Handwritten Numerals Recognition Using an Ant-Miner Algorithm

Pisit Phokharatkul[1], and Supachai Phaiboon[2]

[1]Department of Computer Engineering, [2]Department of Electrical Engineering, Faculty of Engineering, Mahidol University, Phutthamonthon 4 Road, Salaya, Nakhon Pathom , 73170 , Thailand.
Email : egpph@mahidol.ac.th, egspb@mahidol.ac.th

Abstract: This paper presents a system of handwritten numerals recognition, which is based on Ant-miner algorithm (data mining based on Ant colony optimization). At the beginning, three distinct fractures (also called attributes) of each numeral are extracted. The attributes are *Loop zones*, *End points*, and *Feature codes*. After these data are extracted, the attributes are in the form of attribute = value (eg. *End point10* = true). The extraction is started by dividing the numeral into 12 zones. The numbers 1-12 are referenced for each zone. The possible values of *Loop zone* attribute in each zone are "true" and "false". The meaning of "true" is that the zone contains the loop of the numeral. The *Endpoint attribute* being "true" means that this zone contains the end point of the numeral. There are 24 attributes now. The *Feature code attribute* tells us how many lines of a numeral are passed by the referenced line. There are 7 referenced lines used in this experiment. The total attributes are 31. All attributes are used for construction of the classification rules by the Ant-miner algorithm in order to classify 10 numerals. The Ant-miner algorithm is adapted with a little change in this experiment for a better recognition rate. The results showed the system can recognize all of the training set (a thousand items of data from 50 people). When the unseen data is tested from 10 people, the recognition rate is 98 %.

Keywords: Handwritten Numerals Recognition, Ant-Miner Algorithm**.**

1. INTRODUCTION

There are many researches for transforming handwritten characters or handwritten numerals to texts. The researchers have attempted to propose several methods implemented in computers and then apply them in various areas. The details of these approaches are illustrated in [1-3].

The goal of "handwritten numerals recognition" is to make a computer understand and specify what numerals a human wrote. If the problem can be solved, we can use it as a solution method for many different types of information such as transformation of the output into text file for application with other works. Examples of handwritten numerals are illustrated in figure 1.

This research proposes the off-line recognition method of handwritten numerals by description of the three features in each numeral image (Loop, Endpoint and Feature code). Ant-Miner algorithms [4-6] are adapted with small changes for recognizing the handwritten numeral images with a high accuracy recognition rate.

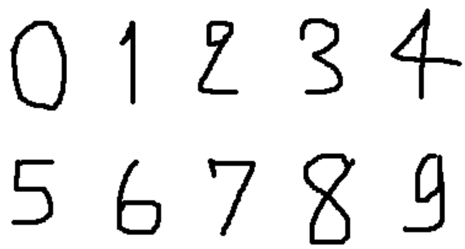

Fig 1. Example of handwritten numeral image.

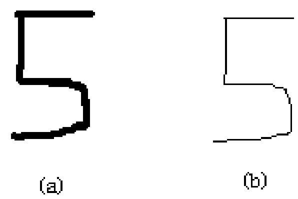

Fig. 2 (a) Normal numeral image.
(b) Skeleton numeral image.

2. PRE-PROCESSING

2.1 Thinning method

The width of the numeral image is normally larger than 1 pixel, but in the feature extraction process, only a numeral skeleton is used. The numeral skeleton is the numeral image that has only 1 pixel in width. See figure 2 for an example. In this step, the algorithm in the reference [2] is used to convert a normal numeral image into a skeleton image.

2.2 Zones of reference

Each numeral will be normalized to 128x128 pixels and divided into 12 zones with the same width and height. The names given to each zone are $z_1, z_2, \ldots, z_{12}$ as shown in figure 3.

z1	z2	z3
z4	z5	z6
z7	z8	z9
z10	z11	z12

Fig. 3 Twelve zone division.

3. THREE FEATURES OF NUMERAL WITH A PAIR OF ATTRIBUTES AND THEIR VALUES

3.1 Loop of numeral

The loop is one of the distinctive features in numerals. It is defined as a circle or a closed loop in a numeral. Some numerals have only one loop; some numerals have more than one, or even have none. It depends on personal habits of writing.

Normally, different numerals have different loop zones. Thus, the attribute ***loop_Zone_Z_n*** is defined by whether the zone z_n has a loop or not. The possible values of this attribute are ***True*** or ***False*** (have or do not have). See figure 4 for an example of looping zone of '2' numeral.

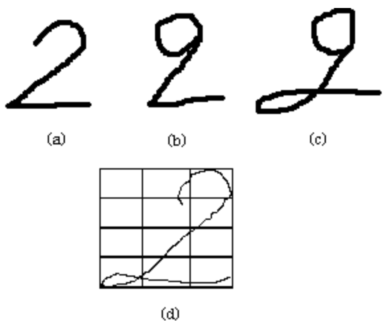

Fig. 4 The numeral of '2' with (a) no loop, (b) one loop, and (c) two loops. (d) An example of the attribute when ***Loop_Zone_Z_{10} = True*** and other zones is False.

3.2 End point

The end point is the point that has only one point connected to it. The attribute *End_point_z_n* is used to define it. A possible value of this attribute is *True* or *False* whether the zone z_n has an end point or not. For an example, see figure 5.

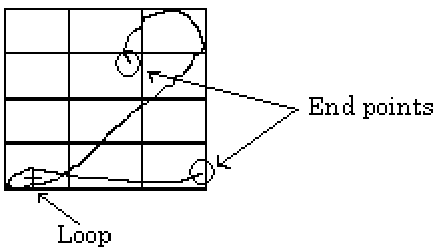

Figure 5 The position of the end point that can represent the attribute by ***End_point_z_5*** **= True**, ***End_point_z_{12}*** **= True**, and other zones are ***false***.

3.3 Feature code

The feature is defined by the maximum number of points that the referent lines pass in its zone. ***Feature_Code_Z_n*** is used to represent an attribute for this feature. Zone Z_1, Z_2, Z_3 and Z_4 use a horizontal referent line and zone Z_5, Z_6 and Z_7 use a vertical referent line. See figure 6 for an example.

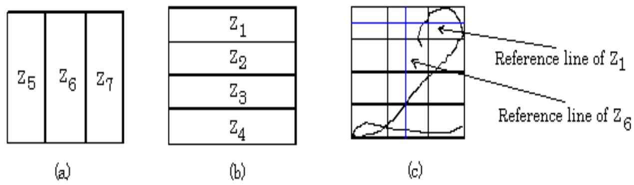

Figure 6 (a) The zones that use vertical referent lines.
(b) The zones that use horizontal referent lines.
(c) Two referent lines pass Zones Z_1 and Z_6 that makes attributes *Feature_Code_Z_1* = 1 and *Feature_Code_Z_6* = 5.

4. ANT-MINER ALGORITHM

Ant-Miner algorithm [4] has been proposed to discover a set of IF-THEN rules from data in the form of **IF < Term1 AND Term2 AND… > THEN < Class >** in the task of data mining. Each term in the rule of an antecedent part is a triple attribute, an operator and a value. The value is the possible value in the domain of each attribute. There is only an operator "=" used in this task such as < Month = January >. The consequent part specifies the class prediction only in the case that predicted attributes satisfy all terms in the antecedent part. The set of rules, which is constructed by this algorithm cover all or almost all the training cases, has a small number of terms and a small number of rules that are good for data mining. The high level of its algorithm is shown in figure 7.

Training set = all training cases;
Rule list = empty;
REPEAT
    i=0;
    Pheromone Initialization;
    REPEAT
        i=*i*+1;
        Ant_i constructs a classification rule;
        Prune the current constructed rule;
        Update the pheromone of the trail followed by Ant_i;
        The best rule is memorized;
    UNTIL (*i* = *No_of_Ants*) or (Ant_i constructed the same rule as the previous Ants continually *No_Rule_Converg* times)
    The best rule is added to the rule list;
    Remove the cases covered by the selected rule from the training set;
UNTIL (Number of cases in the Training set < *Max_uncovered_cases*)

Figure 7 Overview of an Ant-Miner algorithm.

Based on the algorithm, after the pheromone is initialized, many rules are constructed in the inner Repeat-Until loop with the rule pruning and the pheromone updating method. The loop will stop when ants construct the same rule continually more than *No_Rule_Converg* times or the number of rules is equal to the number of ants. When the inner Repeat-Until loop is completed, the best rule will be added to the rule list. Then, all training cases which are predicted by this rule were removed from the training case set. The pheromone is initialized again. This cycle is controlled by the Outer Repeat-Until loop. The Repeat-Until loop will finish when the number of uncovered training cases are less than a threshold, called *Max_uncovered_cases*.

5. TRAINING STEP

In this experiment, the Ant-Miner algorithm is used for training the recognition system (to construct rule list) by utilizing data from three features of a numeral that is described in section 3.

In original version of Ant-Miner [4], the quality of the rule is computed by the equation

$$Q(Rule) = \frac{TP}{TP+FN} \times \frac{TN}{FP+TN} \quad (1)$$

where:

• *TruePos* (TP) is the number of cases covered by the rule and having the same class as that predicted by the rule.

• *FalsePos* (FP) is the number of cases covered by the rule and having a different class from that predicted by the rule.

• *FalseNeg* (FN) is the number of cases that are not covered by the rule, while having the class predicted by the rule.

• *TrueNeg* (TN) is the number of cases that are not covered by the rule which have a different class from the class predicted by the rule.

With the equation, the rules from the algorithm are shortening. It has a minimum number of rules in the rules list with a high accuracy. (Cover all or almost all in the training set). But in this experiment, recognition of all training data is needed. In order to weight Q (Rule), the term ($\frac{1}{FP+1}$) is added. Then the equation of Q (Rule) is changed.

$$Q(Rule) = \left(\frac{TP}{TP+FN} \times \frac{TN}{FP+TN}\right) \times \left(\frac{1}{FP+1}\right) \quad (2)$$

By this Q (Rule), the value of FP is converged to zero (accurate rate is converged to 100%) but number of rules will more than the original. The other parameters are

No_of_ants = 1,500
No_Rule_Converg = 10
Max_uncovered_cases = 0

6. THE EXPERIMENTAL RESULTS

In the training step, the data of 2000 samples from 50 people are classified into 10 classes (0, 1, 2, …, 9), which has 200 items of data in each class and 31 attributes in each data item. The data are input in the Ant-Miner algorithm for learning in order to classify handwritten numerals by construction of a set of rules. Then, the same rule list is used to predict 100 unseen data items of handwritten numerals from 10 persons. The results of the proposed recognition system are shown in Table.1

7. CONCLUSION

This paper presents a system of recognition of handwritten numerals based on the Ant-Miner algorithm. The result shows that the system can recognize all of the training set. From the research, there are many rule lists created by the result. The recognition rate of the best rule list from various styles of unseen data from 10 persons is 98 %. So, the best rule list (Highest recognition rate and Minimum number of rules) is used for recognition-engine construction for numerals. In this experiment the 1st test is selected.

In the future, we plan on adding handwritten Thai characters and Thai numerals into the recognition system. The handwritten character recognition system is a wide area of application, such as OCR system, handwritten input to computers, handwritten reader for blinds, identifies handwritten signatures and handwritten reader for robots or other intelligent systems.

Table 1. Result of handwritten numeral testing with an Ant-miner classifier.

No. of test	Number of rules	No. of all Terms	Terms/Rule	Recognition rate (%)
1	338	886	2.62	98
2	324	895	2.76	97
3	334	918	2.75	97
4	376	981	2.61	98
5	342	906	2.65	96
6	377	934	2.48	96
7	334	921	2.76	97
8	360	950	2.64	97

REFERENCES

[1] P. Phokharatkul, and C. Kimpan, "Handwritten Thai Character Recognition Using Fourier Descriptors and Genetic Neural Networks", Computational Intelligence An International Journal, Vol.18, No. 3, Aug. 2002, pp. 270-293.

[2] P. Choruengwiwat, Thai handwritten character recognition using extraction of distinctive features. Master Thesis, Department of electrical engineering Chulalongkorn University, 1998.

[3] T. Thongkamwitoon, W. Asdornwised, S. Aramvith, S. Jitapunkul, "On-line Thai-English handwritten character recognition using distinctive features." Circuits and Systems, 2002. APCCAS '02. 2002 Asia-Pacific Conference on, vol. 2, 2002, pp. 259 – 264.

[4] R.S. Parpinelli, H.S. Lopes, A.A. Freitas, "Data mining with an ant colony optimization algorithm." IEEE Transactions on Evolutionary Computation, 2002, pp.321 - 332.

[5] L. Bo, H.A. Abbas, B. McKay, "Classification rule discovery with ant colony optimization." IEEE/WIC International Conference, 2003, pp. 83 – 88.

[6] M. Dorigo, S. Thomas, Ant Colony Optimization", A Bradford Book The MIT Press, 2004.