Active Handwritten Word Recognition

Jaehwa Park and Venu Govindaraju
Center of Excellence for Document Analysis and Recognition
Department of Computer Science and Engineering
State University of New York at Buffalo
Amherst, NY 14260, USA
E-mail: jaehwap,govind@cedar.buffalo.edu

An active word recognition paradigm using recursive recognition processing is proposed. To achieve successful recognition result with minimum required processing effort, recursive system architecture which has active combination of a recognition engine and a decision making module is introduced. In the proposed model, a closed loop connection between recognizer and decision maker operates recursively with successive upgrades of recognition accuracy. The recursion can eventually reach a satisfactory terminal condition or a rejection state of exhaustive use of all the resources. The proposed model is implemented in a segmentation based lexicon driven word recognition application and experiments show enhanced recognition results.

1 Introduction

One of the most common architectures of word recognition system is the serially connected processing steps of recognition and decision making. Several choices of raw recognition results are made available by a word recognizer and a decision maker chooses a recognition result to represent as the true answer.

In this architecture, since the flow of process is monotonous, the word recognition engine is usually operated in several rigid process flows which have been built optimally from training data. A decision maker categorizes the best recognition output in order to meet the performance requirements that are imposed by the user of the recognition system. The process is usually tuned to maximize cost-performance globally in a testing set.

However, usually the most cost-effective solution is hard to obtain with such a rigid recognition paradigm. In handwriting recognition, good homogeneity between real and testing data is not guaranteed in most cases. Thus the classification accuracy usually depends on image quality and decision environment.

To maximize recognition performance in a serial recognition engine, more features are usually added to improve classification accuracy. Features extracted at the finer resolution have the advantage of less confusion with local shapes. And multi-resolution features have been found to be more accurate than features at a single scale[1 2].

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 403-412

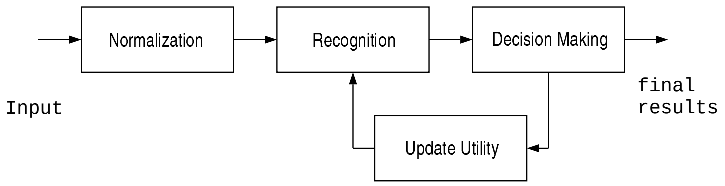

Figure 1: A practical model of recursive recognition system

However, features in higher resolution are not always necessary to reach final acceptance even though such features add more separation between hypotheses. Sometimes additional features at the detailed level are unnecessary for the required classification depth and excessive features generate dimensionality problems. Moreover, they have disadvantage of increasing computational complexity and unregulated metric problems, because of the burden of additional features needed to accommodate the multi-resolution aspect of the feature space.

In order to achieve successful recognition with minimum required processing, an active combination of recognition and decision making is necessary since the relation between recognition and decision making is similar to the relation between an inquiry and search for the answer.

We propose a new word recognition paradigm using recursive recognition processing to achieve active recognition which adapts its recognition ability to a given individual image. The proposed architecture is implemented in a segmentation based lexicon driven word recognition application and shows improved recognition results.

2 Active Word Recognition

One of the simple ways of reaching a cost optimal solution is to have a system in which the recognition and the decision making are connected in a closed loop which operates recursively with successive upgrades of recognition accuracy. The recursion can eventually reach a satisfactory terminal condition or a state of exhaustive use of all the resources. For practical computational implementation, we propose a recursive recognition architecture as shown in Figure 1.

The forward flow of processing steps is similar to that of a conventional word recognizer. But it is combined with a *loose decision maker* and has a feedback loop. The loose decision maker rejects false hypotheses instead of accepting a qualifying top answer. The recognition engine operates by beginning with a global view of the pattern, and increasing the recognition accuracy by recursion depth. A reinforcement step (*Update Utility* in Figure 1) is added as a feedback path from decision making module to the recognizer. During the feedback step, the recognition results are analyzed in term of the decision metric by the decision maker and the reinforcement to the recognizer is updated by the selected (filtered) hypotheses. This feedback helps the recognizer select the most efficient operation to reach terminating conditions.

In[?], a hierarchical character recognizer is formulated using statistical tools, which preserves the benefits of a multi-resolution model while circumventing the disadvantages of a rigid paradigm. However, for a word recognition application, the statistical model of characters in a word does not usually correspond to the probability of general training characters, since words have their own contextual composition probability among characters such as *digrams* and *trigrams*.

Thus, we extend the hierarchical classification to a nearest neighbor method using a distance metric for an active word recognition engine. For computational efficiency, templates obtained from training data are clustered and the primary metric is changed to a minimum distance between templates and extracted feature vector instead of probability.

2.1 Segmentation Driven Word Recognition

Consider a lexicon hypothesis, $\mathbf{h}$ over a finite alphabet, of length N_h, given as

$$\mathbf{h} = [h_0, h_1, \cdots, h_{N_h - 1}] \tag{1}$$

where the C is mutually exclusive classes of size N_c,

$$C = \{c_1, c_2, \cdots, c_{N_c}\} \tag{2}$$

and $h \subset C$.

Let us assume that a stroke segmentation algorithm is applied to a given image and a stroke segment array $\mathbf{s}$ of length N_s is generated as

$$\mathbf{s} = [s_0, s_1, \cdots, s_{N_s - 1}] \tag{3}$$

Also assume that a dynamic programming based word recognizer similar to[4] finds the best matching path of a lexicon entry within a segment array conforming to rules such as (i) at least a character matches a stroke segment, (ii) a character can only match within a window, and (iii) lexicon entries with a length smaller than N_s or the maximum allowable number of characters of stroke array are discarded.

The accuracy of individual class subset associated sub-array between stroke segment s_{a_i} and s_{b_i} of $\mathbf{s}$ is given by

$$A^{a_i,b_i}(h) = \underset{\forall c \in h}{min}\, \delta_h(c) \tag{4}$$

where $\delta_h(c_i)$ is a distance measure function of class c_i in a set of prototypes.

Recognition accuracy of a lexicon entry $\mathbf{h}$ is given by a minimum distance matching array of the individual class subset,

$$A(\mathbf{h}) = [A^{a_0,b_0}(h_0), \cdots, A^{a_{N_h-1},b_{N_h-1}}(h_{N_h-1})] \tag{5}$$

where $a_0 = 0$, $b_{N_h-1} = N_s - 1$ and $a_{i+1} - b_i = 1$.

The conventional recognition accuracy (distance) of a lexicon entry is given by the average of character recognition distance in the lexicon entry as,

$$|A(\mathbf{h})| = \frac{1}{N_h} \sum_{i=0}^{N_h-1} A(h_i) \tag{6}$$

3 Simulating Multi-Resolution in A Distance Metric

3.1 Multi-resolution Feature Space

Let $\mathbf{x}$ be an extracted feature vector from a supposed character image.

$$\mathbf{x} = [x_0, x_1, \cdots, x_{N_x-1}] \tag{7}$$

Also, consider a tree structure (Figure 2) where N_D is the maximum depth and N_B is the number of branches per node.

Let $\mathbf{x}(d,l)$ be an extracted feature vector from sub-image $I(d,l)$ from the tree where, $d \in \{0, 1, \cdots, N_D - 1\}$ and $l \in \{0, 1, \cdots, {N_B}^d - 1\}$.

Features extracted in a larger region provide the global level features while those in a smaller sub-region provide the localized features in finer resolution. The multi-resolution of features is thereby achieved, although the actual feature extraction process and the type of the features remains the same.

Let $\bar{\mathbf{x}}_i$ be a template vector of class c_i which has the same dimension N_x and which has been obtained from a training set. The distance $\nu_i(\mathbf{x})$ between a feature vector $\mathbf{x}$ and templates in a class c_i is defined by the minimum distance to any template vector of the class c_i,

$$\nu_i(\mathbf{x}) = min\, \delta_f(\mathbf{x}, \bar{\mathbf{x}}_i) \tag{8}$$

where $\delta_f(\mathbf{x}_i, \mathbf{x}_j)$ is a distance measure function between two feature vectors which satisfies the triangular inequality.

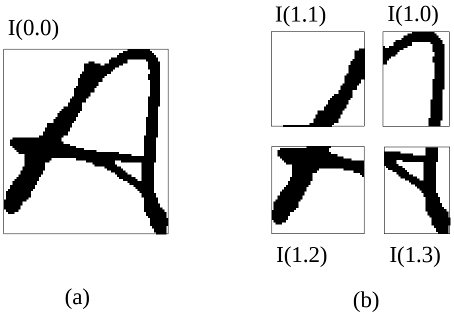

Figure 2: A quad tree structure (a) character image and (b) quad tree division

The distance of class c_i in a layer of a tree, denoted as $\dot{\nu}_i$, is defined by a weighted sum of the local distance measurements of class c_i generated from the sub-images of the layer as,

$$\dot{\nu}_i(d) = \sum_{l=0}^{{N_B}^d-1} \omega_L(d,l)\nu_i(\mathbf{x}(d,l)) \tag{9}$$

where $\nu_i(\mathbf{x}(d,l))$ is the local minimum distance of an extracted feature vector from sub-image $I(d,l)$ to the templates of class c_i, and $\omega_L(d,l)$ is normalized weight of l^{th} sub-image in layer d of a tree, witch satisfies

$$\sum_{l=0}^{{N_B}^d-1} \omega_L(d,l) = 1 \tag{10}$$

Similarly, the distance of class c_i in a tree, denoted as $\ddot{\nu}_i$, is defined by a weighted sum of all the $\dot{\nu}_i$ in the tree as,

$$\ddot{\nu}_i = \sum_{d=0}^{N_D-1} \omega_T(d)\dot{\nu}_i(d) \tag{11}$$

where $\omega_T(d)$ is normalized weight of layer d of the tree, which satisfies

$$\sum_{d=0}^{N_D-1} \omega_T(d) = 1 \tag{12}$$

3.2 Distance in A Recursive Tree

Usually, localized features extracted in finer resolution are useful for enhancing separability of a confusion set in which global features can not provide adequate discrimination. But, empirically, it is realized that localized features are not always useful for better separation, because of the poor homogeneity in patterns of the same class.

Consider a hierarchical image tree and a sub-tree which is a partial tree of the hierarchical image tree. Let us assume that a sub-tree is obtained by successive expansion of a parent image to child images as sub-images by a quad or quin division, and feature vectors extracted from the sub-tree provide the minimum discrimination which satisfy an accepting terminal criterion.

The distance of a class c_i in a sub-tree G, denoted as $\tilde{\nu}_i(G)$, is defined by a weighted sum of local distance of class c_i generated form the sub-nodes which belong to G as,

$$\tilde{\nu}_i(G) = \frac{\sum\limits_{\forall(d,l)\in G} \omega_T(d)\omega_L(d,l)\nu_i(\mathbf{x}(d,l))}{\omega(G)} \tag{13}$$

where

$$\omega(G) = \sum_{\forall(d,l)\in G} \omega_t(d)\omega_L(d,l) \tag{14}$$

Consider a hierarchical feature tree which is extracted from a hierarchical image tree of a given image. Let us assume the feature tree matches perfectly one of the template feature trees of the true class which are built in a classifier, and that there is no identical tree pair in the template trees. Also let us assume that a classifier is caught in a situation that the top two class separability of the distance using global features of the root node does not satisfy the acceptance criterion. Then, the updated distance using sub-nodes will remain zero for the true class but the updated distance to the other classes will be increased, since the generated local distance of sub-nodes will be added up to the weight of the sub-nodes for the non-true classes. In an ideal case, as more nodes are used, the bigger the separability can be obtained.

But in actual applications, a tree which is identical to one of template trees is seldom obtained. Thus, if we assume that an extracted feature tree is close enough to one of template trees of the true class, but relatively not close to those of non-true classes, we can obtain better separation between classes by using sub-node expansion, even though the distance to the true class might increase.

Let a function $g_J(t)$ determine which node is to be examined in the next recursion based on a potential generating function J. $g_J(t)$ is defined returns a sub-node index (d, l) as an output at time t from the pool of expandable nodes. If we choose $g_J(t)$ as a gaze function that finds the most useful node within the pool of *extreme nodes* for updating class distances, the separation will rapidly approach the termination criterion. An extreme node is defined by a child node of examined nodes but unused until t.

If we assume that a sub-tree is recursively expanded by only one node per recursion time, we can rewrite $\tilde{\nu}$ of Equation 13 as a recursive equation,

$$\breve{\nu}_i(t) = \frac{\breve{\omega}(t-1)\breve{\nu}_i(t-1) + \omega_T(d)\omega_L(d,l)\nu_i(\mathbf{x}(d,l))|_{(d,l)=g_J(t)}}{\breve{\omega}(t)} \tag{15}$$

where

$$\breve{\nu}_i(t) = \tilde{\nu}_i(G_t)|_{G_t=G(t)} \quad \text{and} \quad \breve{\omega}(t) = \omega(G_t)|_{G_t=G(t)} \tag{16}$$

$G(t)$ denote a sub-tree expanded until time t and $\breve{\nu}_i(t)$ denotes a recursive distance function of class c_i at time t. Since the sub-node expansion is finite, the recursion index t is only effective in the range of

$$t \in \{0, 1, \cdots, \sum_{d=0}^{N_D-1} N_B^d - 1\} \tag{17}$$

The recursive equations of the weight function and the recursive sub-tree are

$$\breve{\omega}(t) = \breve{\omega}(t-1) + \omega_T(d)\omega_L(d,l)|_{(d,l)=g_J(t)} \tag{18}$$

and

$$G(t) = G(t-1) \uplus I(d,l)|_{(d,l)=g_J(t)} \tag{19}$$

where $\uplus$ denotes the grafting operation of a sub-node I to a tree.

We define a potential measurement of an extreme node, denoted as $J(d, l)$, by weighted separability between the global top two choices of the parent node as,

$$J(d,l) = \omega(d)\omega(d,l)(\nu_{second}(\mathbf{x}(d_p, l_p)) - \nu_{top}(\mathbf{x}(d_p, l_p))) \tag{20}$$

where c_{top} and c_{second} are the global top two choices at the time t, and (d_p, l_p) is the index of parent node of (d, l). After finding all potential measurements of extreme nodes, an extreme node which has the largest potential measurement is selected as an output of $g_J(t)$ to be examined in the next recursion. If global top two choices are changed, potential measurements in all extreme nodes are updated, otherwise they remain the same.

3.3 Word Recognition Accuracy

A time dependent accuracy support of a lexicon entry **h** is defined by an array of minimum distance of the individual class subset at time t as,

$$A(\mathbf{h}, t) = [A(h_0, t), A(h_1, t), \cdots, A(h_{N_h - 1}, t)] \tag{21}$$

where

$$A(h, t) = \min_{\forall c_i \in h} \breve{\nu}_i(t) \tag{22}$$

The conventional recognition distance of a lexicon entry is given by the average of character recognition distance in the lexicon entry as,

$$|A(\mathbf{h}, t)| = \frac{1}{N_h} \sum_{i=0}^{N_h - 1} A(h_i, t) \tag{23}$$

From the above equations, recursion depth t limits the usage of a hierarchical feature tree for individual character recognition. Since a recursive sub-tree is a partial tree of a hierarchical feature tree which starts from the root and expands to sub nodes successively, at $t = 0$, classification results are obtained in the most global view (coarse resolution). By increasing recursion depth, more built-in feature templates are used by comparing to the extracted features in finer resolution. From the recursive equation of 15, an active classification which has multi-resolution and minimal usage of resources can be simulated.

In segmentation driven word recognition in which character matching is limited on a segmented stroke frame, each independent sub-tree is built based on the image generated from each possible combination of stroke segments for character matching of a lexicon entry. But all characters which belong to any of the character subsets of all lexicon entries associated with the character image boundary of a word image share the same sub-tree.

4 Experiment

The proposed recursive word recognition scenario is implemented in a lexicon driven word recognizer for experiments. For a character recognition engine, a hierarchical character recognizer which has a quad tree structure of depth of 4 has been implemented. Thus, 85 (1+4+16+64) sub-nodes have been built. From setting of $N_f = 2$ and $N_g = 8$, total 34 feature measurements are extracted for a feature vector. Using manually isolated character images from word images which are exclusive to testing word images, a maximum of 30 templates per class have been generated for each sub-node by a $K -$

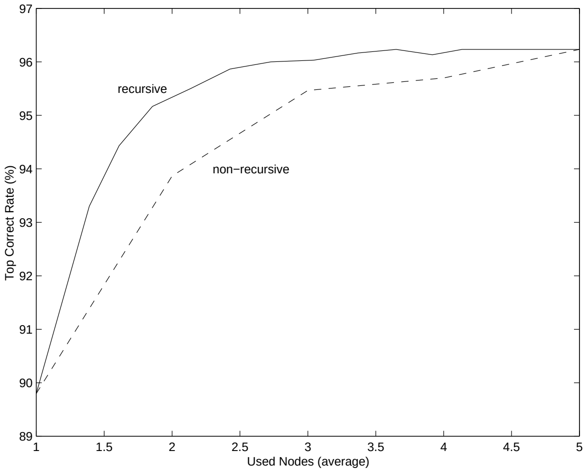

Figure 3: Dynamic recognition characteristics - comparison between recursive system with active searching and non-recursive system with passive adding of sub-nodes

means clustering algorithm with a mean square error threshold of 0.1. A stroke classifier has been used before the word matching step to generate a prime stroke sequence. A dynamic programming technique is used to find the best matching path on the prime stroke array.

3000 word images (digitized at 212 dot-per-inch) including city names, state names, and street names collected from address block of mail pieces in USPS mail stream were used as testing images. The lexicons were generated as subsets which has randomly selected lexicon entries in fixed size (10, 100 or 1000) from the pool of all the possible city, state, and street names. The true word was always present in the lexicon. The word recognizer achieved a maximum of 96.3% of top choice correct rate with size 10 lexicons.

Figure 3 shows word recognition performance. The top choice correct rate is changed by adjusting the usage of nodes. The performance with feedback loop of active recursion controlled by decision maker which reacts to the image quality is superior than that of non-recursive methods with fixed usage of resources. With same degree of usage nodes, the recursive system achieves better top choice correct rate, even though the both steady state recognition performance are the same.

Figure 4 shows the dynamic character recognition characteristics when the acceptance degree of decision making is changed. The liability mean a weighed term considering the similarity of lexicon entries. The x-axis displays the

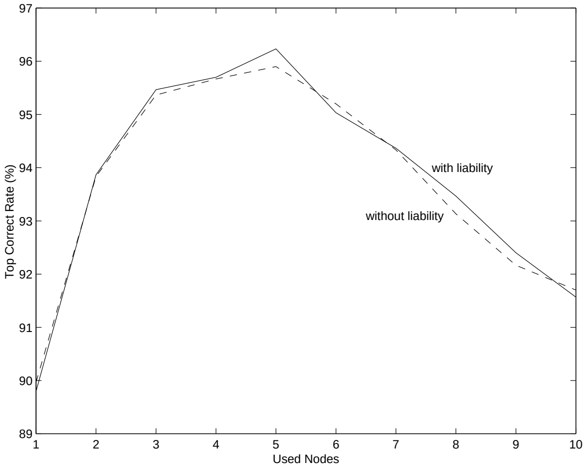

Figure 4: Dynamic recognition characteristics - recognition performance changes by various acceptance criterion

average usage of nodes. If the accepting terminal condition is more restrictive, the usage of features is increased because more discrimination power between classes is necessary. When liability of character confidence within lexicon set is used, the performance can be maximized.

5 Conclusion

In this paper an active recognition scenario has been introduced. An active combination of recognition and decision making has been implemented in a recursive processing architecture. The proposed architecture adapts its recognition accuracy to individual input image through recursion. Experiments show the validity of the proposed model compared to a conventional static method.

1. O.D. Trier, A.K. Jain and R.Taxt, Pattern Recognition **29**, 641 (1996)
2. J.T. Favata and G. Srikantan, Internationa Journal of Imaging Systems and Technology **7**, 304 (1996)
3. J. Park, V. Govindaraju and S.N. Srihari, IEEE Trans. on Pattern Analysis and Machine Intelligence **24**, 400 (2000)
4. G. Kim and V. Govindaraju, IEEE Trans. on Pattern Analysis and Machine Intelligence **19**, 366 (1997)