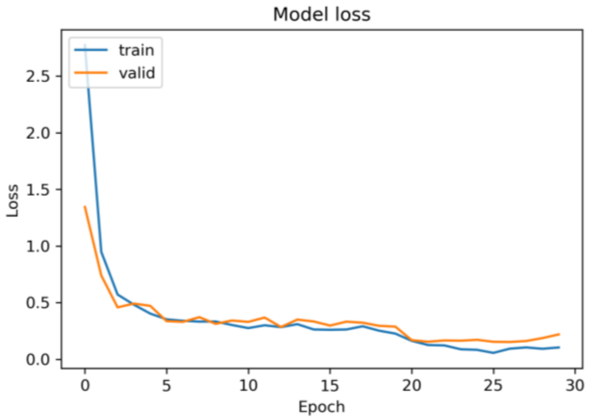
After several attempts on refinement strategies, the proposed CNN model was trained with the following hyper-parameters settings, which resulted in the highest testing accuracy.

- Input image size: 96 x 96
- Epochs: 30
- Batch size: 128
- Optimizer: Adam
- Data augmentation: ZCA whitening and Rotation

As a result, the model achieved a training accuracy of 97.996%, a validation accuracy of 94.929% and a testing accuracy of 94.897%. The plots of change in accuracy and loss, during training and validation for each epoch, is shown in figure 5. The confusion matrix plot of all 156 character classes has been shown in figure 6. Since the testing data contains images that have been manipulated using data augmentation techniques, it can be concluded that the proposed model is robust enough to make predictions on manipulated unseen data and thus the model is reliable.



(a)



(b)

Fig. 5. (a) Accuracy curve during training and validation (b) Loss curve during training and validation

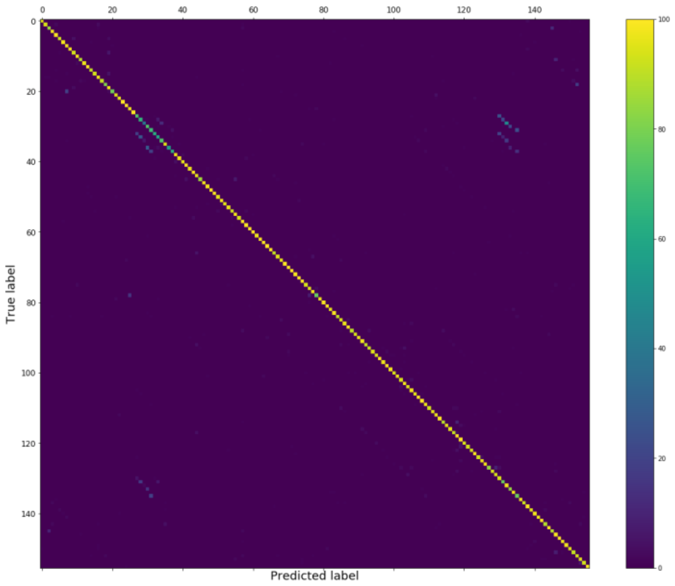


Fig. 6. Confusion matrix of 156 character classes

In this multiclass classification problem, accuracy alone would not be a sufficient metric to evaluate the model's performance, and so precision, recall, and f1-score were also calculated. The average values of all these metrics for 156 classes have been shown in table 2.

Table 2. Classification report

Classes	Precision	Recall	F1-score	Testing samples
156	0.95	0.95	0.95	15600

In the previously published works, no metrics other than the accuracy were used to evaluate the model performance. Consequently, the only way to compare with the results of the previous works is by using testing accuracy. The testing accuracy, 94.897%, of the proposed model is higher than the testing accuracy published in the earlier works. Moreover, this result is the state of the art accuracy in classifying all 156 classes of IWFHR-10 dataset. The comparison of the results of the proposed model with the previous works has been shown in table 3.

Table 3. Comparison of recognition accuracy of the proposed model with other models

	Algorithm	Dataset used		Recognition accuracy
		Name	No. of classes	
Bhattacharya et al. [7]	K-mean clustering and MLP	IWFHR-10	156	89.66%
Vijayaragavan et al. [4]	CNN	IWFHR-10	35	94.4%
Deepa et al. [8]	Hybrid of Levenshtein distance and Genetic algorithm	IWFHR-10	156	89.5%
Deepa et al. [9]	Nearest interest point classifier	IWFHR-10	156	90.2%
Proposed model	CNN	IWFHR-10	156	**94.897%**

IV. CONCLUSION

In this paper, a CNN model for an effective offline Tamil handwritten character recognition system has been proposed. For this, a CNN model that achieved the state of the art accuracy in Chinese HCR system was used and trained end-to-end on IWFHR-10 dataset. After optimizing several hyper-parameters, and implementing the data augmentation technique, the model achieved a testing accuracy of 94.897%, which is the state of the art accuracy in recognizing all 156 characters in IWFHR-10 dataset. From the result, it is evident that CNN has the potential to achieve a groundbreaking result in Tamil HCR. Though the model achieved the state of the art result, the performance can be improved further. Since the dataset was quite small, collecting more data would help to improve the model's performance and also its reliability in prediction. Additionally, the model architecture can be modified so that it would achieve high prediction accuracy with a minimal number of parameters.

Acknowledgment I would like to express my appreciation to HP Labs, India, for making the dataset available for research.

REFERENCES

1. Jebril, Noor A., Hussein R. Al-Zoubi, and Qasem Abu Al-Haija. "Recognition of handwritten Arabic characters using histograms of oriented gradient (HOG)." Pattern Recognition and Image Analysis 28.2 (2018): 321-345.

2. Li, Zhiyuan, et al. "Building efficient CNN architecture for offline handwritten Chinese character recognition." International Journal on Document Analysis and Recognition (IJDAR) 21.4 (2018): 233-240.

3. Yang, Zongjhe, et al. "Character recognition of modern Japanese official documents using CNN for imbalanced learning data." International Workshop on Advanced Image Technology (IWAIT) 2019. Vol. 11049. International Society for Optics and Photonics, 2019.

4. Vijayaraghavan, Prashanth, and Misha Sra. "Handwritten Tamil recognition using a convolutional neural network." (2014).

5. Wahi, Amitabh, and P. Poovizhi. "Handwritten Tamil character recognition." 2013 Fifth International Conference on Advanced Computing (ICoAC). IEEE, 2013.

6. Tamil HCR dataset. Accessed 31 March 2019. Available at: http://shiftleft.com/mirrors/www.hpl.hp.com/india/research/penhw-resources/tamil-iso-char.html

7. Bhattacharya U, Ghosh SK, Parui SK (2007) A two stage recognition scheme for handwritten Tamil characters. In: Proceedings of the ninth international conference on document analysis and recognition (ICDAR 2007). IEEE Computer Society, Washington, DC, 511–515.

8. Nelson, Ashlin Deepa Roselent, and Ramisetty Rajeswara Rao. "A modified GA classifier for offline Tamil handwritten character recognition." International Journal of Applied Pattern Recognition 4.1 (2017): 89-105.

9. Deepa, RN Ashlin, and R. Rajeswara Rao. "A novel nearest interest point classifier for offline Tamil handwritten character recognition." Pattern Analysis and Applications, (2019): 1-14.

10. Mnih, V. et al. Human-level control through deep reinforcement learning. Nature 518, 529–533 (2015).

11. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115.

12. Russakovsky, O. et al. Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. 115, 211–252 (2015).

13. Oquab M, Bottou L, Laptev I, Sivic J. Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the 2014 IEEE conference on computer vision and pattern recognition. 2013. p. 1717–24

14. Weiss, Karl, Taghi M. Khoshgoftaar, and DingDing Wang. "A survey of transfer learning." Journal of Big data 3.1 (2016): 9.

15. Chawla, Nitesh V. "Data mining for imbalanced datasets: An overview." Data mining and knowledge discovery handbook. Springer, Boston, MA, 2009. 875-886.

16. Andrej Karpathy, CS231 Convolutional Neural Network for Visual Recognition, http://cs231n.github.io/convolutional-networks/ Accessed: 13 Apr 2019.

17. Chollet, François. "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

18. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

19. Wensheng, Handwritten Chinese character recognition with deep CNN, (2018), GitHub repository, https://github.com/angzhou/anchor

20. Maas, Andrew L., Awni Y. Hannun and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models." Proc. icml. Vol. 30. No. 1. 2013.
21. Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).
22. X. Cui, V. Goel, and B. Kingsbury. Data augmentation for deep neural network acoustic modeling. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pages 5582–5586. IEEE, 2014.