Optical Character Recognition of Tamil characters based on DCT features using SVM and Neural Network

T.Ganesh[1] and **Niranj Jyothish**[2]

[1]Department of CSE, VIT University, Chennai Campus,
Vandalur-Kelambakkam Road, Chennai (India);
Email : ganesh.t2013@vit.ac.in

[2]Department of CSE, VIT University, Chennai Campus,
Vandalur-Kelambakkam Road, Chennai (India);
Email : niranj.jyothish2013@vit.ac.in

ABSTRACT

Our work aims to design an Optical Character Recognition (OCR) system which can identify handwritten Tamil characters. Typically, OCR involves four different stages, namely - Pre-processing, feature extraction, Character Recognition and Post-processing. In this work, the block-wise Discrete Cosine Transform (DCT) was used for feature extraction. Classification was done using two different techniques namely, Support Vector Machines (SVM) and Neural Network. Also, a performance analysis for the above classifiers is done. The classification was conducted for thirty-four Tamil alphabets.

Keywords—Optical Character Recognition, Handwritten Tamil Character Recognition, Feature Extraction, Block-wise DCT, SVM Classifier, Neural Network Classifier.

Computing Classification System: 1.5.2

1. INTRODUCTION

Optical Character Recognition is defined as the identification of printed or written characters by a computer. This identification facilitates the process of bridging the gap between man and machine communication. During the recent years, character recognition has taken a jump in the field of research and other applications like conversion of handwritten documents to an editable soft format, recognition of postal addresses for automated postal system, data and word processing, data acquisition in bank cheques, processing of archived institutional records etc.

Since OCR's inception in the year 1914, it has been observed that most of the research work done in the field of OCR is solely based on the English Language. But, research done on OCR systems for native Indian languages is comparatively less in number. In case of Tamil, an Indian language, OCR is a challenge because of the high degree of similarity among the characters.

In our work, discrete cosine transform (DCT) plays a major role in feature extraction. The usage of DCT for feature extraction has been seen in Character recognition and Facial Recognition.

Lawgali performed comparative analysis of DCT and discrete wavelet transformation. The results showed that DCT was highly effective in extracting important features, thus yielding a better recognition rate. Qacimy *et al*. compared four different feature extraction processes based on discrete cosine transform. These approaches were, (i) DCT upper-left corner (UCL) coefficients, (ii) DCT zigzag coefficients, (iii) Block-based DCT UCL coefficients and (iv) Block-based DCT zigzag coefficients. The results indicated that the highest accuracy was obtained by using block-based zigzag coefficients as features.

These features are then fed as input to a classification model. The classification models used in this work are Support Vector Machine (SVM) and Neural Networks. Support Vector Machine is a classifier which uses kernels to generate an optimal separating hyper-plane (decision boundary) among classes in higher dimensional feature space. Vapnik introduced the SVM algorithm in his book-"The nature of statistical learning theory".

2. DATASET

The database used is taken from Lipi toolkit which is an open source toolkit for online Handwriting Recognition (HWR), created by HP Labs India. This dataset called 'Isolated Handwritten Tamil Character Dataset' contains approximately 500 isolated samples each of 156 Tamil 'characters' written by native Tamil writers including school children, university graduates, and adults from Bangalore, Karnataka and Salem, Tamil Nadu. Some sample images from the dataset are shown in figure 1.

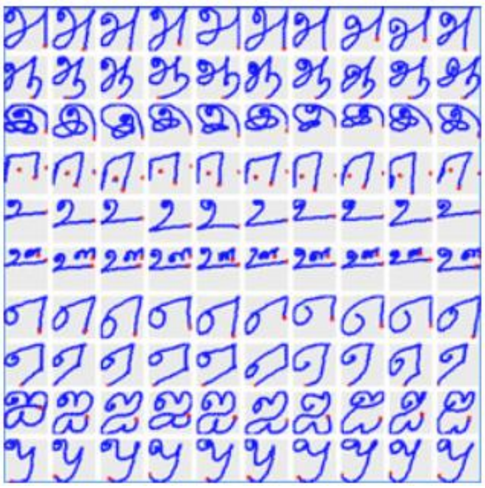

Figure 1. Sample of Lipi toolkit's Isolated Handwritten Tamil Character Dataset.

3. PROPOSED MODEL

The proposed model has been divided into the following sections – (I) Pre-processing, (II) Feature Extraction, (III) Classification and (IV) Post-Processing. Our model consists of the following stages:

3.1 Pre-Processing

Before an input image is fed to a classifier, it should be pre-processed so that all the important features of the image can be extracted effectively. Pre-processing is necessary because an image might have some unwanted noise, undesirable size, the image might have an incompatible format, etc. First the image is subjected to binarization using a specific threshold.

Then, a median filter is applied on the binarized image to remove image noise, which include salt and pepper noise, speckle noise etc. After the noise is filtered from the image, the individual characters in the image are separated. The images of these separate characters were found to have an excess of white space, which was removed. The image is ready for feature extraction after the completion of all the above processes.

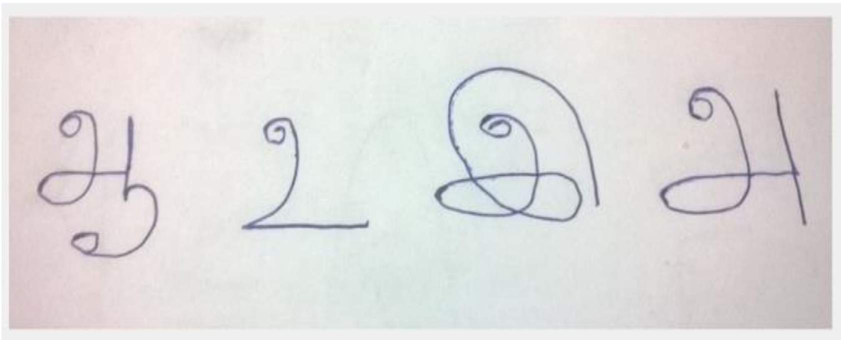

Figure 2. Image before Pre-processing.

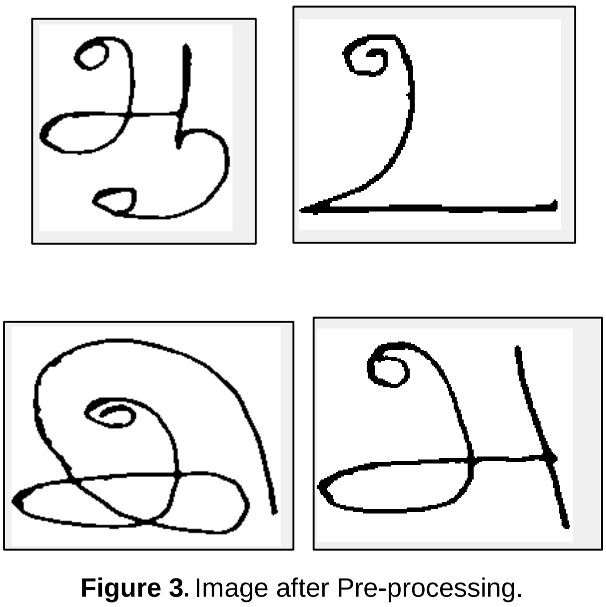

Figure 3. Image after Pre-processing.

3.2. Feature Extraction

An important part of OCR is the extraction of desirable features from the image of a character. The features of an image need to project the unique characteristics of a letter (character), which can be used to distinguish it from others. Before any features are extracted, the image size is reduced to 64X64 pixels. This image, in turn, is divided into 64 blocks, each having size 8X8. On each of these blocks, the discrete cosine transform (DCT) is applied. The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT has the property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT. For this reason, the DCT is often used in image compression applications. For most images, much of the signal energy lies at low frequencies, which appear in the upper left corner of the DCT. After DCT has been applied on the 8 X 8 blocks, for each block, the DCT coefficients are extracted with the help of raster scan. The scanned pixels are stored in a 1-D array. Since the signal energy of an image lies in the upper left corner, the first four values in the 1-D array are preserved, while the rest are pruned. The 1-D array is appended with the specific DCT coefficients for each block that undergoes raster scan. The vector thus obtained would have 256 coefficients which act as the significant features for a particular character.

The general equation for a 2-D (*M* by *N* image) DCT is defined by the following equation:

$$DCT(p,q) = \alpha_p q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2m+1)q}{2N}, \quad 0 \le p \le M-1, 0 \le q \le N-1$$

where,

$$\propto_{\boldsymbol{p}} = \begin{cases} \frac{1}{\sqrt{M}}, p = 0 \\ \sqrt{\frac{2}{M}}, 1 \le p \le M-1 \end{cases}$$

and

$$\propto_{\boldsymbol{q}} = \begin{cases} \frac{1}{\sqrt{N}}, q = 0 \\ \sqrt{\frac{2}{N}}, 1 \le q \le N-1 \end{cases}$$

M and N are rows and columns of input image, A respectively. DCT (p,q) represents the coefficient calculated afor pixel A(p,q).

3.3. CLASSIFICATION

3.3.1. Support Vector Machine

There are two popular methods for multiclass classification using SVM, namely, one-versus-all method using winner-takes-all strategy, and the one-versus-one method implemented by max-wins voting. The training process of the one-versus-one method is quicker than that of the one-versus-all method. Also, it is more suitable for classification problems with larger number of labels. Hence, the one-versus-one method has been used.

In this strategy, for every distinct pair of M classes, the one-versus-one approach generates binary classifiers. After training, there would be (M x (M-1) /2) binary classifiers denoted by C_{ij} ,where i and j are two distinct labels . For each test image x, fed to the trained algorithm, if classifier C_{ij} categorizes it in class represented by label 'i', then the vote for that class is incremented by one. Otherwise, the vote for class 'j' is incremented by one. After each binary classifier casts its vote, the test image 'x' is assigned with the class having maximum number of votes. The kernel of each binary classifier is a linear polynomial.

3.3.2. Neural Network

We have used a four-layer feed-forward backpropagation network, with three hidden layers and the last layer being the classification layer.

4. RESULTS AND DISCUSSION

The experiment is conducted on the Lipi toolkit's Handwritten Tamil dataset. 125 images for each of the 34 alphabets are used for the experiment. After performing feature extraction on the images (mentioned in the proposed model), the 256 features of each of the 4250 images are given as input to the classification models – SVM and Neural Network.

4.1. Classification using SVM

Each of the feature vectors is appended with its respective class label and given as input to the multi-class SVM model. The one-versus-one, max-voting method, has been used for training. To cross-validate the results obtained from the model, k-fold cross validation is performed with k = 10. In k-fold cross validation, the whole dataset (feature vector matrix) is divided into k equal parts, P_1, P_2….P_k. The validation is done k times. For every i_{th} iteration, P_i is held out for testing while the other k-1 parts are used for training. The accuracy achieved is 81.2 %.

4.2. Classification using Neural Network

Each of the feature vectors is appended with its respective class label and given as input to the Convolution neural network. The CNN is constructed such that there are three hidden layers with 78, 40, and 20 neurons, respectively. The network has been depicted in Fig. 4. 75% of the dataset is used for training, 15% for validation, while the rest 15% for testing. The accuracy achieved from this model is 76.6 %.

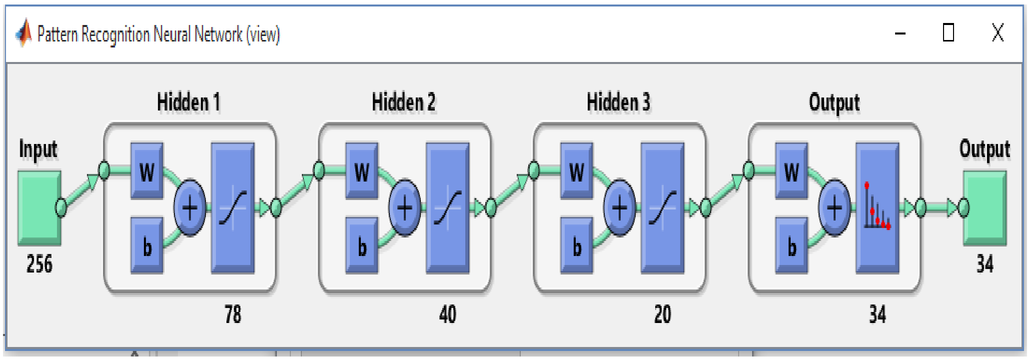

Figure 4. Architecture of Neural Network.

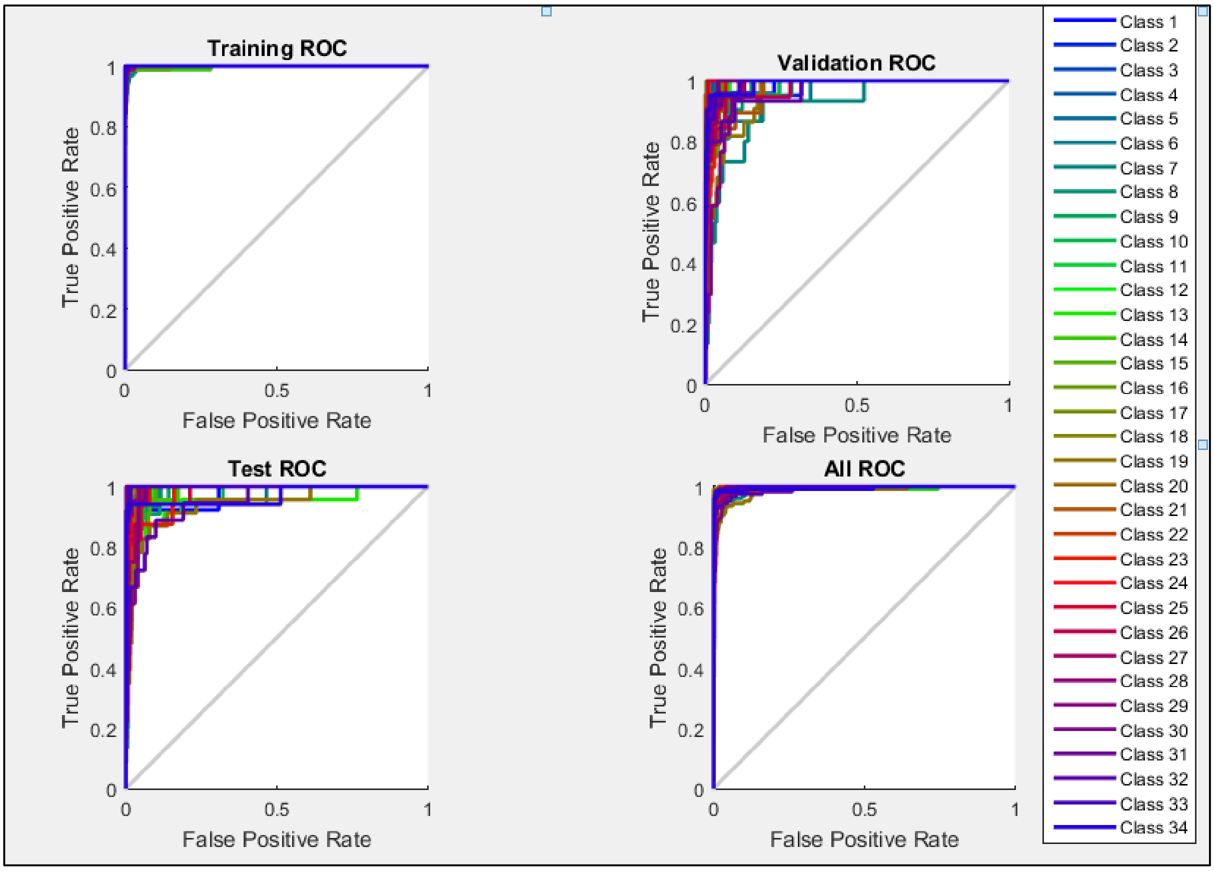

Figure 5. ROC curves of Training, Validation and Test data

4.3. Discussion

Figure 5 depicts the Class wise comparison of SVM and Neural Network. From the bar plot, it is clear that for a majority of Tamil Characters, SVM performed better than the Neural Network.

5. CONCLUSION

In this work, we have designed an optical character recognition system to identify thirty four characters (eleven vowels and twenty-two consonants) of the Tamil language. This system has been trained using the Lipi Toolkit's 'Isolated Handwritten Tamil Character Dataset'. We have used block-wise DCT for feature extraction. Convolution Neural Network and Multi-class Support Vector Machine have been for the classification of the testing data. It was found that SVM had a better recognition rate than Neural Net.

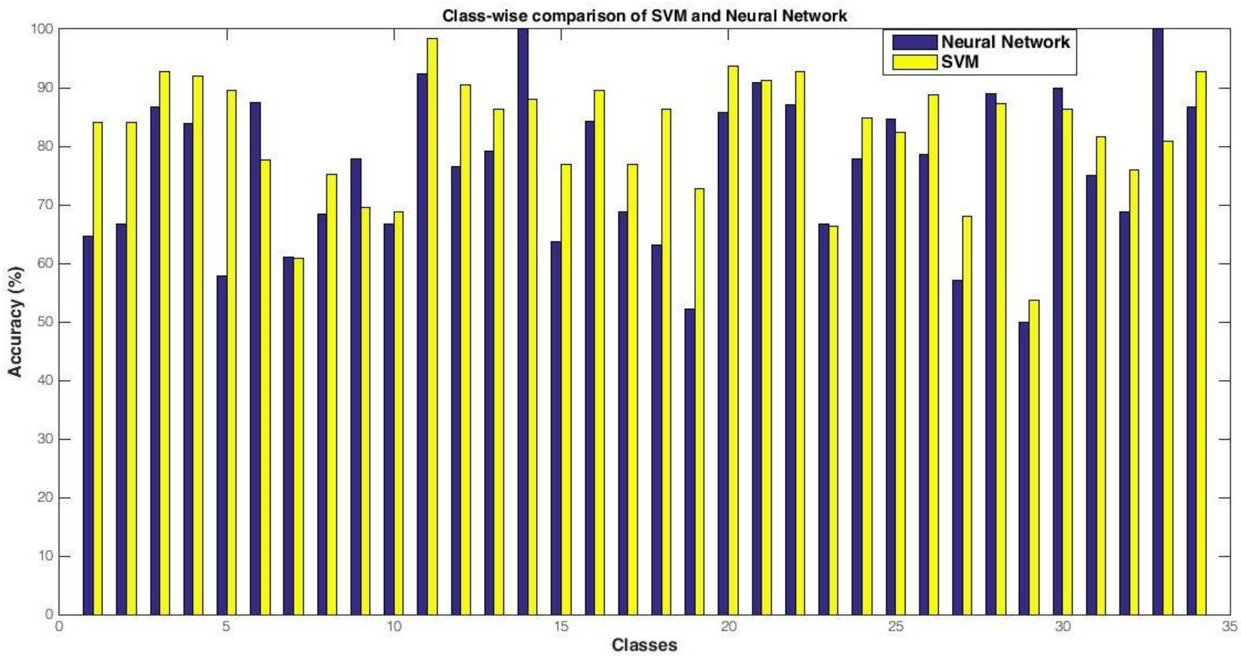

Figure 5. Class-wise comparison of SVM and Neural Network

6. REFERENCES

Aman, R. Chadha, Pallavi, P. Vaidya and Mani Roja, M., 2011, *Face Recognition Using Discrete Cosine Transform for Global and Local Features*, Proceedings of the 2011 International Conference on Recent Advancements in Electrical, Electronics and Control Engineering (IConRAEeCE)
.

Bharath, A. and Madhvanath, Sriganesh, 2009, *Online Handwriting recognition for Indic Scripts*, Advances in Pattern Recognition pp 209-234, Spring-Verlag London Limited 2009.

Bouchra El, Qacimy, Mounir Ait kerroum and Ahmed, Hammouch, 2014, *Handwritten Digit Recognition based on DCT features and SVM Classifier*, 2014 Second World Conference on Complex Systems (WCCS).

Hsu, Chih-Wei. and Lin, Chih.-Jen, 2002, *A comparison of methods for multi-class support vector machines*, IEEE transactions on Neural Networks, vol. 13, pp. 415-425.

Jonathan, Milgram, Cheriet, Mohamed and Robert, Sabourin, 2006, "*One Against One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs?*, Guy Lorette. Tenth International Workshop on Frontiers in Handwriting Recognition.

Lawgali, A., Bouridane, A., Angelova, M. and Ghassemlooy, Z., 2011, *Handwritten Arabic character recognition: Which feature extraction method*, International Journal of Advanced Science and Technology, vol. 34, pp.1–8.

Munish, Kumar, Jindal, M.K. and Sharma, R.K., 2011, *Review on OCR for Handwritten Indian Scripts Character Recognition*, Springer-Verlag Berlin Heidelberg.

Sriganesh, Madhvanath,Deepu, Vijayasenan and Thanigai, Murugan Kadiresan, 2006, *LipiTk: a generic toolkit for online handwriting recognition*, 10th national workshop on frontiers in handwriting recognition (IWFHR), La Baule, France.

Saidas, S.R., Rohithram T., Sanoj K.p. and Manju Manuel, 2016, *Malayalam Character Recognition using Discrete Cosine Transform*, International Journal Of Engineering And Computer Science,Volume 05, Issue 2.

Vapnik, V. N.,1995, “The nature of statistical learning theory”, Springer-Verlag New York.