VNU Journal of Science, Mathematics - Physics 26 (2010) 123-139

Isolated Handwritten Vietnamese Character Recognition with Feature Extraction and Classifier Combination

Le Anh Cuong[*], Ngo Tien Dat, Nguyen Viet Ha

University of Engineering and Technology, VNU, E3-144 Xuan Thuy, Cau Giay, Hanoi, Vietnam

Received 5 July 2010

Abstract. Handwritten text recognition is a difficult problem in the field of pattern recognition. This paper focuses on two aspects of the work on recognizing isolated handwritten Vietnamese characters, including feature extraction and classifier combination. For the first task, based on the work in [1] we will present how to extract features for Vietnamese characters based on gradient, structural, and concavity characteristics of optical character images. For the second task, we first develop a general framework of classifier combination under the context of optical character recognition. Some combination rules are then derived, based on the Naive Bayesian inference and the Ordered Weighted Aggregating (OWA) operators. The experiments for all the proposed models are conducted on the 6194 patterns of handwritten character images. Experimental results will show the effective approach (with the error rate is about 4%) for recognizing isolated handwritten Vietnamese characters.

Keywords: artificial intelligence; optical character recognition; classifier combination.

1. Introduction

The problem handwriting recognition receives input as intelligible handwritten sources such as paper documents, photographs, touch-screens and other devices, and try to output as correct as possible the text corresponding to the sources. The image of the written text may be sensed off-line from a piece of paper by optical scanning, so actually it lies in the field of optical character recognition. Alternatively, the movements of the pen tip may be sensed on-line, for example by a pen-based computer screen surface. Off-line handwriting recognition is generally observed to be harder than online handwriting recognition. In the online case, features can be extracted from both the pen trajectory and the resulting image, whereas in the off-line case only the image is available. Firstly, only the recognition of isolated handwritten characters was investigated [2], but later whole words [3] were addressed. Most of the systems reported in the literature until today consider constrained recognition problems based on small vocabularies from specific domains, e.g., the recognition of handwritten check amounts [4] or postal addresses [5]. Free handwriting recognition, without domain-specific constraints and large vocabularies, was addressed later in a some papers such as in [6, 7]. The recognition rate of such systems is still low, and there is a need to improve it. There are a few related

[*] Corresponding author. Tel.: 84-902134662
E-mail: cuongla@vnu.edu.vn

studies for Vietnamese, such as [8] for recognizing online characters and [9] for recognizing off-line characters. As one of the beginning studies of handwritten recognition for Vietnamese, in this paper we just consider the off-line recognition and focus on the isolated character recognition.

There are two important factors which most affect the quality of a recognition/classification system (among the methods which follow machine learning approaches). They include the features extracted from the data (i.e. which kinds of features will be selected and how to extract them) and the machine learning algorithms to be used. In our opinion, feature extraction plays the most important role for any systems because it provides knowledge resources for those systems. For the work of text recognition, feature extraction aims to extract useful information from input images and represents the extracted information of a image as a vector of features. Because Vietnamese has a diacritic system that forms much similar character groups, so discriminating these characters is very difficult. To extract features for the work of recognition we will focus on the main characteristics forming the difference between them. Encouraging by the studies in [10, 1] we will use the three kinds of features including gradient, structural, and concavity features. This approach is suitable for applying to images which have different sizes, and it has been shown effective for Arabic as presented in [10, 1]. In this paper we will present in detail how to extract these features for Vietnamese characters. Different from previous studies these kinds of features will be investigated as separate feature sets as well as the combination set. In addition, these feature sets will be also used in various machine learning models including classifier combination strategies/rules. All these works aim to find out the appropriate model for Vietnamese character recognition.

In addition, for the aspect of classifier combination, different from previous studies, this paper applies some combination rules which are new to the problem of character recognition. It is also important that these rules are interpreted in a general framework of classifier combination, and then derived under OWA operators and Naive Bayesian inference. This helps to understand the meaning of the obtained combination rules and to discuss the appropriate obtained results of the recognizing. In addition, we also investigate various kinds of individual classifiers, one is based on different representations of features and the other based on different machine learning algorithms. Note that beside combination rules we also investigate effectiveness of the three machine learning algorithms including neural network (NN), maximum entropy model (MEM), and support vector machines (SVM). Among them support vector machine will be shown as the very effective method for this work.

The rest of the paper is organized as follows. Section 2 presents related works. Section 3 presents Vietnamese characters. Section 4 presents feature extraction for three kinds of them including gradient, structural, and concavity ones. The classifier combination strategies/rules are presented in section 5. All our experiments and discussions are presented in section 6. Finally, we summarize the obtained results in section 7.

2. Related Works

As well known, the researches in OCR can be divided into two approaches: template matching and structure analysis. Template matching is done by taking the total sum of the differences between each sampled value and the corresponding template value, each of which is normalized. The template matching process can be divided into two processes: inputting the shape on template and measuring the degree of coincidence between the input shape and the generated template. Concurrently, in [11] the authors firstly design a feature vector for representing boundary point distances from the center of gravity of the characters and then derive models (i.e. templates) for each character. The classification is performed using the Euclidean distance between the test characters and the generated models.

Currently most studies consider the task of optical character recognition as a classification problem. They firstly represent the image of each character/word as a set of features, and then use a machine learning method to train a classifier. Therefore, the problems here in this approach is how to extract useful features and how to design effective machine learning methods. There are some well-known machine learning algorithms have been used, such as neural networks [12, 13], hidden markov model [13, 1], graph-based method [10]. Note that in this paper we will investigate the three machine learning algorithms NN, SVM, and MEM, which have shown their effectiveness for many pattern recognition problems.

As observed in studies of pattern recognition systems, although one could choose one of learning systems available based on the analysis of an experimental assessment of these to hopefully achieve the best performance for the pattern recognition problem at hand, the set of patterns misclassified by them would not necessarily overlap [14]. This means that different classifiers may potentially over complementary information about patterns to be classified. The research domain of multiple classifier systems (MCS) examines how individual classifiers can be combined to obtain a better classification system. As well known there are two main issues in MCS research: the first issue is how the individual classifiers are generated, and the second is how those classifiers are combined. It is straightforward to apply MCS methods to the problem of handwriting recognition. There are many published studies of the such approach. For example, in [15] a multilayer perceptron is trained to classify handwritten numerals. For each pair of numerals that is likely to be confused a support vector machine is trained. If the two best classes output by the multilayer perceptron are the digits of such a pair, the specialized classifier is invoked. Otherwise the best class of the multilayer perceptron is output. In [16] a rule-based classifier is serially combined with four Hopfield neural networks, which are only used when the first classifier rejects the input. The task of the first neural network is to select one of the other three as the final classifier. This system was also used for recognition of handwritten digits. For handwritten word recognition, using Hidden Markov Models have become a standard method. Consequently, combinations of HMMs with other classifiers have been proposed. In [17] an HMM classifier is combined with a pattern matching classifier. A holistic classifier to reduce the vocabulary for an HMM classifier is described in [18]. More recently, a number of studies are interested in apply techniques in classifier combination strategies for the problem of handwritten recognition, such as [19-22].

It is worth to emphasize that in this paper we will follow a different approach of MCS for handwriting recognition. We will first formulate a general framework of classifier combination and then derive some combination rules based on OWA operators and Naive Bayesian inference, which help to understand the underlying meaning of the used combination rules. These rules are then applied to the Vietnamese handwriting recognition using the individual classifiers which are built from the most effective algorithms (including SVM, NN, and MEM).

3. Vietnamese characters

The Vietnamese alphabet is the current writing system for the Vietnamese language. It is based on the Latin alphabet with some digraphs and the addition of nice accent marks or diacritics - four of them to create additional sounds, and the other five to indicate the tone of each word. The many diacritics, often two on the same letter, make written Vietnamese easily recognizable. The Vietnamese alphabet consists of 29 letters, including:

- The 26 letters of the English alphabet except f, j, w, and z.

- Seven modified letters using diacritics đ, ă, â, ô, ơ, ư

In addition, Vietnamese is a tonal language with 6 tones. These tones can be marked on the letters a ă â e ê i o ô ơ u ư y as follows:

1. level (không dấu) a ă â e ê i o ô ơ u ư y
2. high rising (dấu sắc) á ắ ấ é ế í ó ố ớ ú ứ ý
3. low/falling (dấu huyền) à ằ ầ è ề ì ò ồ ờ ù ừ ỳ
4. dipping-rising (dấu hỏi) ả ẳ ẩ ẻ ể ỉ ỏ ổ ở ủ ử ỷ
5. high rising glottalized (dấu ngã) ã ẵ ẫ ẽ ễ ĩ õ ỗ ỡ ũ ữ ỹ
6. low glottalized (dẫu nặng) ạ ặ ậ ẹ ệ ị ọ ộ ợ ụ ự ỵ

Therefore, excepting special symbols such as comma, colon, stop, …, the number of isolated letters in Vietnamese is 93 including 26 *English alphabet letters*, 7 *diacritic letters*, and 12 *letters* (each has 5 *tones*).

In this work we try to recognize these separate letters. Note that to recognizing each letter there are normally two options. In the first option, each letter is treated as a completed unit/pattern to be recognized. In the second one, the complex letters which contain diacritic or tones or both of them will be recognized in a different way. Each letter is firstly divided into three separate parts: the tone part, the diacritic part, and the remain is a normal letter. Each part is recognized separately and then their results are combined to obtain the whole letter. However, for handwritten letter, it is so difficult to separate these parts. That is because they are written in a flexible style and usually stick together. Therefore, in this work, we choose the first option, and we will extract features for that it can express characteristics of all the three parts. It is also worth to note that our approach in this paper is far different from previous studies on Vietnamese handwritten recognition [9, 23] on both aspects of feature extraction and classifier combination.

4. Feature extraction

4.1. Gradient feature extraction

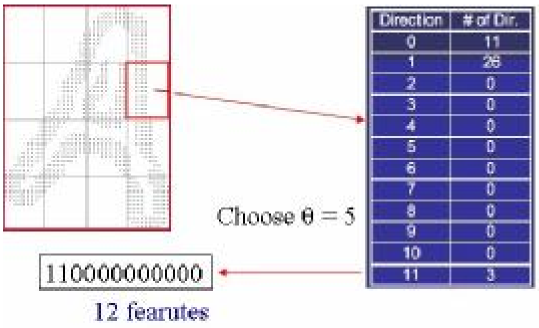
The method of gradient feature extraction will detect features which stand for direction of boundary pixel of the image. These extracted features are represented by a gradient vector. This vector expresses the velocity when changing image pixel value to directions x and y (i.e. vertical axis and horizontal axis).

With an image which its pixels are expressed by a grey value function $f(x; y)$, the gradient vector at pixel $(x; y)$ is computed as in the following formula:

$$\nabla f = [Gx, Gy] = [\partial f / x, \partial f / \partial y] \tag{1}$$

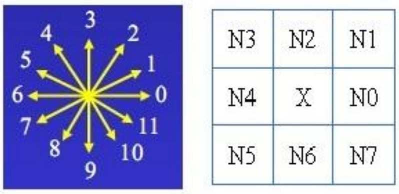
Magnitude of the gradient vector at pixel $(x; y)$ is computed as:

$$\text{Magnitude}(\nabla f(x, y)) = [G^2x + G^2y]^{1/2} \tag{2}$$



Direction	# of Dir.
0	11
1	26
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	3

Fig. 1. An Example of gradient feature extraction.



N3	N2	N1
N4	X	N0
N5	N6	N7

Fig. 2. Directions and Neighbors of a pixel X.

Direction of the gradient vector at pixel (*x*, *y*) is computed as:

$$\alpha(x; y) = \tan^{-1}\frac{Gy}{Gx} \tag{3}$$

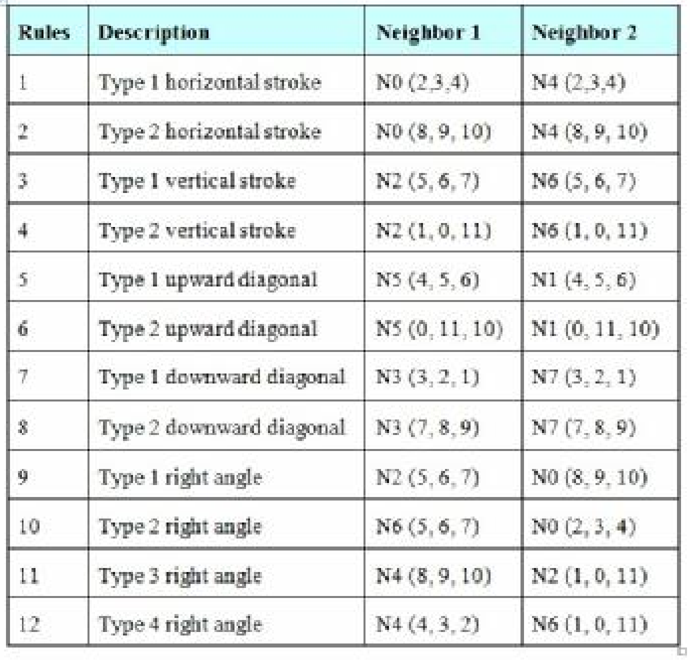
The core of this method is finding out directional frequencies of image boundary pixel. We can describe this algorithm by four steps as: (1) Finding a bounding box around the image to eliminate white space containing no information and decrease computing cost. (2) Computing gradient map about direction and magnitude. The gradient maps are computed by firstly convolving two Sobel operators on the bounded image. These operators approximate derivatives at each pixel to x and y directions. And then compute the direction and magnitude of the gradient vector at each pixel. (3) We only take care about pixels lying on the image boundary where magnitude of gradient vector exceeds a threshold. If $r(i; j) >$ then pixel at $(i; j)$ is a boundary pixel. (4) We design that the direction of the gradient ranges from 0 to 2 ∏ radian. For each boundary pixel, this range is divided into twelve radian parts. Divide the bounded image into 4x4 equal parts. In each part, a histogram is calculated in each direction at every pixel. One threshold is set to the histogram and feature is on if corresponding counting number is greater than the threshold.

Finally, we collect 192 gradient features. For example, the figure 1 illustrates an example when finding 12 gradient features in 8-th part of a image of the character ‘A’.

4.2. Structural feature extraction

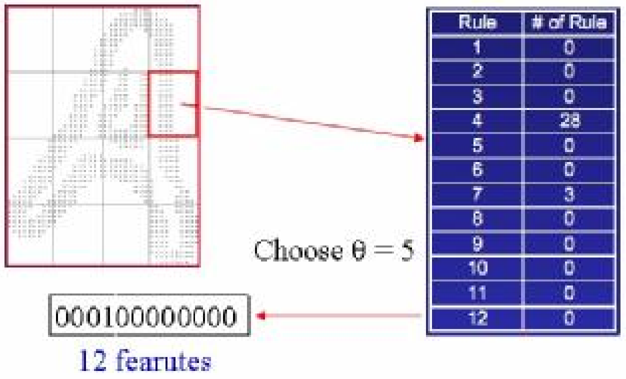
Structural features bases on the gradient map to find out short stroke types in the image. As presented in [1], there are 12 structural feature extraction rules (see Figure 3). These rules manipulate on the eight neighbors of each image pixel. Each rule examines a particular pattern of the neighboring pixels for allowed gradient ranges. There rules correspond to the directions: horizontal stroke, vertical stroke, upward diagonal, downward diagonal, and right angle. Figure 2 shows how to determine and index the directions and neighbors of a pixel.

And now to determine structural features, we divide the bounded image into 4x4 equal parts. In each part, we consider the 12 rules in turn and count the number of pixels satisfying this rule. Assuming we get 12 value a_i where $i = 0 \ldots 11$ and a_i is number of pixel satisfying i^{th} rule. Choosing a threshold θ, we set the feature corresponding to a_i to be true if $a_i > \theta$ and *false* in otherwise. Each part in 4x4 size grid brings us 12 features, so total number of features on the whole image is 192. For example, Figure illustrates the result when finding 12 structural features in 8th part of the image.



Rules	Description	Neighbor 1	Neighbor 2
1	Type 1 horizontal stroke	N0 (2,3,4)	N4 (2,3,4)
2	Type 2 horizontal stroke	N0 (8, 9, 10)	N4 (8, 9, 10)
3	Type 1 vertical stroke	N2 (5, 6, 7)	N6 (5, 6, 7)
4	Type 2 vertical stroke	N2 (1, 0, 11)	N6 (1, 0, 11)
5	Type 1 upward diagonal	N5 (4, 5, 6)	N1 (4, 5, 6)
6	Type 2 upward diagonal	N5 (0, 11, 10)	N1 (0, 11, 10)
7	Type 1 downward diagonal	N3 (3, 2, 1)	N7 (3, 2, 1)
8	Type 2 downward diagonal	N3 (7, 8, 9)	N7 (7, 8, 9)
9	Type 1 right angle	N2 (5, 6, 7)	N0 (8, 9, 10)
10	Type 2 right angle	N6 (5, 6, 7)	N0 (2, 3, 4)
11	Type 3 right angle	N4 (8, 9, 10)	N2 (1, 0, 11)
12	Type 4 right angle	N4 (4, 3, 2)	N6 (1, 0, 11)

Fig. 3. Rules for structural feature extraction.



Rule	# of Rula
1	0
2	0
3	0
4	28
5	0
6	0
7	3
8	0
9	0
10	0
11	0
12	0

Fig. 4. An example of extracting structural features.

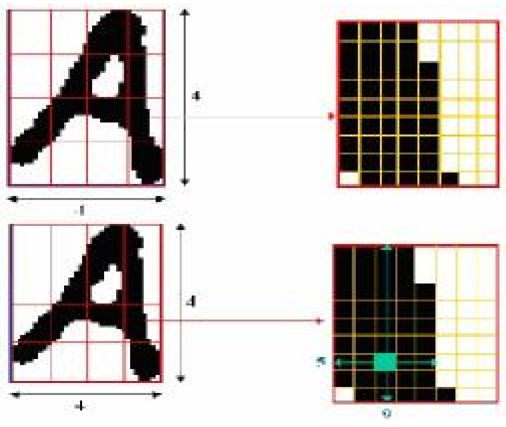


Fig. 5. Coast pixel density and horizontal /vertical stroke features.

4.3. Concavity feature extraction

The concavity features are used to determine the relationship between strokes at a large scale across the image. Concavity features are belonging to 8 feature types: black pixel density, vertical stroke, horizontal stroke, leftward concavity, rightward concavity, upward concavity, and downward concavity. They are computed by placing a 4x4 sampling grid on the image. In each part of 4x4 size grid, we count the number of pixels satisfying one of 8 characteristics above and choose a threshold to determine whether 8 features in that part are on or off. These features consist of: one coarse pixel density features, 2 large stroke features (horizontal and vertical strokes), and 5 concavity or hole features.

For determining **coarse pixel density feature**, we first count the number of black pixel in each part of sampling grid and choose a threshold θ to set this feature true or false. The figure 5 illustrate that the number of black pixels in the 12*th* part of the image is 43 and threshold θ = 20 so this feature is active (i.e. its value is 1).

For determining **horizontal/vertical large stroke features** we first denote c_1 be the length of continuous horizontal black pixels in which the pixel is in, and denote r_1 be the length of continuous vertical black pixels in which the pixel is in. Then, vertical large stroke and horizontal large stroke features are determined based on the relation between c_1 and r_1. For example, in this work, we choose that if $c_1 < r_1 \times 0{:}75$, then it is a vertical large stroke, and if $c_1 > r_1 \times 1{:}5$, then it's a horizontal large stroke. See the figure 5 for an example.

For determining **feature of concavity and hole**, as clearly presented in [1] we design a convolving operator on the image which shoots rays in eight directions and determines what each ray hits. A ray can hit an image pixel or the edge of the image. A table is built to store the termination status of the rays emitted from each white pixel of the image. The class of each pixel is determined by applying rules to the termination status patterns of the pixel. Currently, upward/downward, left/right pointing concavities are detected along with holes. The rules are relaxed to allow nearly enclosed holes (broken holes) to be detected as holes. This gives a bit more robustness to noisy images. These features can overlap in that in certain cases more than one feature can be detected at a pixel location. Figure 6 show some features of concavity and hole.

Finally, we have 128 features in the model of concavity feature extraction.

Note that some studies such as [1] use the combination of these three feature extraction methods. It is useful for detect features ranging from local scale to large scale, from single pixel to multiple pixel relationship. The set of these entire features are called GSC features (i.e. Gradient, Structural, and Concavity). A GSC feature vector consists of 512 features. Note that in this paper, we will investigate each of the three feature sets as well as the GSC feature set separately.

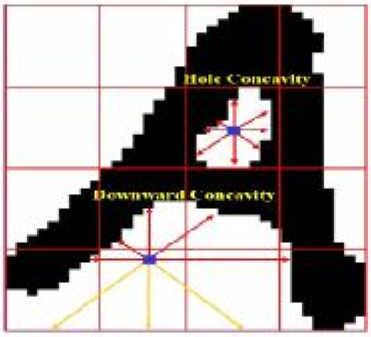


Fig. 6. Feature of concavity and hole.

5. Classifier combination strategies

5.1. Architecture of Multi-Classifier Combination

Classifier combination has been studied intensively in the last decade, and has been shown to be successful in improving performance on diverse applications [14, 19, 22, 24]. The intuition behind classifier combination is that individual classifiers have different strengths and perform well on different subtypes of test data. Fig. 7 intuitively presents the architecture of multiple classifier combination.

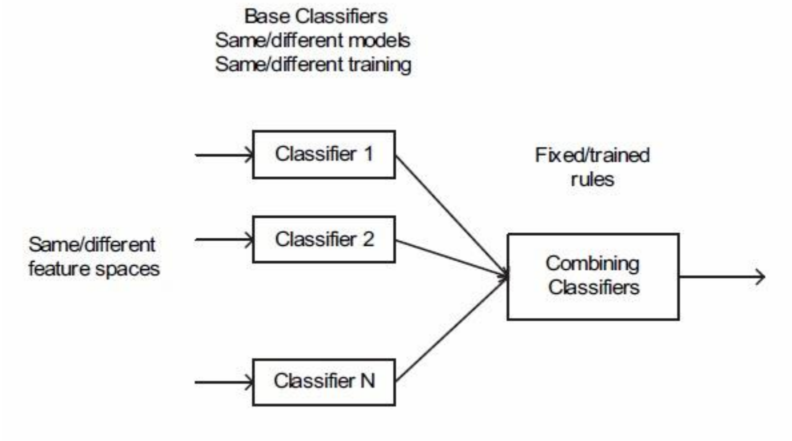


Fig. 7. Architecture of multiple classifier combination.

From this figure, we can see that the base classifiers (also called individual classifiers) can be created based on the different feature spaces, different training datasets, or different models (machine learning algorithms). Note that by combining these different types, we can also create other set of individual classifiers. Fig. 7 also shows the general process of applying classifier combination strategies for a problem such as classification. That is, firstly the set of individual classifiers are built and then they are used for detecting test examples. Outputs of these individual classifiers are then combined using fixed combination rules or training combination rules to generate consensus decisions.

5.2. General framework of classifier combination

We first convert the optical character recognition (OCR) problem to the classification. Suppose *img* is the image to be recognized. Let $D = \{D_1, ..., D_R\}$ be a set of classifiers, and let $S = \{c_1, ..., c_M\}$ be a set of potential labels (corresponding to characters need to be mapped). Alternatively, we may define the classifier output to be a *M*-dimensional vector

$$D_i(img) \;=\; [d_{i,1}(img),\ ...,\ d_{i,M}(img)] \tag{4}$$

where $d_{i,j}(img)$ is the degree of "support" given by classifier D_i to the hypothesis that *img* comes from class c_j. Most often $d_{i,j}(img)$ is an estimation of the posterior probability $P(c_i|img)$. In fact, the detailed interpretation of $d_{i,j}(img)$ beyond a "degree support" is not important for the operation for any of the combination methods studies here. It is convenient to organize the outputs of all *R* classifiers in a decision matrix as follows.

$$DP(img) \;=\; \begin{bmatrix} d_{1,1}(img) & ... & d_{1,M}(img) \\ ... & & \\ d_{i,1}(img) & ... & d_{i,M}(img) \\ ... & & \\ d_{R,1}(img) & \cdots & d_{R,M}(img) \end{bmatrix} \tag{5}$$

Thus, the output of classifier D_i is the i^{th} row of the decision matrix, and the support for class c_j is the j^{th} column. *Combining classifiers* means to find a class label based on the *R* classifiers outputs. We look for a vector with *M* final degrees of support for the classes, denoted by

$$D(img) = [\mu_1(w), \ldots, \mu_M(img)] \tag{6}$$

where $\mu_j(img)$is the overall support degree obtained by combining R support degrees $\{d_{1,j}(img), \ldots, d_{R,j}(img)\}$from outputs of the R individual classifiers, under a combination operator ⊕, as presented in the formula (7) below.

$$\mu_j(img) = \oplus\ (d_{1,j}(img), \ldots, d_{R,j}(img), \ \text{for } j = 1, \ldots, M \tag{7}$$

If a single class label of *img* is needed, we use the maximum membership rule: Assign *img* to class c_k iff

$$\mu_k(img)\ \geq \mu_t(img);\ \ "t = 1, \ldots, M. \tag{8}$$

If we assume that each individual classifier D_i is based on a knowledge source, namely f_i, and then when detecting characters for *img*, the classifier D_i outputs a probability distribution over the label set *S*, denote by $\left\{P\left(c_j \,|\, f_i\right)\right\}_{j=1}^{M}$. In other word, to distinguish the individual classifiers we assume each classifier D_i is based on a knowledge source $\boldsymbol{f}_i$, and consequently we can represent $d_{i,j}(img)$ by the posterior probability, $P(c_j|\boldsymbol{f}_i)$, or by following equation:

$$d_{i,j}(img) = P(c_j|\boldsymbol{f}_i) \tag{9}$$

Note that this representation is used for all types of generation of different individual classifiers even though it is more appropriate for individual classifiers based on different feature spaces than different machine learning algorithms.

Under a mutually exclusive assumption of a set $\boldsymbol{F} = \boldsymbol{f}_i$ $(i = 1, \ldots, R)$, the Bayesian theory suggests that the image *img* should be assigned to class c_k provided the a posteriori probability of that class is maximum, namely

$$k = \arg\max_j P\left(c_j \mid \boldsymbol{f}_1, \ldots, \boldsymbol{f}_R\right) \tag{10}$$

That is, in order to utilize all the available information to reach a decision, it is essential to consider all the representations of the target simultaneously.

The decision rule (10) can be rewritten using Bayes theorem as follows:

$$k = \arg\max_j \frac{P\left(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_R \middle| c_j\right) P\left(c_j\right)}{P\left(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_R\right)}$$

Because the value of $P(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_R)$ is unchanged with variance of c_j , we have

$$k = \arg\max_j P\left(f_1, \ldots f_R \middle| c_j\right) P\left(c_j\right) \tag{11}$$

As we see, $P(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_R|c_j)$ represents the joint probability distribution of the knowledge sources corresponding to the individual classifiers. Assume that these knowledge sources are conditionally independent, so that the decision rule (11) can be rewritten as follows:

$$k = \arg\max_j P\left(c_j\right) \prod_{j=1}^{R} P\left(\boldsymbol{f}_i \middle| c_j\right) \tag{12}$$

According to Bayes rule, we have:

$$P\left(\boldsymbol{f}_i \middle| c_j\right) = \frac{P\left(c_j \middle| \boldsymbol{f}_i\right) P\left(\boldsymbol{f}_i\right)}{P\left(c_j\right)} \tag{13}$$

Substituting (13) into (12), we obtain the Naive Bayesian (NB) Rule:

$$k = \arg\max_j \left[P\left(c_j\right)\right]^{-(R-1)} \prod_{i=1}^{R} P\left(c_i \middle| \boldsymbol{f}_j\right) \tag{14}$$

It is worth to emphasize that we can consider NB rule as the NB classifier built on the combination of all feature subsets $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_R$.

Product Rule

Like [14], if we assume that all prior probabilities $P(c_j)$ $(j=1, \ldots, M)$ are of equal value, and then we obtain the equation, as follows.

$$k = \arg\max_j \prod_{i=1}^{R} P\left(c_i \middle| \boldsymbol{f}_j\right) \tag{15}$$

The decision rule (15) quantifies the likelihood of a hypothesis by combining the a posteriori probabilities generated by the individual classifiers by means of a product rule.

5.3. OWA operators

5.3.1 OWA Operators and derive combination rules

The notion of OWA operators was first introduced in [25] regarding the problem of aggregating multi-criteria to form an overall decision function. A mapping

$$F : [0, 1]^n \rightarrow [0, 1]$$

is called an OWA operator of dimension n if it is associated with a weighting vector $W = [w_1, \ldots, w_n]$, such that 1) $w_i \in [0, 1]$ and 2) $\sum_i w_i = 1$, and

$$F(a_1, \ldots, a_n) = \sum_{j=1}^{n} w_i b_i$$

where b_i is the i^{th} largest element in the collection $a_1, \ldots, a_n$.

The fuzzy linguistic quantifiers were introduced by Zadeh in [26]. According to Zadeh, there are basically two types of quantifiers: absolute, and relative. Here we focus on the relative quantifiers typified by terms such as *there exists, for all.* A relative quantifier Q is defined as a mapping $Q : [0, 1]^n \rightarrow [0, 1]$ verifying $Q(0) = 0$, there exists $r \in [0, 1]$ such that $Q(r) = 1$, and Q is a non-decreasing function. For example, the membership function of relative quantifiers can be defined [27] as

$$Q(r) = \begin{cases} 0 & \text{if } r < a \\ \dfrac{r-a}{b-a} & \text{if } a \le r \le b \\ 1 & \text{if } r > b \end{cases} \tag{16}$$

with parameters $a, b \in [0, 1]$.

Then, Yager [25] proposed to compute the weights w_i is based on the linguistic quantifier represented by Q as follows:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{(i-1)}{n}\right) \text{, for } i = 1, \ldots, n. \tag{17}$$

5.3.2 OWA Operator Based Combination Scheme

Let us return to the problem of identifying the right character of a given optical image *img* of a character as described above. Let us assume that we have R classifiers corresponding to R information sources f_i (or R different machine learning algorithms), each of which provides a soft decision for identifying the right character of the optical image *img* in the form of a posterior probability $P(c_k|f_i)$, for $i = 1, \ldots, R$.

Under such a consideration, we now can define an overall decision function D, with the help of an OWA operator F of dimension R, which combines individual opinions to derive a consensus decision as follows:

$$D(c_k) = F(P(c_k|f_1), \ldots, P(c_k|f_R)) = \sum_{i=1}^{R} w_i p_i \tag{18}$$

where p_i is the i-th largest element in the collection $P(c_k|f_1), \ldots, P(c_k|f_R)$, and $W = [w_1, \ldots, w_R]$ is a weighting vector semantically associated with a fuzzy linguistic quantifier. Then, the fuzzy majority based voting strategy suggests that the optical image *img* should be assigned to class c_j provided that $D(c_j)$ is maximum, namely

$$j = \arg\max_k D(c_k) \;\; j = arg \tag{19}$$

As studied in [25], using Zadeh's concept of linguistic quantifiers and Yager's idea of associating their semantics to various weighting vectors W, we can obtain many commonly used decision rules as following.

Max Rule First let us use the quantifier *there exists* which can be relatively represented as a fuzzy set Q of [0, 1] such that $Q(r) = 0$, for $r < 1/R$ and $Q(r) = 1$; for $r \geq 1/R$. We then obtain from (17) the weighting vector $W = [1, 0,\ldots, 0]$, which yields from (18) and (19) the Max Decision Rule as

$$j = \arg\max_{k}\left[\max_{i} P(c_k \mid \boldsymbol{f}_i)\right] \tag{20}$$

According to the quantifier-based semantics associated with this weighting vector [25], the max decision rule states that a decision of labeling for example $\boldsymbol{e}$ should be made if *there exists* an individual classifier supports that decision. In other word, the label selection of an example is followed the decision of the best classifier or followed the decision that obtains the maximum degree of supports from all the individual classifier.

Min Rule Similarly, if we use the quantifier *for all* which can be defined as a fuzzy set Q of [0, 1] such that $Q(1) = 1$ and $Q(r) = 0$; for $r \neq 1$ [25]. We then obtain from (17) the weighting vector $W = [1, 0, \ldots, 0]$, which yields from (18) and (19) the Min Decision Rule as

$$j = \arg\max_{k}\left[\min_{i} P(c_k \mid \boldsymbol{f}_i)\right] \tag{21}$$

Also, according to the quantifier-based semantics associated with W, the min decision rule states that a decision of labeling for example $\boldsymbol{e}$ should be made if all individual classifiers support that decision.

Suppose that the i^{th} classifier supports class c_j a degree $d_{i,j}$(e), then it means this classifier opposes a degree (1 – $d_{i,j}$(e)) to c_j. For each support degree $d_{i,j}$(e) let we name its corresponding opposition is $\overline{d'_{i,j}}(\boldsymbol{e})$ (e) then the min rule can be rewritten as follow.

$$\hat{j} = \arg\min_{j=1\ldots c}\left[\min_{i=1}^{R}\left[1 - d_{i,j}(\boldsymbol{e})\right]\right] = \arg\min_{j=1\ldots c}\left[\min_{i=1}^{R}\left[\overline{d'_{i,j}(\boldsymbol{e})}\right]\right] \tag{22}$$

With this presentation of the min rule, the class/label selection of an example is followed the decision which minimums the opposition from all the individual classifiers.

Median Rule In order to have the Median decision rule, we use the absolute quantifier at least one which can be equivalently represented as a relative quantifier with the parameter pair (0; 1) for the membership function Q in (16). Then we obtain from (17) the weighting vector $W = [1/R,\ldots,1/R]$, which from (18) and (19) leads to the median decision rule as:

$$j = \arg\max_{k}\left[\frac{1}{R}\sum_{j=1}^{R} P(c_k \mid \boldsymbol{f}_i)\right] \tag{23}$$

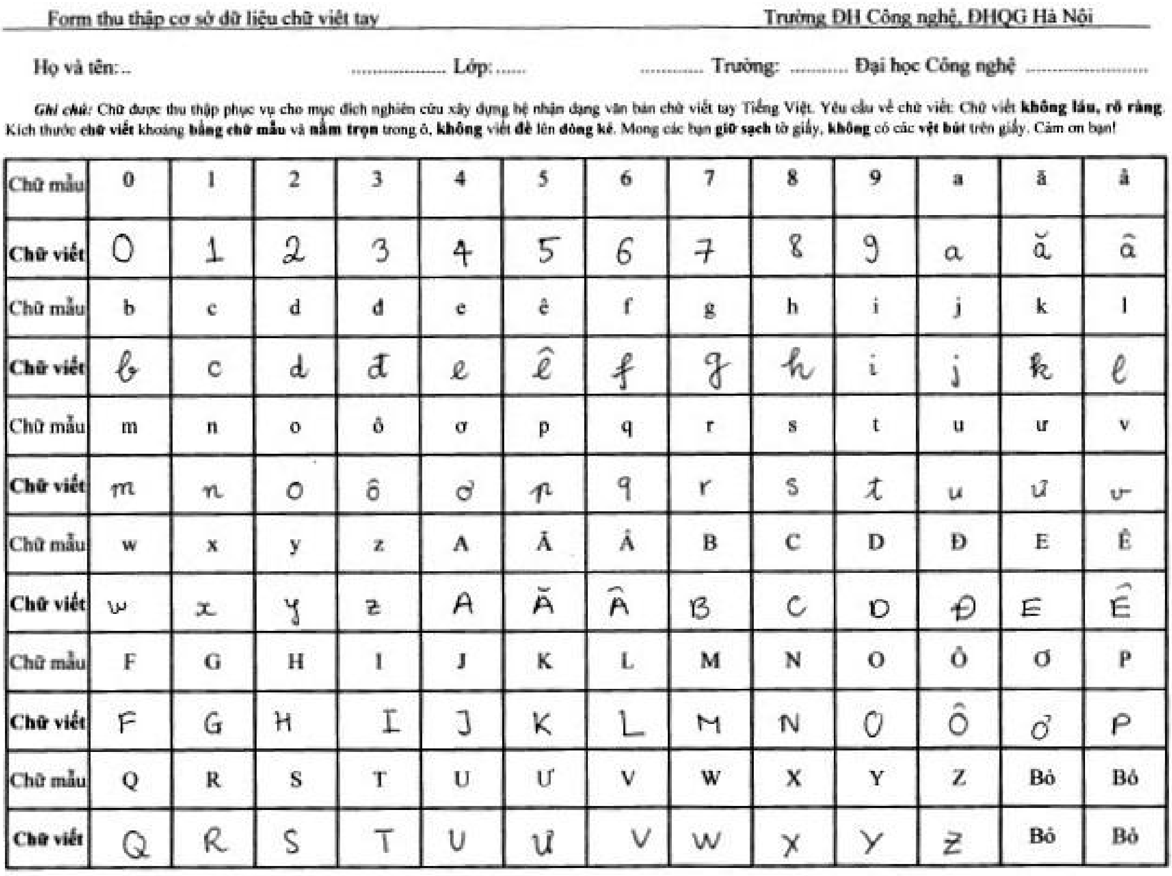
6. Experiment

6.1. Experimental models

For the experiment, we will investigate various models including the models for generating individual classifiers and the models for combining the individual classifiers. The machine learning algorithms to be used include neural network (NN), maximum entropy (ME), and support vector machine (SVM) which have shown their effectiveness for many classification/recognition problems. From these three machines learning algorithms, by using with the four feature sets (including gradient, structural, concavity feature sets and the combination of them) we totally generate 12 individual classifiers.

For classification combination, in section 4 we have presented various strategies including max rule, min rule, median rule, and product rule. There are two main scenarios of generating individual/base classifiers. In the first scenario, the individual classifiers are built based on different views of the data. It means that for each representation of the optical image of a character we have a base classifier. Concretely, we here consider the image under the three kinds of feature extraction: gradient, structural feature, and concavity extraction, each of which corresponds to a feature set. Note that we also integrate all these three feature sets for the overall representation (called GSC the feature set) of each optical image. In the second scenario, base classifiers use the same presentation with different machine learning methods (using NN, ME, and SVM).

In summary, we have four different sets of individual classifiers. For the first three sets of individual classifiers, each of them bases on one algorithms and different feature sets (i.e. gradient, structural, and concavity feature sets). The four set of individual classifier bases on one feature set (i.e. the GSC set) and different machine learning algorithms (i.e. NN, ME, and SVM).



Form thu thập cơ sở dữ liệu chữ viết tay Trường ĐH Công nghệ, ĐHQG Hà Nội

Họ và tên:.. .... Lớp:...... .... Trường: .... Đại học Công nghệ ....

Ghi chú: Chữ được thu thập phục vụ cho mục đích nghiên cứu xây dựng hệ nhận dạng văn bản chữ viết tay Tiếng Việt. Yêu cầu về chữ viết: Chữ viết **không láu, rõ ràng**. Kích thước **chữ viết** khoảng **bằng chữ mẫu** và **nằm trọn** trong ô, **không** viết **đè** lên **dòng kẻ**. Mong các bạn **giữ sạch** tờ giấy, **không** có các **vệt bút** trên giấy. Cảm ơn bạn!

Chữ mẫu	0	1	2	3	4	5	6	7	8	9	a	ă	â
Chữ viết	0	1	2	3	4	5	6	7	8	9	a	ă	â
Chữ mẫu	b	c	d	đ	e	ê	f	g	h	i	j	k	l
Chữ viết	b	c	d	đ	e	ê	f	g	h	i	j	k	l
Chữ mẫu	m	n	o	ô	ơ	p	q	r	s	t	u	ư	v
Chữ viết	m	n	o	ô	ơ	p	q	r	s	t	u	ư	v
Chữ mẫu	w	x	y	z	A	Ă	Â	B	C	D	Đ	E	Ê
Chữ viết	w	x	y	z	A	Ă	Â	B	C	D	Đ	E	Ê
Chữ mẫu	F	G	H	I	J	K	L	M	N	O	Ô	Ơ	P
Chữ viết	F	G	H	I	J	K	L	M	N	O	Ô	Ơ	P
Chữ mẫu	Q	R	S	T	U	Ư	V	W	X	Y	Z	**Bỏ**	**Bỏ**
Chữ viết	Q	R	S	T	U	Ư	V	W	X	Y	Z	**Bỏ**	**Bỏ**

Fig. 8. Form for collecting data.

6.2. Data preparation

Up to now there is no a standard database for this work. We therefore have built our self a database for isolated handwritten Vietnamese character recognition. Firstly we design a form by which a person can fill in the characters at specified places (see the figure 8). Secondly, after collecting these filled forms we built a tool for extracting the specified characters. To do this work, we have used 188 students from the University of Engineering and Technology, Vietnam National University of Hanoi. We have selected all 10 digits and 66 Vietnamese characters including upper and lower ones. After discarding illegal forms we obtain 6194 patterns. Note that the images of characters are then shrunk to 28x28 grey images which are exported to database files in MNIST [1] format which are normally useful for such the studies of text recognition. They are then divided into three sub-sets with 3716 patterns for the training set, 1239 patterns for the development set, and 1239 patterns for the test set.

Table 1. Experimental results (show the error rates of recognition)

	Individual classifiers			Combination rules			
	Gradient	Structural	Concavity	Median	Max	Min	Product
ME	11.2%	12.4%	14.5%	7.6%	7.6%	8.2%	**6.9**%
NN	11.1%	11.1%	15.5%	7.1%	8.1%	9.0%	8.6%
SVM	7.0%	7.0%	10.3%	4.6%	5.7%	5.0%	**4.5**%
GSC	7.0%(ME)	5.9%(NN)	**4.0**%(SVM)	4.3%	4.5%	**4.0**%	4.3%

6.3. Experimental results and discussion

Table 1 shows all of our experimental results which contain the error rates for all individual classifiers (generated by using various kinds of features and machine learning algorithms) and the combination models. The first three columns stand for three feature sets including gradient, structural, and concavity ones respectively. The next four columns stand for the combination strategies/rules including median, max, min, and product rules respectively. The first three rows present the case when individual classifiers are generated by using one machine learning algorithms with different features sets as mentioned above. Note that the last row presents the case when individual classifiers using the GSC feature set with different machine learning algorithm. From the obtained results we have the following discussions.

- Regarding the first three rows of the table, when using the feature sets separately we can see that gradient and structural feature sets have shown their similar effectiveness while using concavity feature set gives lower accuracies (i.e. higher error rates). Specially using the combination rules on the individual classifiers give better results in comparison with these individual classifiers. Among these combination rules the product rule gives the best results (error rates are 6:9% and 4:5%) in two cases and the median rule give the best result in the remain case (error rate is 7:1%). Note that the product rule is generated based on the assumption that the resources used in individual classifiers are independent with each other. From the way of feature extraction we can see that gradient features, structural features, and concavity features seems to be independent with each other, therefore it has shown the better results in most cases. The remaining case causes a difference may because the private characteristic of NN doesn't support the assumption of the product rule that all prior probabilities are of equal value.

- In the other aspect, when using the GSC feature set which is the combination of gradient, structural, and concavity features we can see that the obtained classifiers give the best results in comparison with using them separately, as shown at the last line of the table 1. Moreover, the support vector machine algorithms showed that it is the best method for this work. For the combination rules in this case, the min rule gets the best result and it is the same to SVM method (the error rates are equal to 4:0%).

Note that in the related work [9], the authors used some kinds of features including horizontal, vertical direction and the two diagonals of the optical images. These features are just a part of structural features as we presented in section 3. Therefore our work of feature extraction is more general than [9].

In summary the obtained results have shown that using GSC features gives low error rates. It means that combine all the three kinds of features (including gradient, structural, and concavity ones) will provide rich knowledge resources to recognize Vietnamese characters. The experiments also affirm that using the proposed combination rules will give the best results for all the cases, especially in the case we are having weak individual classifiers (when they uses separately feature sets). Even that in the last case when the support vector model has shown the same result as using the min rule, in general using combination strategies will provide us a safe way to obtain the best classifier. In some situations in which we obtain weak classifiers, for example when we do not have enough training data or do not have enough evidences then using combination is adequate to obtain a strong classifier. Furthermore in our opinion, we can use an effective strategy that uses a development data set to determine which one is the best among combination rules and individual classifiers.

7. Conclusion

To recognize isolated Vietnamese characters, this paper firstly investigated gradient, structural, and concavity characteristics of the character images for feature extraction. These kinds of features are used for the three machine learning algorithms: neural network, maximum entropy, and support vector machines. The experimental results have shown that gradient and structural feature sets give the same effect while concavity feature set gives higher error rate. Specially, the combination of these three feature kinds gives the best result. From these result, when conducting experiment on NN, ME, and SVM algorithm we obtain that SVM produces the best result.

Secondly, to investigate the effectiveness of applying classifier combination rules to this work, a general framework for classifier combination has been proposed and then derived several combination rules based on Naive Bayesian inference and OWA operators. These rules have been applied on individual classifiers which are based on different feature sets or different machine learning algorithms. The experimental results have shown the effectiveness of using the combination approach, specially in the case when these individual classifiers lack of adequate knowledge resources. In summary, using GSC feature set with SVM algorithm and classifier combination strategies have been shown to be the appropriate solution for recognizing isolated handwritten Vietnamese characters. In addition, by using a development set, we can decide which model is the most appropriate in a particular context.

Acknowledgments. The work in this paper was partly supported by the research project QG.07.25 granted by the Vietnam National University, Hanoi. We would like to thank all the project members for their support.

References

[1] S.M. Awaidah, S.A. Mahmoud, A multiple feature/resolution scheme to Arabic (Indian) numerals recognition using hidden Markov models, *Signal Processing* 89(6) (2009) 1176.

[2] C.Y. Suen, C. Nadal, R. Legault, T.A. Mai, L. Lam, Computer recognition of unconstrained handwritten numerals, *Proc IEEE* 80(7) (1992) 1162.

[3] J.C. Simon, Offline cursive word recognition, *Proc IEEE,* 80(7) (1992) 1150.

[4] S. Impedovo, P. Wang, H. Bunke, *Automatic Bankcheck Processing,* World Scientific, Singapore, 1997.

[5] A. Kaltenmeier, T. Caesar, J. Gloger, E. Mandler, Sophisticated topology of hidden Markov models for cursive script recognition. In Proc. of the Second Int, *Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, (1993) 139.

[6] U.V. Marti, H. Bunke, Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System Int'l, *J. Pattern Recognition and Artificial Intelligence,* vol. 15, No.1 (2001) 65.

[7] Alessandro Vinciarelli, Samy Bengio, Horst Bunke, *Offline Recognition of Large Vocabulary Cursive Hand-written Text Proceedings of 7th IEEE International Conference on Document Analysis and Recognition,* Edinburgh (2003) 1101.

[8] Nguyen Duy Khuong, Bui The Duy, Recognizing Vietnamese Online Handwritten Separated Characters, *Proceedings of the 2008 International Conference on Advanced Language Processing andWeb Information Technology* (2008) 279.

[9] Pham Anh Phuong, Ngo Quoc Tao, Luong Chi Mai, An Efficient Model for Isolated Vietnamese Handwritten Recog-nition, *International Conference on Intelligent Information Hiding and Multimedia Signal Processing* 2008.

[10] J.T. Favata, Offline general handwritten word recognition using an approximate BEAM matching algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 23 (2001) 1009.

[11] F. Al-Omari, Handwritten Indian numeral recognition system using template matching approaches, *IEEE International Conference on Computer Systems and Applications*, Beirut (2001) 83.

[12] P.D. Gader, M. Mohamed, J.H. Chiang, Handwritten Word Recognition With Character and Inter-Character Neural Networks, *IEEE Trans. Sys.Man Cybernetics*, vol. 27, No. 1 (1997) 158.

[13] M.Y. Chen, A. Kundu, J. Zhou, Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network, *IEEE Trans. on PAMI*, vol. 16 (1994) 481.

[14] J. Kitter, M. Hatef, R. Duin, J. Matas, On combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 2 No. 3 (1998) 226.

[15] A. Bellili, M. Gilloux, P. Gallinari, An hybrid MLP-SVM handwritten digit recognizer, *In Proc. of the 6th Int. Conference on Document Analysis and Recognition,* Seattle, USA, 2001, 28.

[16] L. Lee, N. Gomes, Disconnected handwritten numeral image recognition, *In Proc. of the 4th Int. Conference on Document Analysis and Recognition*, Ulm, Germany, vol. 2 (1997) 467.

[17] K. Maruyama, Y. Nakano, Recognition method for cursive Japanese word written in latin characters, *In Proc. of the 7th Int. Workshop on Frontiers in Handwriting Recognition,* Amsterdam, The Netherlands, 2000, 133.

[18] G. Kaufmann, H. Bunke, M. Hadorn, Lexicon reduction in an HMM-framework based on quantized feature vectors, *In Proc. of the 4th Int. Conference on Document Analysis and Recognition*, Ulm, Germany, volume 2 (1997) 1097.

[19] Roman Bertolami, Horst Bunke, Ensemble Methods to Improve the Performance of an English Handwritten Text Line Recognizer, *Proceedings of the 2006 conference on Arabic and Chinese handwriting recognition*, 265.

[20] Roman Bertolami, Horst Bunke, *Multiple Classifier Methods for Offline Handwritten Text Line Recognition,* MCS 2007, LNCS 4472, (2007) 72.

[21] S. Gunter, H Bunke, Optimization of Weights in a Multiple Classifier Handwritten Word Recognition System Using a Genetic Algorithm, *in Electronic Letters Computer Vision and Image Analysis* 3 (1) (2004) 25.

[22] Simon Gunter, Horst Bunke, Offline cursive handwriting recognition using multiple classifier systems on the influence of vocabulary, ensemble, and training set size, *Optics and Lasers in Engineering*, 43 (2005) 437.

[23] Pham Anh Phuong, Some methods for effectively extracting features for Isolated Vietnamese Handwritten Recognition, *Journal of Science* (Hue university), 53 (2009) 73. (in Vietnamese)

[24] D. Klein, K. Toutanova, H. Tolga Ilhan, S.D. Kamvar, C.D. Manning, Combining Heterogeneous Classifiers for Word Sense Disambiguation, *ACL WSD Workshop*, (2002) 74.

[25] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE Transactions on Systems, Man, and Cybernetics*, 18 (1988) 183.

[26] L.A. Zadeh, A computational approach to fuzzy quantifiers in natural languages, *Computers and Mathematics with Applications*, 9 (1983) 149.

[27] F. Herrera, J.L. Verdegay, A linguistic decision process in group decision making, *Group Decision Negotiation* (5) (1996) 165.