Comparative Study of Part-Based Handwritten Character Recognition Methods

Wang Song, Seiichi Uchida
Kyushu University, Fukuoka, Japan
wangsong@human.ait.kyushu-u.ac.jp, uchida@ait.kyushu-u.ac.jp

Marcus Liwicki
DFKI, Kaiserslautern, Germany
Marcus.Liwicki@dfki.de

***Abstract*—The purpose of this paper is to introduce three part-based methods for handwritten character recognition and then compare their performances experimentally. All of those methods decompose handwritten characters into "parts". Then some recognition processes are done in a part-wise manner and, finally, the recognition results at all the parts are combined via voting to have the recognition result of the entire character. Since part-based methods do not rely on the global structure of the character, we can expect their robustness against various deformations. Three voting methods have been investigated for the combination: single voting, multiple voting, and class distance. All of them use different strategies for voting. Experimental results on the MNIST database showed the relative superiority of the class distance method and the robustness of the multiple voting method against the reduction of training set.**

***Keywords*-handwritten character recognition, local features, voting**

I. Introduction

Part-based methods have been proposed for object recognition. In those methods, a query image is first decomposed into keypoints, each of which describes a local part of image. The part-based methods have following properties.

- The part-based methods will use multiple (say, 100) keypoints to represent a single image.
- Global features (e.g., position of keypoint in the image and global topological feature) are often disregarded on evaluating the similarity. This improves the robustness against the variations in object appearance.
- Similarity of two images depends on the comparison of two sets of keypoints. The images with the similar sets of keypoints will be considered as the images from the same class.
- Each class is sometimes represented by a large set of keypoints extracted from multiple (i.e. different) images of the class in order to deal with more variations.

For character recognition research, part-based methods have been rarely tried so far. This may be because most researchers believe that global features are very essential for representing characters. However, if we find that a part-based method is applicable to characters, we will be able to develop various part-based recognition methods which are robust to global deformation, partial occlusion, partial overlap and concatenation, broken (fragmented) stroke, etc. Moreover, the part-based method has a potential to develop word recognition methods without explicit segmentation into individual characters. This is similar to part-based object recognition, where a car is also recognized without explicit segmentation into tires, windows, body, etc.

Suen et al. [1] have tried a part-based method for character recognition; however, their trial still uses global features, that is, the global position of parts. An exceptional trial has been done quite recently in [2]–[4], where a part-based method is applied to an ancient manuscript recognition task. However, while those characters might be degraded, they are more comparable to machine printed characters nowadays, because of the regular writing style in medieval times. In [5], a part-based method was first applied to a general handwritten character recognition task. This part-based method employs the speeded-up robust features (SURF) [6] keypoints to describe the local parts of an image. Although the recognition rate of each single keypoint is only about 50%, a simple majority voting process of the recognition results of all the keypoints achieved to 93.8% as the final recognition rate, amazingly.

The purpose of this paper is to compare and analyse three part-based methods experimentally for handwritten character recognition:

- The first method is the single voting method proposed in [5]. It is based on a very simple process. As noted above, the simple method could achieve 93.8% accuracy.
- The second method, called multiple voting method, is newly proposed in this paper. This method is an extended version of the single voting method and can incorporate a class distribution of each keypoint.
- The third method is the class distance method, which has originally been proposed for object recognition in [7]. In spite of its simplicity, it has a theoretical background of statistic pattern recognition.

Experiments using handwritten digits from the MNIST will be set up for comparison of three methods. The comparison will be made first from the viewpoints of recognition accuracy and then go a deeper inspection to clarify why difference in the accuracy arises.

II. Descriptor of the Three Part-Based Methods

In this paper, all of the three methods use a version of the SURF keypoint detector and descriptor [6] for describing local parts. SURF detects keypoints as the local maxima of Hessian values in a scale space and then uses

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.167

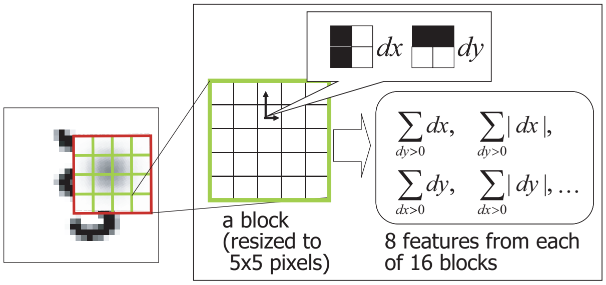
dx
dy
$\sum_{dy>0} dx$, $\sum_{dy>0} |dx|$,
$\sum_{dx>0} dy$, $\sum_{dx>0} |dy|$, …
a block
(resized to
5x5 pixels)
8 features from each
of 16 blocks

Figure 1. Process of SURF.

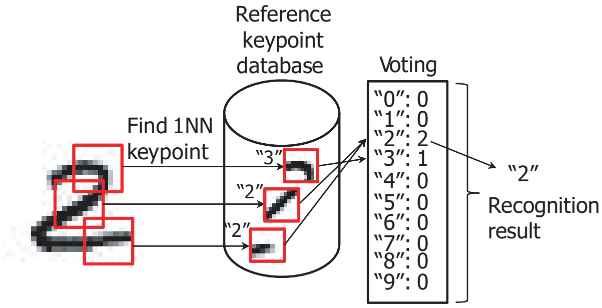

Figure 2. The single voting method.

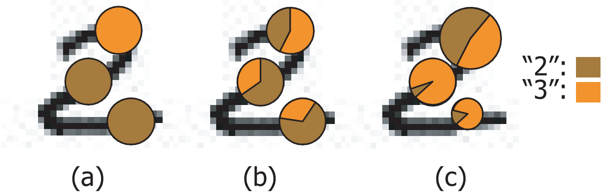

Figure 3. Illustrative interpertation of the votes in three methods. Here, for a simpler illustration, the numbers of classes and keypoints are limited to 2 and 3, respectively.

128-dimentional feature vector to describe the keypoints. Figure 1 shows how SURF describes a local part around a keypoint. The original SURF descriptors are all invariant against scale and rotation. In contrast, our SURF descriptor in this paper is changed to be scale-fixed and non-rotational for observing the basic performance of the part-based methods. The Euclidean distance in the 128-dimensional space will be used for measuring the dissimilarity of the keypoints. Note that in average, 59 keypoints are detected from a single digit image from the MNIST. (Further details are described in [5].)

III. Single Voting Method

The single voting method [5] has three steps as shown in Fig. 2. First, a reference keypoint database will be set up by extracting SURF keypoints from a set of training samples. Second, for each keypoint of query sample, its Euclidean 1-nearest-neighbor (1NN) keypoint is searched in the reference keypoint database. The query keypoint will be labeled by the class of its 1NN reference keypoint. Third, a simple majority voting of all the labels of query keypoints is conducted and the class with the maximum votes becomes the final recognition result of the query sample. In this paper, all the three part-based methods do not utilize the absolute position of individual parts and there is a possibility that a part from the top area of a character is selected as the 1NN of a part from the bottom area of another character. Since each query keypoint will contribute one single vote, this method is called the single voting method.

In [5], it was reported that the accuracy of the 1NN class in the second step is about 50%. That is, at the third step, only half of the votes go to the correct class. However, fortunately, the remaining votes will go to the other classes stochastically. Consequently, the correct class will have a great chance to win in the voting.

Figure 3 (a) gives an illustrative interpretation of votes in the single voting method. In this figure, the digit "2" is a query image and the circle corresponds to a keypoint of the query image. The color of the circle indicates the class to be voted by the keypoint. In the single voting method, each keypoint is related to one class and thus each the circle is filled by a single color. The class with the maximum votes will be the recognition result. Accordingly, in (a), the query image is recognized as "2".

The single voting method assumes that each keypoint belongs to only one class. In other words, it assumes that the appearance of the local part around each keypoint only appears in one class. In fact, at the first step of the single voting method, all the reference keypoints have a single label of the class from which they are extracted. Similarly, at the second step, the single class is assigned to the query keypoint as the class of its 1NN reference keypoint.

There are two skeptic viewpoints against the above assumption. First, two or more classes may have local parts with the same appearance. For example, samples from class "2" and "3" may have a very close appearance around their top-right curves. (In fact, because of this inter-class similarity, misrecognition happens in the second step.) Thus, it may be a more natural assumption that a single keypoint can belong to multiple classes (with a certain probability). Second, it may be better to consider how much each keypoint belongs to its class. In other words, two query keypoints with typical and rare appearances respectively should not have the votes with the same weight.

IV. Multiple Voting Method

The multiple voting method is based on a different assumption from the single voting method. Specifically, it assumes that a keypoint does not appears in only one class, that is, each reference keypoint appears in all the classes according to a certain probabilistic distribution. For example, assuming a reference keypoint with 50% probability for class

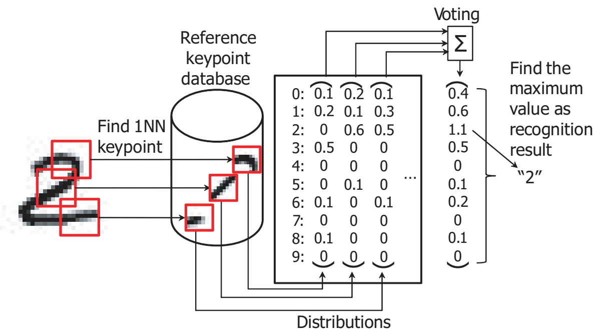

Figure 4. The multiple voting method.

1, 30% for class 2, 20% for class 3, and 0% for the other classes, the class distribution of this keypoint can be written as $(0.5, 0.3, 0.2, 0, 0, \cdots)$. Based on this assumption, during voting, if this reference keypoint is selected as 1NN, it will product the following weights: 0.5 to class 1, 0.3 to class 2, 0.2 to class 3, and 0 to the other classes.

For the realization of the multiple voting, the class distribution of each reference keypoint should be estimated as its first step. Two training sets, 1 and 2, are used for this estimation. First, from training set 1, reference keypoints are extracted like during the single voting method. Then, from training set 2, keypoints are extracted for estimating the class distribution of each reference keypoint. Specifically, for each keypoint of set 2, its 1NN is selected from the reference keypoints of set 1. Then, if a reference keypoint is selected 3 times by class 1, 3 times by class 2, 4 times by class 3, the class distribution of this reference keypoint can be obtained as $(0.3, 0.3, 0.4, 0, 0, \cdots)$. Although this is just an empirical approximation of the real class distribution in the SURF feature space, it will be experimentally sufficient to improve the accuracy of the part-based recognition process.

As shown in Fig. 4, the second and third steps of the multiple voting method are the same as the single voting method, except that the multiple votes will be done according to the class distribution. The class with the maximum votes will be the final recognition result.

In Fig. 3 (b) gives an illustrative interpretation of the votes in the multiple voting method. Now the vote of a query keypoint is accompanied by its class distribution. In this figure, the multiple colors of each circle indicate that each vote is shared by more than one class. Since the total value becomes 1, the circle can be drawn in the same size.

V. Class Distance Method

The class distance method is an answer to the second skeptic viewpoint against the single voting method. That is, by this method, we can consider how much each keypoint belongs to its class. The class distance method has a theoretical background of statistical pattern recognition (Appendix outlines the theory of the class distance method). According to [7], given a query sample Q, let $d_1, \ldots, d_n$ denote all the keypoints of Q. If we have reference keypoints from class C as $d_1^C, \ldots, d_L^C$, the class C of Q is determined by the following equation, where d_{1NN}^C is the 1NN reference keypoint of d_i:

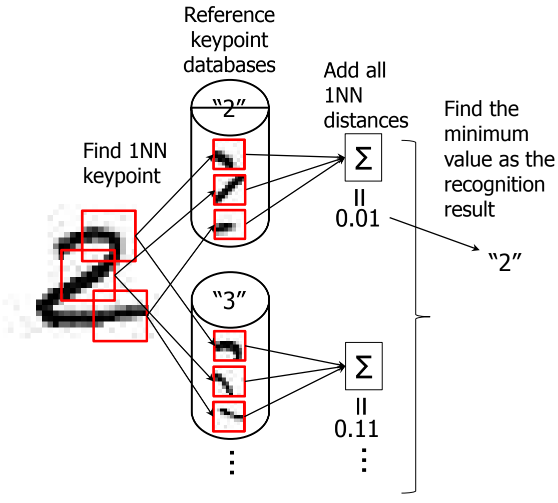

Figure 5. The class distance method.

$$\hat{C} = \operatorname*{argmin}_{C} \frac{1}{n} \sum_{i=1}^{n} \left(d_i - d_{\mathrm{1NN}}^C\right)^2. \tag{1}$$

Figure 5 shows the three steps of the class distance method. First, a reference keypoint database is prepared for each class like the other methods. Second, for each keypoint of the query sample, an 1NN reference is searched for among the reference database of the class C, and the Euclidean distances between the query keypoints and their 1NN keypoints (i.e., d_{1NN}^C) are summarized as the distance between Q and class C. The class with the minimum class distance will be the final recognition result.

The class distance method superficially doesn't employ any voting method; however, as shown in Fig. 3 (c), it can be interpreted as a weighted voting scheme. In fact, the 1NN distances to all classes from a query keypoint can be seen as a class distribution around the query keypoint. Then like the multiple voting method, the distributions of the query keypoints are summarized and the class with the minimum value becomes the final recognition result. The difference from the multiple voting method is that (i) the multiple voting method uses the class distribution of the 1NN reference keypoints, whereas the class distance method uses the class distribution of the query keypoint, and (ii) the class distance method does not employ any normalization for the class distribution. From Fig. 3 (c), we can see that the vote of the class distance is also shared by multiple classes. The size of the vote, however, are different.

Table I
RECOGNITION RATES (%).

	Size of training set	Single voting	Multiple voting	Class distance
Recognition rate	1000/class	93.57	94.92	97.91
	50/class	86.11	93.40	92.80

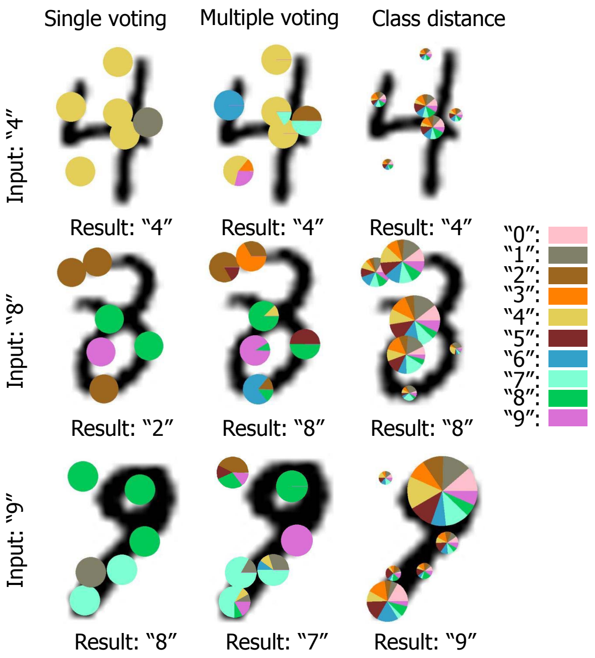

Figure 6. Examples of the votes in experiments.

VI. EXPERIMENTAL COMPARISON

A. Dataset

Using the MNIST isolated handwritten digit database, a comparative experiment has been made in order to observe the performance of the three methods. For stable extraction of SURF keypoints, each sample (a 28×28 grayscale image) was magnified four times after addition of 10-pixel margin (the final size is 192×192). The size of local parts for describing SURF feature was 16×16. The average number of keypoints per image was 59.

The first 1,000 samples per class of the MNIST training set were used for extracting the reference keypoints. The average number of the reference keypoints per class was 59,105. Then the next 4,000 samples per class were used as the training set 2 in the multiple voting method. All of the experiments used the MNIST test set (total number of the test samples is 10,000) for their test set. [1]

B. Recognition Rates

The recognition rate is shown in the first row in Table I. As can be seen the class distance method achieved the best accuracy among the three methods. This accuracy is far higher than the accuracy of the single voting method, which was reported in [5]. Thus, this new result increases the potential of the part-based method.

Figure 6 shows some examples from the experiments (with 1000/class training set) of the three methods. From the examples, we can see how the same reference keypoint affects the recognition result depending on the combination method. For the input "4" all of the three methods had the correct recognition result. For the input "8" the single voting method misrecognized as "2". This is because class "2" gets the maximum votes. In contrast, in the multiple voting method the votes change the distributions. Since class "8" occupies most area in total, the recognition result of the multiple voting method was "8". For the input "9" the single voting and multiple voting methods were both failed while the class distance method successfully recognized the sample as "9". In the class distance method the votes appear in different sizes, and the larger votes mainly determine the recognition result. As can be seen in the last image, class "9" occupied very small areas in the larger votes (recall that the class which with the smallest area will win in the class distance method), and therefore class "9" became the recognition result.

C. Discussion

First it needs to be discussed why the multiple voting method outperformed the single voting method. In the voting process, the "votes" of the multiple voting method are accompanied by the class distributions of 1NN reference keypoints. Compared with the votes of single voting method the votes of distributions contain more information. In fact, the single voting method can be seen as a special case of the multiple voting method where each distribution has the value 1 at one class and 0 at the remaining classes. Clearly, these special distributions are not the true distributions. In other words, in the single voting method, it is impossible to differentiate a reference keypoint which is definitely from a certain class from a reference keypoint which is ambiguous and thus lying on the boundary of two or more classes.

Second, the differences between the multiple voting method and the class distance method are discussed. Their first difference is how they utilize the class distributions. In the multiple voting method, the total value of a distribution is normalized to be 1. This means all the distributions (votes) has the same weight in voting process, whereas in the class distance method, they are not the same. Since the weights in the class distance method are derived theoretically, they will exert positive influence.

Another difference is the accuracy of the class distribution. In the multiple voting method, the class distribution of

[1] The recognition rates of the same MNIST by the state-of-the-art methods are listed in `http://yann.lecun.com/exdb/mnist/`

the 1NN reference keypoint is used. This indicates that if there is a large distance between the query keypoint and the 1NN reference keypoint, the distribution to be used may have a large error from the true one. In the class distance method, the class distribution of query keypoint is approximately estimated by using 1NN distance.

D. Experiments with a smaller datase

Another experiment was done with an extremely small database whose reference keypoints was extracted from 50 samples per class. The average number of reference keypoints per class was 2,968. The training set 2 of multiple voting was the same with the above experiment. The ratio of training set 1's size/training set 2's size determines the accuracy of approximation of distributions in the multiple voting method. Using training sets of the ratio 50/4,000, the multiple voting could have more accurate class distributions than using the ratio 1,000/4,000. As a result, the advantage of the multiple voting method became obvious. The test set of the experiments was also the MNIST test set.

The recognition rate is shown in the second row of Table I. It can be seen that the multiple voting had a much better recognition rate than the single voting method. In the table we can also see that the distance method didn't perform as well as the multiple voting method. It is because the multiple voting method had the training set 2 of a large size, thus the recognition rate of the multiple voting didn't decrease as fast as the class distance method.

VII. Conclusion

The purpose of this paper is to compare and analyze the performances of three part-based methods, called single voting method, multiple voting method, and class distance method. The experimental result has shown that the class distance method achieved the highest accuracy and the single voting method the lowest. An explanation of their difference was given from the viewpoint of voting: the performance of the three methods depends on what kind of votes they use in the voting process, the more information the votes contain, and the higher recognition rate the method has. It was also observed that the class distance method lost its superiority to the multiple voting method under a smaller database. This result also supports the above explanation because a vote by the multiple vote method contains more information than others even a smaller database size.

Future work will focus on more accurate estimation of class distribution at each keypoint with smaller dataset of samples. Practically, fast 1NN search methods can be expected based on the accurate class distribution. Furthermore, recognition tasks with more classes of the part-based method may be considered.

Acknowledgment

This research was partially supported by JST, CREST, JSPS and the Research Grant (No. 23300072) of The Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] C. Y. Suen, J. Guo, Z. C. Li, "Analysis and Recognition of Alphanumeric Handprints by Parts," IEEE Trans. SMC, vol. 24, no. 4, 1994.

[2] M. Diem and R. Sablatnig, "Are Characters Objects?," Proc. ICFHR, 2010.

[3] M. Diem and R. Sablatnig, "Recognition of Degraded Handwritten Characters Using Local Features," Proc. ICDAR, 2009.

[4] A. Garz, M. Diem, and R. Sablatnig, "Detecting Text Areas and Decorative Elements in Ancient Manuscripts," Proc. ICFHR, 2010.

[5] S. Uchida and M. Liwicki, "Part-Based Recognition of Handwritten Characters," Proc. ICFHR, pp. 545-550, 2010.

[6] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robus Features," Proc. ECCV, 2006.

[7] O. Boiman, E. Shechtman, and M. Irani, "In Defense of Nearest-Neighbor Based Image Classification," Proc. CVPR, 2008.

Appendix

In this Appendix, the theoretical background of the class distance method is outlined [7]. Consider a problem of determining the class C of a query sample Q. If the $p(C)$ is uniform, the Maximum-Likelihood (ML) classifier can provide the best class according to the Bayes decision rule:

$$\hat{C} = \operatorname*{argmax}_C p(C|Q) = \operatorname*{argmax}_C p(Q|C).$$

Consider the query Q can be represented by a set of keypoints $d_1, \ldots, d_n$. If we assume that all the keypoints $d_1, \ldots, d_n$ are i.i.d., given class C, namely:

$$p(Q|C) = p(d_1, \ldots, d_n|C) = \prod_{i=1}^{n} p(d_i|C).$$

Taking the log probability of the ML decision rule, we get:

$$\hat{C} = \operatorname*{argmax}_C \frac{1}{n} \sum_{i=1}^{n} \log p(d_i|C).$$

The estimation of $p(d|C)$ can be done by using Parzen density estimation. Letting $d_1^C, \ldots, d_L^C$ denote all the reference keypoints of class C, we get:

$$p(d|C) \sim \frac{1}{L} \sum_{l=1}^{L} K(d - d_l^C),$$

where $K(\cdot)$ is the Parzen kernel function. If we use Gaussian kernel, all of L reference keypoits is necessary for the calculation of $p(d_i|C)$. In practice, however, the Gaussian kernel decreases quickly and thus several reference keypoints $\{d_l^c\}$ neighbor to d_i are dominant for $p(d_i|C)$. As an extreme case, if we use the nearest d_l^c to d_i, we have (1).