An Improved Online Tamil Character Recognition Using Neural Networks

M. V. Ishwarya

Department of Computer Science & Engineering,
Hindustan University, Chennai, India
icee.spice@gmail.com

Abstract

In this paper we propose a Conventional Neural Network which recognizes the online Tamil characters. Convolutional Neural Networks are a special kind of multi-layer neural networks. Like almost every other neural networks they are trained with a version of the back- propagation algorithm, where they differ is in the architecture. Convolutional Neural Networks are designed to recognize visual patterns directly from pixel images with minimal preprocessing. They can recognize patterns with extreme variability (such as handwritten characters), and with robustness to distortions and geometric transformations. The existing script-specific post processing schemes improved the recognition rate of online Tamil characters. At the first level, features derived at each sample point of the preprocessed character are used to construct a subspace using the 2DPCA algorithm. Recognition of the test sample is performed using a nearest neighbor classifier. This strategy reduces the recognition error among the confused character sets handled, by more than 4%.

Keywords: *Convolutional Neural Networks, Post Processing, Backpropogation Algorithm, Character Feature Extraction*

1. Introduction

Convolutional neural networks (CNNs) are well suited for solving visual document tasks that rely on recognition and classification. In contrast to fully connected neural networks (NNs), CNNs have been shown to be simpler to build and use. They present a flexible architecture that does not require complex methods, such as momentum, weight decay, structure dependent learning rates, averaging layers, tangent prop, or even finely- tuning the architecture. CNNs have also achieved the state-of-the-art results for character recognition on the MNIST data set of handwritten English digit images.

Tamil is a popular classical language spoken by a significant population in South East Asian countries. There are 156 distinct symbols in Tamil [1]. For the recognition of online Tamil characters, Deepu [2] uses class specific subspaces, while Niranjan et al. [1] have employed elastic matching schemes. Dinesh et al. [3] have recently proposed ‘star -based’ features for the same. Hidden Markov models for recognition have also been reported in [4] [5].

We have a two level scheme for recognizing online Tamil symbols. It is an established fact that one way of assessing the performance of any given classifier depends on how well it can perform on an unknown test sample. To this effect, a confusion matrix is constructed with the training samples of all the 156 classes by employing the leave-one-out cross-validation (LOOCV) technique.

2. Feature Extraction

The strokes of multistroke Tamil characters are first combined into a single trace, retaining the stroke order. Prior to feature extraction and recognition, the input raw character is smoothed to reduce noise. Dehooking algorithms are applied to remove any spurious hooks at the start of the character. The character is then resampled along the trace length to obtain a constant number of points, following which it is normalized by centering and rescaling. Let the number of sample points in the preprocessed character be N_p. At each sample point (x_i, y_i), we extract a set of local features. Let F_{ij} represent the j^{th} feature derived from the i^{th} sample point of the character.

2.1 Character Feature Matrix

Features corresponding to each sample point are stacked to form the rows of a matrix, referred to as the character feature matrix (CFM). The following are the features extracted. The normalized x and y coordinates of the sample points are used as features and are denoted by $F_1^i \; and \; F_2^i$. The distance and angle of the sample point with respect to the the features $F_3^i \; and \; F_4^i$. We divide the length of the preprocessed character into 4 equal segments. The radial distance and polar angle of the sample point of the character with respect to the mean of the segment in which it lies are the features $F_5^i \; and \; F_6^i$. We relate the position of the sample point with respect to its immediate neighbors. We take a sliding window of size W (W odd) centered on the sample point and perform an n order polynomial fit on the samples within that window. We use the resulting n+1polynomial coefficients as features. We Use the values W=3, n=2 (quadratic fit) and accordingly denote the features $F_7^i, F_8^i \; and \; F_9^i$. We separately model the x and y coordinates of the sample point by two n order autoregressive (AR) processes and use the resultant AR coefficients also as features. We employ a 2nd order AR process and accordingly obtain the features F_{10}^i, F_{11}^i, F_{12}^i, $F_{13}^i, F_{14}^i \; and \; F_{15}^i$. It is to be explicitly stated that for obtaining the polynomial fit and AR coefficients of the first and last sample points of the character, we have concatenated the last stroke with the first stroke. This ensures that the notion of neighborhood is not lost. The set of 15 features obtained at a sample point (x_i, y_i) are concatenated to form a feature vector FV^1 of size 1 X 15.

$$FV^1 = [F_1^i \; F_2^i \ldots F_{15}^i] \qquad (1)$$

We then construct the character feature matrix C by stacking the feature vectors of the sample points of the preprocessed character.

$$C = \begin{bmatrix} FV^1 \\ FV^2 \\ \ldots\ldots \\ FV^N \quad , \end{bmatrix} \qquad (2)$$

The i^{th} row of matrix C corresponds to the feature vector derived for the i^{th} sample point. Therefore, the size of C is $N\,p \; X \;\; 15$.

3. Recognition Using 2DPCA

In the 2DPCA method [3], we project the character feature matrix C onto a set of projection axes {P_1, P_2 ,...,P_d} that maximize the total scatter of the projected samples. The projection axes are the orthonormal eigenvectors corresponding to the *d* largest eigenvalues of the character scatter matrix G_t given below:

$$G_t = \frac{1}{N_T}\sum_{j=1}^{N_T}\left(C_j - M\right)^T\left(C_j - M\right) \qquad (3)$$

where *NT* is the total number of training samples,{C1,C2,...,CNT} are the *NT* CFMs and *M* is the mean of the pooled training CFMs. Thus, the size of matrix G_t *is 15*15.* On applying the 2DPCA technique to the character feature matrix *C*, we get a family of principal component feature vectors $\{Y_1, Y_2, ..., Y_d\}$ as defined below:

$$Y_k = CX_k, k = 1,2 \ldots d \qquad (4)$$

The *d* principal component vectors can be stacked column-wise to form the projected feature matrix *B* of dimension Np X d .

$$B = [Y_1 Y_2 \ldots \ldots Y_d] \qquad (5)$$

Let N_T be the total number of training CFMs. After transformation by 2DPCA, we get N_T projected feature matrices.

$$B_i = \left[Y_1^i, Y_2^i, \ldots \ldots Y_d^i\right] \quad i = 1,2 \ldots N_T \qquad (6)$$

Let *Bt* be the projected feature matrix for the test character. The Euclidean distance between the projected feature matrices B_t and B_i *is*

$$d(B_i, B_t) = \sum_{k=1}^{d}\left\|Y_k^i - Y_k^t\right\| \qquad (7)$$

The test character is assigned to the class of the training sample *B*, that satisfies the condition,

$$d(B_i, B_t) = min_i d(B_i, B_t) \qquad (8)$$

4. Analysis of the Confusion Matrix

The multilevel recognition engines [1] generally pass the Top N choices from a previous classifier to the next level..In our work, we exploit prior knowledge of this probability to circumvent the aforementioned drawback of the top N choice approach. We construct a confusion matrix from the training samples using the leave- one-out cross-validation (LOOCV) technique as shown in Figure 1. The rows and columns of the confusion matrix correspond to the true and estimated class labels, respectively. In such a scenario, there is no need to post-process test data recognized as class *j*. A careful analysis of the confusion matrix revealed that if less than 2.5% of the total number of training samples of i^{th} class gets misrecognized to class *j*, they are regarded outliers that are produced mainly due to incorrect styles of writing class that get recognized as the j^{th} class. Accordingly, we do not consider class i^{th} in the post processing module designed for the j^{th} class.

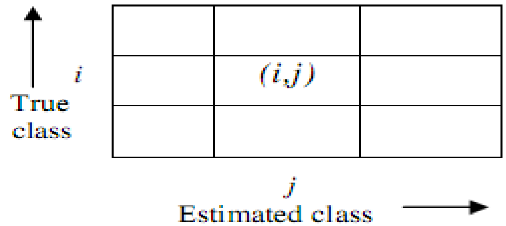

Figure 1. Structure of the Confusion Matrix

Given a test character, we get its estimated class label from the first level classifier. If it corresponds to a class for which the first level classifier is highly discriminative, it is regarded as the final recognition label. Otherwise, the test data is fed to an appropriate post processing module. The output of the second classifier is the final recognition label.

5. Post-Processing Methods

The frequently confused pairs were manually grouped into two categories A and B as shown in Table1. In this section, we propose appropriate post-processing techniques to each group of confused pairs.

Table 1. List of Confused Pairs

Group A	(கி,கீ) (ஙி,ஙீ) (சி,சீ) (ஞி,ஞீ) (ணி,ணீ) (தி,தீ) (நி,நீ) (பி,பீ) (லி,லீ) (வி,வீ) (ளி,ளீ) (னி,னீ) (ஜி,ஜீ)
Group B	(ஏ,ர) (ஐ,ஜ) (க,ச) (ல,வ) (ள,ன) (மு,மூ) (மு,ழு) (டு,டூ) (நு,னு) (நூ,னூ)

5.1 Disambiguating Group A Pairs

The confusions in this group appear between pairs of Tamil consonant-vowel combinations sharing the same base consonant but different vowel modifiers. The most frequently confused vowel modifiers contributing to such errors are the sub strokes ¤ and ¦ Popular writing styles of Tamil script demand that the vowel modifier always forms the last stroke in any multi -stroke consonant – vowel combination character. However, for CV combinations written as a single stroke (where the vowel modifiers get attached to the base consonant), one can regard the subset of sample points traced before the final PEN UP to be the vowel modifier. The number of such sample points is chosen to be a function of the length of the character. It is worth re-emphasizing that the confused pairs in Group A correspond to CV combinations sharing the same base consonant (BC). Let ω_1 and ω_2 denote the class labels of BC+ ¤and BC+ ¦combinations, respectively. We outline below the algorithm employed for distinguishing BC+¤ and BC+ ¦ For a preprocessed character (BC+Vowel modifier combination) resampled to N_p points, let $S = \{(x_i, y_i)\}$ denote the pen coordinates of the extracted vowel modifier. Here '*b'* denotes the pen position of the start of the vowel modifier. A point (x_i, y_i) in S is said to be an 'interest point' if the following two conditions are satisfied.

1.$y_i < y_i - 1$ and $y_i < y_i + 1$

2.$x_i + 1 < x_i$

1. Find the sample point (x_s, y_s) satisfying the relation $y_s = max\ i > b\ y_i$ (see Figure 2)

2. Starting from (x_s, y_s), move along the trajectory to locate interest points, if any. Let N trajectory to locate interest points if any. Let N denote the number of interest points encountered. If $N > 0$, assign the character to class ω_2 If N=0, we invoke (3).

3. Locate the sample point (x_m, y_m) satisfying the relation $x_m = max\ i > s\ x_i$

$$Define\ the\ ratio\ r = \frac{x_m - x_{N_p}}{x_m - x_b} \qquad (9)$$

If r = and $y_{Np} > y_b$ assign the character to class ω_1, is a threshold, empirically set to a value of 0.02.

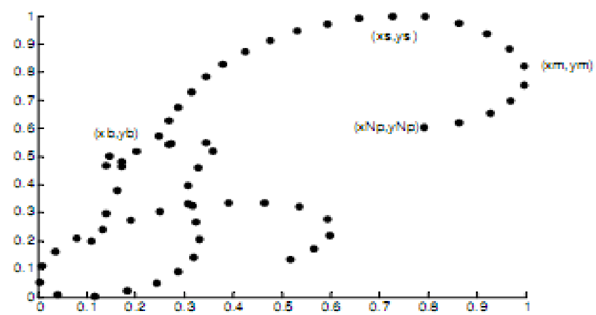

Figure 2. Extraction of script Specific Features from Group A

To illustrate this scheme, consider the character (BC+ Vowel modifier combination) shown in Figure 2 analysis. Accordingly, the character is assigned to class ¸ £ We now give an intuitive reasoning for the proposed post- classification scheme. It is observed that in modern Tamil script, there are many lexemic styles for the sub strokes ¢and and as shown in Fig3 Samples of top row correspond to writing styles of, while those of the bottom correspond to For these cases, mere elastic / rigid matching schemes may not be good enough in distinguishing finer nuances between the sub- strokes ¢and £ In such scenarios, the proposed technique outperforms conventional matching schemes.

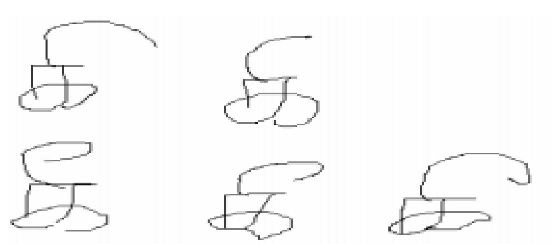

Figure 3. Lexemic Styles of ¸¢ and ¸£. Samples of top and bottom rows correspond to ¸¢ and ¸£ respectively

5.2 Disambiguating Group B Pairs

For this group, Fourier descriptor features corresponding to parts of strokes that discriminate frequently confusing classes are fed to a second level classifier. Similar to Group A pairs, there exist other pairs in Group B that differ predominantly towards the end such as However, the structure to be analyzed for these pairs is strikingly different from those treated in Group A. Moreover, there are certain characters that differ either at the start or middle of their trace such as Such pairs are also incorporated in Group B (See Table 1). As an illustration, consider the characters and . Instead of feeding the (x, y) coordinates of these characters as a whole to the post- processing module, we focus on the shape of sub strokes forming the tails of these characters and extract Fourier descriptors from them, after resampling the extracted shape to 30 points. The number of Fourier coefficients chosen is set empirically to 10. A nearest neighbor classifier is used to obtain the final recognition label of a test character.

6. Proposed Convolutional Neural Network

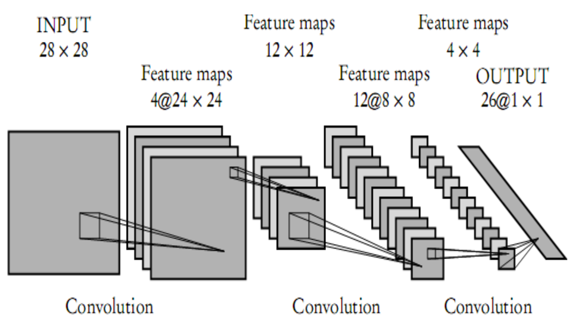

Figure 4. A six-layer Convolutional Neural Network

While the multilayer perceptron performs well for rather simple classification problems, it has several drawbacks when it comes to real-world applications. First, the number of trainable parameters becomes extremely large. For example, a 24×24 input layer would already have 600 connections per single neuron in the hidden layer. Secondly, it offers little or no invariance to shifting, scaling, and other forms of distortion . Third, the topology of the input data is completely ignored, yielding similar training results for all permutations of the input vector. To overcome these problems, a classifier is often split up into a hand-crafted feature extractor and the actual trainable classifier module. Designing the feature extractor by hand requires a lot of heuristics and, most notably, a great deal of time. The concept of convolutional neural networks, which were neurobiologically motivated by the findings of locally sensitive and orientation-selective nerve cells in the visual cortex of the cat. They designed a network structure that implicitely extracts relevant features, by restricting the neural weights of one layer to a local receptive field in the previous layer. Thus, a feature map is obtained in the second layer. By reducing the spatial resolution of the feature map, a certain degree of shift and distortion invariance is achieved. Also, the number of free parameters is significantly decreased by using the same set of weights for all features in the feature map.

6.1 Topology

A simple convolutional neural network is shown in Figure 4. As can be seen, the first convolutional layer contains four feature maps, where each neuron has a receptive field of 5 × 5 in the input image. To eliminate the need for boundary processing, the feature maps have a slightly smaller resolution than the input [1]. The first subsampling layer consists of four feature maps, which are scaled local averages of the respective feature maps in the previous convolutional layer. Each neuron in the subsampling layer has one trainable coefficient for the local average, and one trainable bias . The next convolutional and subsampling layers operate in the same manner. However, the second convolutional layer has 12 feature maps, resulting in an increase in feature space . Similarily, the subsampling layers reduce the spatial resolution. The last convolutional layer is the output layer and contains only one feature map . To obtain the final result of the classifier, the neurons of the output layer can then be compared to detect a winner neuron .Potentially, each feature map receives input from all feature maps in the previous layer. However, in order to prevent all feature maps from learning the same, this is usually not the case.

6.2 Training

As a special form of the multilayer perceptron, convolutional neural networks are trained through backpropagation. Because all neurons in one feature map of a convolutional layer share the same weights and bias, the number of parameters is dramatically smaller then in a fully interconnected multilayer perceptron, leading to an implicit reduction of the gap e_{test} - $e_{train.}$ The subsampling layers have one trainable weight (local average coefficient) and one trainable bias, so the number of free parameters in the subsampling layers is even lower than in the convolutional layers. Because for this low number of free parameters, the training of convolutional neural networks requires far less computational effort than the training of multilayer perceptrons. This, as well as the implicit feature extraction and distortion invariance (to some degree), make convolutional networks an obvious candidate for classification tasks, especially pattern recognition. They have successfully been used for various pattern recognition tasks, such as handwriting and face recognition.

7. Experimental Results

The performance data for training set and the test set are listed in Table 2.These include the percentages that the scores of the desired class are within top 1, 2, 3 highest scores. We also simulate the performance for recognition by using 1-layer fully connected neural network and 2-layer fully connected neural network in Table 3.

Table 2. Performance for Multiple Convolutional Neural Networks

Top	Training Set	Test Set
1	98.8%	94.2%
2	99.2%	98.4%
3	99.4%	98.8%

Table 3. Performance for 1-Layer Fully Connected N.N and 2-Layer Fully Connected NN

	Training Set	Test Set
1-Layer Fully Connected	99.9%	80.6%
2-Layer Fully Connected	99.9%	91.8%

8. Conclusion

In this work we have adopted the neural networks to recognize online handwritten Tamil characters using a multilayer perception with one hidden layer. The feature extracted from the handwritten character is Fourier descriptors. Also an analysis was carried out to determine the number of hidden layer nodes to achieve high performance of back propagation network in the recognition of handwritten Tamil characters. The system was trained using several different forms of handwriting provided by both male and female participants of different age groups. Test results indicate that Fourier descriptors combined with back propagation network provide good recognition accuracy of 97% for handwritten Tamil characters.

9. Discussion and Related Work

The two level recognition technique is tested on the IWFHR 2006 Tamil Competition dataset. This dataset contains 26926 random test samples (approximately 177 test samples per character). We have used 270 training samples for each character. The characters are resampled to 60 points and normalized to [0, 1]. We compute features at each point of the resampled characters used for training and construct character feature matrices of size 60×15. We then transform the features to an 8-dimensional subspace by performing 2DPCA on the training CFMs. A nearest neighbor classifier is used to classify the test character in the subspace. If the estimated class label is one of the confusion pairs in Table 1, we input the test character to an appropriate postprocessing scheme at the next level. There is an increase in the classification accuracy of a few frequently confused characters after the post processing step. The improvement in performance is observed in both the validation/ training and test sets. Validation on the training set is performed using the leave-one-out cross-validation (LOOCV) technique. On the average, there is an 8 % reduction in the recognition error among the confused characters in the validation set. On the test set, the improvement in recognition is around 5%.

References

[1] N. Joshi, G Sita and A. G. R. S. Madhavanath, “Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition”, Proc. Intl Workshop Frontiers Handwriting Recog., **(2004)**, pp. 444- 449.

[2] V. Deepu and S. Madhavanath, “Principal Component Analysis for Online Handwritten Character Recognition”, Proc. Intl Conf. Pattern Recog., vol. 2, **(2004)**, pp. 327-330.

[3] M. Dinesh, M. K. Sridhar, “A Feature based on Encoding the Relative Position of a Point in the Character for Online Handwritten Character Recognition”, Proc. Intl. Conf. Doc. Anal. Recog., vol. 2, **(2007)**, pp. 1014- 1017.

[4] A. H. Toselli, M. Pastor and E. Vidal, "On-Line Handwriting Recognition System for Tamil Characters", Proc. Iberian conf. Pattern Recog. Image Anal., Lecture Notes Comp. Science, vol. 4477, no. 1, **(2007)**, pp. 370-377.
[5] A. Bharath and S. Madhvanath, "Hidden Markov Models for Online Handwritten Tamil Word Recognition", Proc. Intl. Conf. Doc. Anal. Recog., vol. 1, **(2007)**, pp. 506-510.
[6] Y. Jiang, Z. David, F. Alejandro and J. Yang, "Two Dimensional PCA: a New Approach to Appearance based Face Representation and Recognition", IEEE Trans. PAMI, vol. 26, no. 1, **(2004)**, pp. 131- 137.
[7] http://www.hpl.hp.com/india/research/penhwinterfaceslinguistics.html#datasets
[8] G. Nagy, "Twenty years of document image analysis in pattern analysis and machine intelligence", IEEE Tran. PAMI, **(2000)**, pp. 38–82.
[9] J. Mantas, "An overview of character recognition methodologies", Pattern recognition, vol. 19, no. 6, **(1986)**, pp. 425-430.
[10] V. K. Govindan and A. P. Shivaprasad, "Character Recognition-A Review", Pattern Recognition, vol. 23, no. 7, **(1990)**, pp. 671-683.
[11] S. Hewavitharana and H. C. Fernando, "A Two Stage Classification Approach to Tamil Handwriting Recognition", **(2002)**, pp. 118-124, Tamil Internet 2002, California, USA.
[12] P. Chinnuswamy and S. G. Krishnamoorthy, "Recognition of Hand printed Tamil Characters", Pattern Recognition, vol. 12, **(1980)**, pp. 141-152.
[13] R. M. Suresh, S. Arumugam and K. P. Aravanan, "Recognition of handwritten Tamil characters using fuzzy classificatory approach", Proc. The Tamil Internet 2000 Conference, Singapore, **(2000)** July.
[14] Siromoney, et. al., "Computer Recognition of Printed Tamil Character", Pattern Recognition, vol. 10, **(1978)**, pp. 243-247.
[15] H. Bunke, M. Roth and E. G. Schukat – Talamazzini, "Offline Cursive Handwriting Recognition using Hidden Markov Models, Pattern Recognition, vol. 28, no. 9, **(1995)**, pp. 1399–1413.

Author

M. V. Ishwarya is pursuing Ph.D. in Department of Computer Science and Engineering in Hindusthan University, Chennai, Tamilnadu India. Her area of interest includes Image Processing, Pattern Recognition, and Optimization Techniques.