Off-line Myanmar Character Recognition based on Competitive Neural Trees

Theingi Htike
University of Computer Studies, Yangon
htike83@gmail.com

Dr.Yadana Thein
University of Computer Studies, Yangon

Abstract

Neural network classifier methods and decision trees are widely used in various pattern recognition research areas. Among them, printed character recognition still faces some issues in all languages. Myanmar sentences character recognition based on Competitive Neural Trees (CNeT) is proposed in this paper. CNeT performs hierarchical classification and apply competitive unsupervised learning at node label. The goals of Myanmar character recognition are to obtain better recognition accuracy rate and robust in geometric character shapes of different styles. Three main steps such as preprocessing, shape feature descriptors extraction and recognition are implemented in our experiment. Shape feature descriptors are extracted from preprocessed images which are used in Competitive Neural Trees (CNeT) recognition. This paper discusses a global search method for the CNeT, which is utilized for training.

Keywords – Myanmar printed characters, CNeT, global search method

Introduction

Optical Character Recognition (OCR) is a field of research in pattern recognition, artificial intelligence and machine vision. It refers to the mechanical or electronic translation of images of handwritten, typewritten or printed text into machine-editable text. Nowadays, the accurate recognition of machine printed characters is considered largely a solved problem. The domain of OCR recognition has two completely different problems of On-line and Off-line character recognition.

On-line character recognition [1] involves the automatic conversion of characters as it is written on a special digitizer, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. The off-line character recognition is comparatively difficult, as different people have different handwriting styles and also the characters are extracted from documents of different intensity and background [2].

A review of the character recognition work done on Myanmar languages is excellently reviewed. Paper [5] proposed a system to recognize off –line Myanmar handwriting. They used discrete Hidden Markov Model. In paper [6] proposed Handwritten Myanmar Optical Character Recognition System. They used histogram labeling method for recognition. Recognition accuracy rate 98.18% was obtained.

Paper [8] proposed an effective recognition approach for Myanmar Handwritten Characters. Hybrid approach use MICR (Myanmar Intelligent Character Recognition) and back-propagation neural network. In Hybrid approach, the features of MICR have been used in back-propagation neural network as input nodes. The back-propagation algorithm has been used to train the feed-forward neural network and adjustment of weights to require the desired output. Using Hybrid approach, over-all recognition accuracy of 95% was obtained.

The Myanmar language is the official language and is more than one thousand years old. Myanmar script is considered a complex script by software developers, as it originated from Indic scripts like Thai or Khmer. The Myanmar (formerly known as Burmese) script developed from the Mon script, which was adapted from a southern Indian script during the 8th century.

In this paper, competitive neural trees are proposed for character recognition. In Myanmar character recognition fields, competitive neural trees had not been applied for recognition. Therefore, competitive neural trees are applied to develop Myanmar character recognition system. It is one of the fast supervised neural networks. Neural trees were introduced for character recognition in an attempt to combine advantages of neural networks and decision trees. The aim of the proposed system is to implement an effective approach which is able to recognize for Myanmar printed characters.

The remainder of the paper is organized as follows: **Section 2** describes Myanmar language Nature. **Section 3** explains proposed system design and the various steps involved in the OCR System. **Section 4** presents competitive neural trees, **Section 5** describes the experimental results and Section 6 explains the conclusion.

2. Myanmar Language Nature

The interests in Myanmar characters recognition research have grown over the past few years but practical research is only a few works in research field. Because, the problem of Myanmar characters recognition is more difficult than English languages in respects including the similarity of characters, absence space between each word, etc. So, various character recognitions method not enough complete recognize Myanmar character. They are still in research field, not complete work.

The Myanmar language belongs to the Sino-Tibetan family of languages of which the Tibetan-Myanmar (Tibeto-Burman) subfamily forms a part. It has been classified by linguists as a monosyllabic or isolating language with agglutinative features. It is a tonal and analytic language. Texts in the Myanmar language use the Myanmar script, which derives from a Brahmirelated script borrowed from South India in about the eight century for the Mon language. The first inscription in Burmese dates from the following years and is written in an alphabet almost identical with Mon inscriptions.

2.1 Myanmar Language Pattern

Myanmar language is the official language and widely used in many Myanmar states. Myanmar Language includes Kachin, Kayar, Kayin, Chin, Myanmar, Rakhine and Shan, etc. Myanmar language consists of (33) consonants, (12) vowels and (4) medials.

က ခ ဂ ဃ င စ ဆ ဇ ဈ ည
ဋ ဌ ဍ ဎ ဏ တ ထ ဒ ဓ န
ပ ဖ ဗ ဘ မ ယ ရ လ ဝ သ
ဟ ဠ အ

Figure.1. A Set of Myanmar Characters

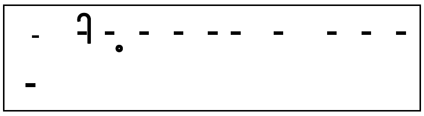

Figure.2. A Set of Myanmar Vowels

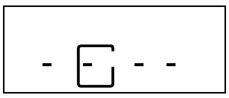

Figure.3. A Set of Myanmar Medials

3. Proposed System Design

A typical character recognition system is characterized by a number of steps, which include

(1) Image Acquisition
(2) Preprocessing
(3) Feature Extraction, and
(4) Recognition.

The steps required for character recognition are described here in detail:

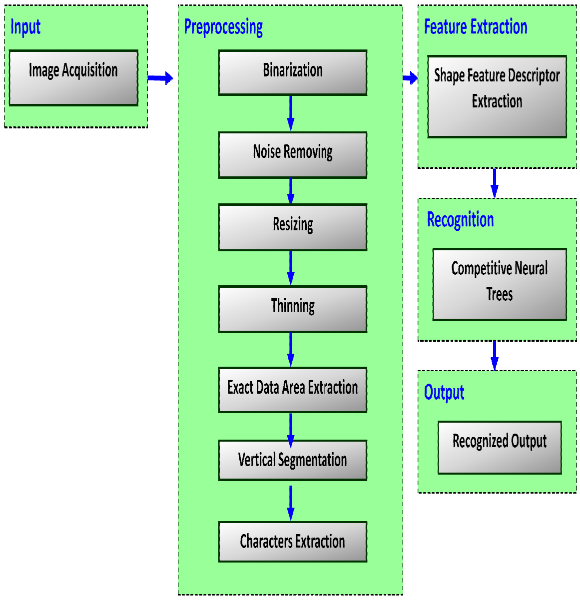

Figure.4. Proposed System Design

3.1 Image acquisition

It performs getting image from other sources like scanned image. The following Myanmar characters are scanned with Canon Scan Lide 110 model.

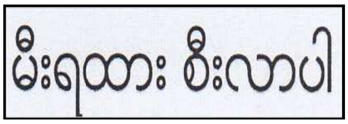

Figure.5. Input Sentence of Myanmar Characters

3.2 Preprocessing

Preprocessing aims at eliminating the variability that is inherent in printed characters. The preprocessing techniques that have been employed in an attempt to increase the performance of the recognition process are as follows:

Binarization: All printed characters are scanned into gray scale images. Each character image is traced vertically after converting the gray scale image into binary matrix.

Figure.6. Binary Image of Myanmar Characters

Noise Removing: The presence of noise can cost the efficiency of the character recognition system. Morphological operations are used to remove small objects from a binary image all connected components that have fewer than pixels.

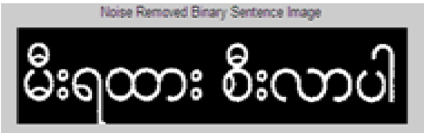

Figure.7. Noise Free Image of Myanmar Characters

Resizing: The input image may have different size, which will affect the recognition results. Since the images are various sizes, we have to convert standard size image. The standard size we used is 100 x 100 pixels.

Figure.8. Resize Image of Myanmar Characters

Thinning: Thinning is a morphological operation that removes selected foreground pixels from binary images. The thinning process reduces the width of pattern to just a single pixel.

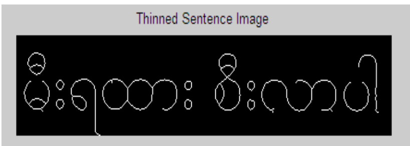

Figure.9. Thinning Image of Myanmar Characters

Exact Data Area Extraction: Each input sentence is extracted row and column to get exact data area for each input sentence.

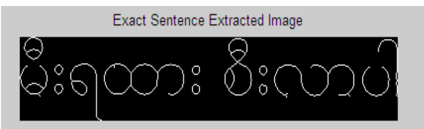

Figure.10. Exact Sentence Image of Myanmar Characters

Vertical Segmentation: Segmentation is an important task of any OCR system. It separates the image text documents into lines, words and characters. The accuracy of OCR system mainly depends on the segmentation algorithm being used. Vertical segmentation algorithm is proposed for segmenting the input text line. Figure 12 shows the vertical segmentation algorithm. Its input is single text line and its output is multiple characters.

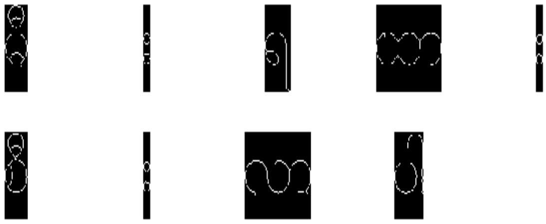

Figure.11. Vertical Segmentation Image of Myanmar Characters

```
% Vertical Segmentation Algorithm
% parameter input sentence
% return vertical_segmented_chars
BEGIN
        Initialize start index to 0;
        Initialize end index to 0;
        Initialize output index to 1;
        Initialize i to 1;
        [Rows Cols]=size (input sentence);
        start index =1;
          while (i<cols)
           while(i<cols)
             if (colsum (1,i)==0)
               end index=i-1
               break;
             end
              i=i+1;
             end
            if i==cols
             break;
            end
           Crop and assign to onech with [start index, 1, end
        index-start  index, rows-1]
           if(end index-start index>3)
             charsarray{k,1}=onech;
             k=k+1
           end

           while(i<cols)
             if(colsum(1,i)>0)
               start index=i
               break;
             end
             i=i+1
           end

    Crop and assign to onech with [start index, 1, end index-start
index, rows-1]
        charsarray {k,1}=onech;
End
```

Figure.12. Vertical Segmentation Algorithm

Characters Extraction: After the vertical segmentation, characters are needed to extract to get one char. Characters Extraction algorithms such as char _longyitinsankhart and char_yachar algorithms are described in the figure 13 and 15.

```
char_ longyitinsankhart algorithm

%parameter1 B is char_longyitinsankhart

% parameter2 T is threshold value of char's height

% return char and longyitinsankhart

Begin

temp=B;

[r c]=size (temp)

Char= temp (1: T, :)

Longyitinsankhart = temp (T+1: r, :)

End
```

Figure.13. Char_longyitinsankhart Algorithm

Figure.14. Char Extraction of Longyitinsankhart

```
char_yachar algorithm

% parameter1 B is char_yachar

% parameter2 T is threshold value of char's width

% return char and yachar

Begin

temp=B;

[r c]=size(temp)

char =temp (:,1:T)

yachar =temp (:,T+1:c)

End.
```

Figure.15. Char_yachar Algorithm

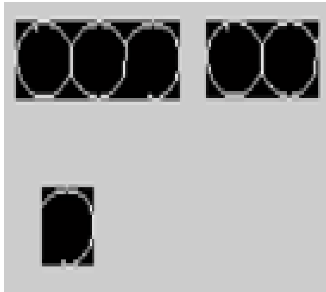

Figure.14. Char Extraction of yachar

Feature extraction: Selection of feature extraction methods is one of the key steps in achieving high recognition accuracy. After pre-processing, features for each character image are extracted based on Shape description. Shape description techniques can be generally classified into two classes of methods: contour-based methods and region-based methods. In this paper, region-based methods are used to extract features .The wide range of shape variations for characters requires an adequate representation of the discriminating features for classification. Eighteen features are extracted for the entire image based on the regional properties namely:

Area: It is calculated the actual number of pixels in the region.

Bounding Box: It is calculated the smallest rectangle containing the region.

Centroid: It is calculated the center of mass of the region. The first element of centroid is the horizontal coordinate of the centre of mass and the second element is the vertical coordinate.

Major Axis Length: It is calculated the length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region.

Minor Axis Length: It is calculated the length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the region.

Eccentricity: It is calculated as the ratio of the distance between the centre of the ellipse and its major axis length.

Euler Number: It is calculated as the difference of Number of Objects and Number of holes in the image.

Orientation: It is the angle between the x-axis and the major axis of the ellipse that has the same second-moments as the region.

Extent: It is calculated as the ratio of pixels in the region to the pixels in the total bounding box.

Equiv Diameter: It is calculated the diameter of a circle with the same area as the region.

Solidity: It is calculated the proportion of the pixels in the convex hull that are also in the region.

Perimeter: It is calculated the distance around the boundary of the region.
Filled Area: It is calculated the number of pixels in filling Image.
Convex Area: It is calculated the number of pixels in 'Convex Image'. Convex Image is binary image (logical) that specifies the convex hull, with all pixels within the hull filled in. Convex Hull specifies the smallest convex polygon that can contain the region.

Table.1. Feature vectors of each character

Character	က	ခ	ဂ	ဃ	င
Area	176	123	92	176	97
Centroid	39.75,17.86	20.96,21.67	19.8,16.84	37.41,23	17.48,19.62
Bounding Box	0.5,0.5,77,39	0.5,0.5,39,39	0.5,0.5,38,39	0.5,0.5,38,39	0.5,0.5,40,39
Major Axis	100.1	52.16	56.82	100.1	56.58
Minor Axis	52.78	46.64	48.06	53.44	50.42
Eccentricity	0.85	0.448	0.533	0.846	0.454
Orientation	-1.27	83.68	-1.72	-1.91	88.85
Convex Area	2636	1202	1179	2707	1240
Filled Area	176	349	92	176	97
Euler Number	1	0	1	1	1
Equiv Diameter	14.97	12.51	10.82	14.97	11.11
Solidity	0.067	0.102	0.078	0.065	0.078
Extent	0.059	0.081	0.062	0.058	0.062
Perimeter	411.2031	227.62237	213.48023	413.1026	225.96551

3.3 Character Recognition with CNeT

This is the stage where an automated system declares that the inputted character belongs to a particular category. Competitive neural trees are used to recognize characters. An efficient character recognition scheme must be capable of producing appropriate class labels for input vectors that do not belong to the training set. This is called generalization.

Paper [4] proposed a structural adaptive intelligent tree (SAINT). The input feature space is hierarchically partitioned by using a tree-structured network that preserves a lattice topology at each sub network. Experimental results reveal that SAINT is very effective for the classification of a large set of real-world handwritten characters.

In paper [3], they proposed balanced neural tree to reduce tree size and improve classification with respect to classical neural tree. Two main innovations have been introduced (a) perceptron substitution and (b) pattern removal. The first innovation aims to balance the structure of the tree .If the last-trained perceptron largely misclassifies the given training set into a reduced number of classes, then this perceptron is substituted with a new perceptron..The second novelty consists of the introduction of a new criterion for the removal of tough training patterns that generate the problem of over-fitting. The experimental results show that the proposed BNT leads to satisfactory results in terms of both tree depth reduction and classification accuracy.

Recently, a neural network tree (NNTree) classifier [7] using a multi layer perceptron (MLP) at each node was proposed for designing tree-structured pattern classifiers. To limit the depth of the tree, a new uniformity index was introduced. Such an index accomplishes the intuitive goal of reducing the misclassification rate. The performance of the NNTree has been evaluated in different contexts, such as in letter recognition satellite image classification and splice-junction and protein coding region identification. Experimental comparisons have been proposed with respect to other classifiers. The main drawback is the necessity of an ad hoc definition of some parameters for each context and training set, such as the network architecture (e.g., the number of hidden layers, number of nodes for each layer, etc.) and the uniformity index.

4. Competitive Neural Trees (CNeT)

The CNeT contains **m-ary** nodes and grows during learning by using inheritance to initialize new nodes. An **m-ary** tree is a data structure employed to improve external sorting in which for every node in the tree there are no more than **m** child nodes. Binary trees are a specific implementation of an m-ary tree where there are **m=2** child nodes for every node on the tree.

At the node level, the CNeT employs unsupervised competitive learning. The CNeT performs hierarchical clustering of the feature vectors presented to it as samples, while its growth is controlled by forward pruning. Because of the tree structure, the prototype in the CNeT close to any sample can be determined by searching only a fraction of the tree.

4.1. CNeT Architecture

The CNeT has a structured architecture. Each node contains **m** slots $\boldsymbol{S_1, S_2, .., S_m}$ and a counter age that is incremented each time a sample is presented to that node. The behavior of the node changes as the counter age increases. Each slot $\mathbf{S_i}$ stores a prototype $\boldsymbol{v_i} \in \boldsymbol{V} \subset \boldsymbol{R^n}$, a counter count, and a pointer to a node. The prototypes are updated to represent clusters of samples. The slot counter count is incremented each time the prototype of that slot is updated to match a sample.

4.2 CNeT Learning

In the learning phase, the tree grows starting from a single node, the root. The prototypes of each node form a minuscule competitive network. When a sample arrives at a node, all of its prototypes compete to match it. If denotes the distance between and , the prototype is the winner if . The distance measure used in this paper is the squared Euclidean norm, defined as

(1)

According to this scheme, the winner is the only prototype that is attracted by the input arriving at the node. The winner is updated according to the equation

(2)

The learning rate decreases exponentially with the ***age*** of a node according to the equation

(3)

where is the initial value of the learning rate and determines how fast decreases. The update (2) moves the winner closer to the sample and, therefore, decreases the distance between the two. After a sequence of sample presentations and updates, each of the prototypes will respond to samples from a particular sub region of the input space.

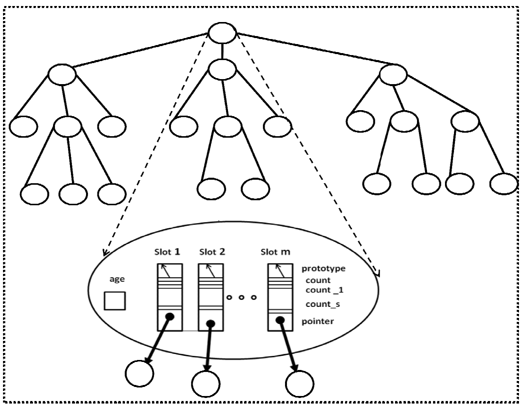

Figure.15. Tree structure of CNeT and node detail

Life Cycle: Each node goes through a life cycle. The life cycle of a node may be partitioned into the following phases.

Creation: The node inherits properties from the parent slot such as the prototype and a fraction of the class counters.

Youth: The prototypes compete to respond to the samples and the winning prototype is updated. The prototypes split the region of the input space that the node sees into sub regions.

Maturity: The prototypes still compete for the samples and they are updated. If a splitting criterion is , then a new child is created and is assigned to a slot.

Frozen: the prototypes compete for the inputs but they are not updated. If the winner has a child-node assigned, then it sends the sample to the child.

Destruction: All children have been destroyed.

Training Procedure: Do while stopping criterion is :

Do while stopping criterion is FALSE:

(1) Select randomly a sample . Let be class that belongs to.

(2) Traverse the tree starting from the root to find a terminal prototype that is close to . Let and be the node and the slot that belongs to, respectively.

(3) If the node is not frozen, then update the prototype according to (2).

(4) If a splitting criterion for the slot is TRUE, then assign a new node as child to and freeze the node .

(5) Increment the counter for class , the counter in slot , and the counter age in node .

4.3 Global ( ) Search Method

The global ( ) search method expands the nodes of the tree level by level, starting at the root. After this is done for all the nodes that are to be expanded at this level of the tree, the prototypes with the smallest distances are selected. If a selected prototype has a child-node assigned, this child-node will be expanded during the next expansion step. Suppose a selected prototype is a terminal prototype. If its distance to the given feature vector is the smallest so far, then the smallest distance is updated and the prototype is the new candidate to be selected for return. When no more prototypes are to be expanded, the global ( ) search method terminates and returns the best terminal prototype seen.

The global ( ) search method searches a subtree that has a width of at most . Hence, the time required by the global ($\boldsymbol{w}$) search method to return a terminal prototype is $\boldsymbol{O}$ ($\boldsymbol{wD_{tree}}$). Clearly, the speed of this search method depends on the choice of the search width. If $\boldsymbol{w} > 1$, then the search by the global ($\boldsymbol{w}$) method takes longer but the probability that the search returns the closest prototype increases. Thus, the selection of w allows the user to balance the tradeoff between the time required for the search and the generalization ability of the CNeT.

5. Experimental Result

In this paper, Myanmar sentences in Grade I Text Book are scanned with Canon Scan Lide 110 model. Training data 100 sentences and testing data 60 sentences are used in Grade I Text Book. The root node of CNeT tree is randomly chosen from the training datasets. The operation of a search method applied to a CNeT and trained to recognize characters using the feature vectors obtained after preprocessing stages. The CNeT was trained and grown using the global search method. Forty nine (33 Myanmar characters, 12 Vowels and 4 Medials) prototypes are selected and CNeT tree is constructed at learning rate $\propto$ is 0.02. Maturity age of the tree is 298.

Recall was performed by interpolating between the two closet prototypes to the input example. The performance of the CNeT trained and grown using the global search method when the search width varied from $w = 8$ to $w = 32$. As w increases, the global method searches a larger subtree and is expected to return more often a terminal prototype that is the closest to the input sample. Recognition accuracy rate 97% is obtained with our testing data 60 sentences. The complexity of our method is quite better than other neural network methods. The performance of CNeT depends on the search method employed in the learning phase.

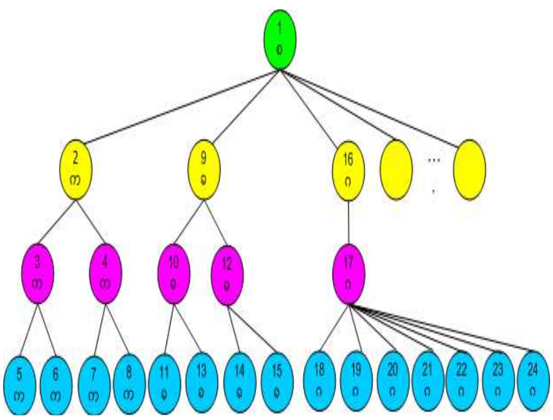

Figure.16. Sample of CNeT Tree for Myanmar Character Recognition

6. Conclusion

Competitive Neural Trees (CNeT) is applied to implement character recognition system. This system includes image acquisition, preprocessing, segmentation, feature extraction and character recognition. In the segmentation stage, vertical segmentation algorithm is proposed. In this experiment, Myanmar sentences in Grade I Text Book are used. The overall character recognition accuracy rate achieved 95% in Grade I Text Book.

REFERENCES

[1] A. Bharath and S. Madhvanath, “Free Pad: a novel handwriting-based text input for pen and touch interfaces”, Proceedings of the 13th international Conference on Intelligent User Interfaces, pp. 297-300, 2008.

[2] Bhardwaj, F. Farooq, H. Cao and V. Govindaraju, “Topic based language models for OCR correction”, Proceedings of the Second Workshop on Analytics For Noisy Unstructured Text Data, pp. 107-112, 2008.

[3] C.Micheloni, A.Rani, S.Kumar and G.Foresti, “ A balanced neural tree for pattern classification”, Neural Networks, vol. 27 , pp- 81–90,2012

[4] H. H. Song and S. W. Lee, “A self-organizing neural tree for large-set classification,” IEEE Trans. Neural Networks, vol. 9, pp. 369–380, May 1998

[5] K.Sandar, “Off-line Myanmar Handwriting Recognition using Hidden Markov Models”, 2005

[6] M.M.Nge, “Automatic Segmentation Of handwritten Myanmar Digits and Scripts on the Voucher Sheet”, 2004

[7] Maji, “Effeicient design of neural network tree using a single splitting criterion”, Nero computing, vol.71, pp-787-800, 2008

[8] S.S.S.Yee, Y.Thein, “High Accuracy Myanmar Handwritten Character Recognition using Hybrid approach through MICR and Neural Network”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 6, November 2010