Virtual Example Synthesis Based on PCA for Off-Line Handwritten Character Recognition

Hidetoshi Miyao and Minoru Maruyama

Dept. of Information Engineering, Faculty of Engineering,
Shinshu University 4-17-1 Wakasato, Nagano 380-8553, Japan
{miyao, maruyama}@cs.shinshu-u.ac.jp

Abstract. This paper proposes a method to improve off-line character classifiers learned from examples using virtual examples synthesized from an on-line character database. To obtain good classifiers, a large database which contains a large enough number of variations of handwritten characters is usually required. However, in practice, collecting enough data is time-consuming and costly. In this paper, we propose a method to train SVM for off-line character recognition based on artificially augmented examples using on-line characters.

In our method, virtual examples are synthesized from on-line characters by the following two steps: (1) applying affine transformation to each stroke of "real" characters, and (2) applying affine transformation to each stroke of artificial characters, which are synthesized on the basis of PCA. SVM classifiers are trained by using the training samples containing artificially generated patterns and real characters. We examine the effectiveness of the proposed method with respect to the recognition rates and number of support vectors of SVM through experiments involving the handwritten Japanese Hiragana character classification.

1 Introduction

To recognize handwritten characters, classifiers obtained by techniques of learning from examples are often used[4, 11]. Usually, the performance of the classifiers strongly depends on the quality of the training examples. If a large database which includes almost every possible variation of handwritten patterns is provided, the classifiers learned from the database are likely to perform quite well. However, in practice, collecting a sufficient number of *good* examples is not easy. It is usually costly and time-consuming. Learning algorithms often lack training examples that incorporate enough variations of handwriting. One possible method to overcome this difficulty is to synthesize virtual examples from a small number of real examples [5, 7, 9, 10, 13]. The simplest approach to synthesize training examples appears to be to apply geometrical transformations, such as the affine transform, to character images (i.e., the simple perturbation method[2]). Although this method is effective, an off-line character often lacks information on how it is written. Thus, it is often difficult to generate a set of artificial patterns which contains enough variations of handwriting from the

H. Bunke and A.L. Spitz (Eds.): DAS 2006, LNCS 3872, pp. 96–105, 2006.

character images. If on-line character data is available, the information on strokes can be utilized to synthesize virtual examples for off-line recognition[9, 10, 13].

In our previous paper[13], training examples for SVM classifiers were synthesized by applying the affine transform to each stroke of real training samples, and SVMs were learned on the basis of the generated patterns. Compared to the simple character-wise perturbation, the recognition rate was improved by this method. In this method, artificial patterns are generated on the basis of given specific character samples. When the number of samples is very small, it is likely that augmented patterns cannot efficiently cover the space of every possible character pattern. To overcome this difficulty, we tried to build a model of character patterns and then create virtual character patterns that can be used as "seeds" for pattern synthesis based on the affine transform. To build the character-generation model, DP-based character matching is carried out. Then, from the set of resultant difference vectors, the character generation model is built through the Principle Component Analysis (PCA) technique. To improve the performance of classifiers, SVMs are learned on the basis of the patterns which are generated by the proposed method. We examine the effectiveness of the proposed method through experiments involving the handwritten Japanese Hiragana character classification.

2 Generation of Virtual Examples in the Previous Work

In this chapter, we briefly review our previous method[13].

In our method, we use an on-line character database that contains stroke information to generate artificial character patterns for off-line classification [1, 9, 10]. In the on-line database, we assume that each character is divided into strokes, each of which is a connected component from pen-down to pen-up. We also assume that each stroke is represented as a sequence of 2D coordinates of pen positions. Most Japanese Hiragana characters consist of several strokes.

In our previous work, we simply applied the following affine transformation to each point of a stroke.

$$\mathbf{x}' = \bar{\mathbf{x}} + A(\mathbf{x} - \bar{\mathbf{x}}) + \mathbf{t}, \tag{1}$$

where $\mathbf{t} = (t_x, t_y)^T$ is the translation and $\bar{\mathbf{x}}$ is the center of the bounding box of the stroke. A 2×2 matrix A is given as the product of a shear matrix S and a rotation matrix R

$$A = A(\theta, \varepsilon_x, \varepsilon_y) = R(\theta)S(\varepsilon_x, \varepsilon_y), \tag{2}$$

where $R(\theta)$ and $S(\varepsilon_x, \varepsilon_y)$ are given by

$$S(\varepsilon_x, \varepsilon_y) = \begin{pmatrix} 1 & \varepsilon_x \\ \varepsilon_y & 1 \end{pmatrix}, \quad R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{3}$$

The transformation in (1) is specified by 5 parameters $(t_x, t_y, \theta, \varepsilon_x, \varepsilon_y)$. They are given uniformly at random for each stroke. After transforming 2D coordinates of strokes, a character image is generated through line thickening.

3 Virtual Example Synthesis Based on PCA

For a character class c, suppose that the number of real on-line samples is $K_c + 1$. One character sample is selected from the real samples as the base sample. The point sequence on the base sample is represented by $\{\mathbf{a}_{c,i} \mid \mathbf{a}_{c,i} = (x^b_{c,i}, y^b_{c,i})^T - \mathbf{P}^b_c, i = 1 \cdots I_c\}$, where $\mathbf{P}^b_c$ is the center position of the bounding box of the base sample. The point sequence on the other K_c real samples is represented by $\{\mathbf{b}^k_{c,j} \mid \mathbf{b}^k_{c,j} = (x^k_{c,j}, y^k_{c,j})^T - \mathbf{P}^k_c, j = 1 \cdots J^k_c, 1 \leq k \leq K_c\}$, where $\mathbf{P}^k_c$ is the center position of the bounding box of the kth sample. If two point sequences $\{\mathbf{a}_{c,1}, \mathbf{a}_{c,2}, \cdots, \mathbf{a}_{c,i}\}$ and $\{\mathbf{b}^k_{c,1}, \mathbf{b}^k_{c,2}, \cdots, \mathbf{b}^k_{c,j}\}$, are optimally matched, the accumulated distance $g(i,j)$ at DP is calculated by the following expression:

Initial value:

$$\begin{cases} g(i,1) = d(\mathbf{a}_{c,1}, \mathbf{b}^k_{c,1}) + d(\mathbf{a}_{c,2}, \mathbf{b}^k_{c,1}) + \cdots + d(\mathbf{a}_{c,i}, \mathbf{b}^k_{c,1}), \ for\ 1 \leq i \leq I_c \\ g(1,j) = d(\mathbf{a}_{c,1}, \mathbf{b}^k_{c,1}) + d(\mathbf{a}_{c,1}, \mathbf{b}^k_{c,2}) + \cdots + d(\mathbf{a}_{c,1}, \mathbf{b}^k_{c,j}), \ for\ 1 \leq j \leq J^k_c \end{cases} \quad (4)$$

Recurrence formula:

$$g(i,j) = d(\mathbf{a}_{c,i}, \mathbf{b}^k_{c,j}) + min \begin{cases} g(i-1, j-1), \\ g(i-1, j), \\ g(i, j-1), \end{cases} \quad (5)$$

where $d(\mathbf{a}_{c,i}, \mathbf{b}^k_{c,j})$ is the distance between two points, $\mathbf{a}_{c,i}$ and $\mathbf{b}^k_{c,j}$.

Optimal point sequence matching can be attained by calculating the minimum accumulated distance $g(I_c, J^k_c)$ between two patterns and backtracking the path obtained. In this way, we can obtain the corresponding point $\mathbf{b}^k_{c,j(i)}$ for the point $\mathbf{a}_{c,i}$, where $j(1), ..., j(I_c)$ represents the corresponding state between the two patterns. In Figure 1, we show an example of matching results.

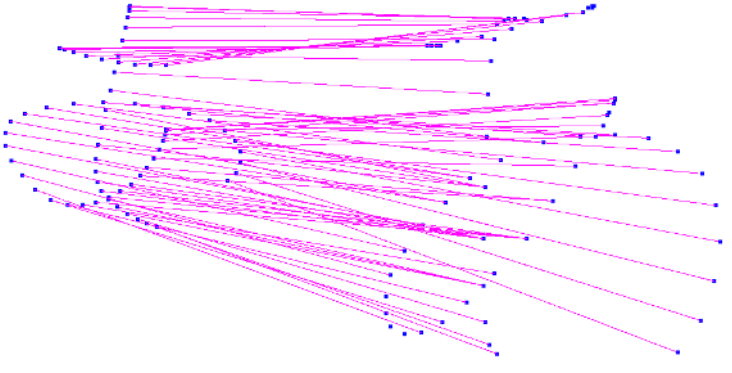

Fig. 1. Example of matching results for a point sequence on two patterns

The average difference vector between the base sample and the kth sample is given by

$$\begin{aligned}\mathbf{v}_{c,k} = (\ & x^k_{c,j(1)} - x^b_{c,1} - \bar{x}_{c,1},\ y^k_{c,j(1)} - y^b_{c,1} - \bar{y}_{c,1}, \cdots, \\ & x^k_{c,j(i)} - x^b_{c,i} - \bar{x}_{c,i},\ y^k_{c,j(i)} - y^b_{c,i} - \bar{y}_{c,i}, \cdots, \\ & x^k_{c,j(I_c)} - x^b_{c,I_c} - \bar{x}_{c,I_c},\ y^k_{c,j(I_c)} - y^b_{c,I_c} - \bar{y}_{c,I_c})^T,\end{aligned} \tag{6}$$

where

$$\begin{cases} \bar{x}_{c,i} = \dfrac{1}{K_c}\displaystyle\sum_{k=1}^{K_c}(x^k_{c,j(i)} - x^b_{c,i}), \\ \bar{y}_{c,i} = \dfrac{1}{K_c}\displaystyle\sum_{k=1}^{K_c}(y^k_{c,j(i)} - y^b_{c,i}). \end{cases} \tag{7}$$

We apply PCA to K_c difference vectors $\mathbf{v}_{c,k}$ and obtain eigen vectors $\{\mathbf{u}_{c,1}, \cdots, \mathbf{u}_{c,m}, \cdots, \mathbf{u}_{c,M}\}$ and eigen values $\{\lambda_{c,1}, \cdots, \lambda_{c,m}, \cdots, \lambda_{c,M} \mid \lambda_{c,1} \geq \lambda_{c,2} \geq \cdots \geq \lambda_{c,M} \geq 0\}$. Using the m-largest eigen values and corresponding eigen vectors, the following pattern generation model is obtained:

$$\begin{aligned}\hat{\mathbf{a}}_{c,m}(\alpha_1, \cdots, \alpha_m) = (\ & x^b_{c,1} + \bar{x}_{c,1}, y^b_{c,1} + \bar{y}_{c,1}, \cdots, x^b_{c,I_c} + \bar{x}_{c,I_c}, y^b_{c,I_c} + \bar{y}_{c,I_c})^T \\ & + \sum_{n=1}^{m} \alpha_n \mathbf{u}_{c,n} \quad (m \ll M),\end{aligned} \tag{8}$$

where α_n represents a normal random variable, ($\alpha_n \sim N(0, \lambda_{c,n})$). By using this model, we can generate artificial patterns. Examples of generated patterns are shown in Figure 2.

In our method, virtual examples are synthesized by applying the affine transformation described in Chapter 2 to each stroke of the generated patterns.

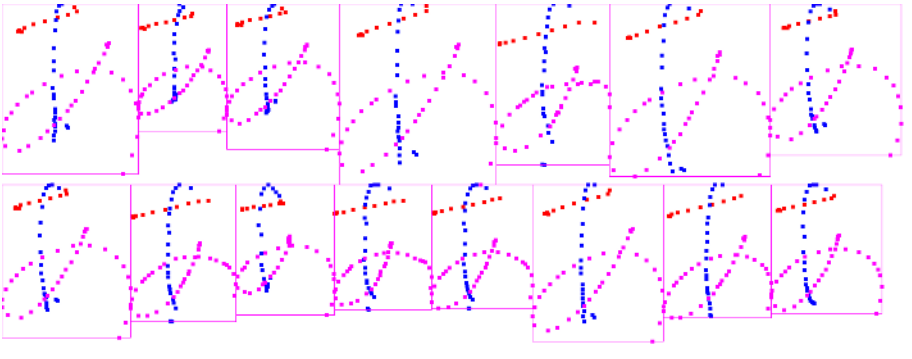

Fig. 2. Examples of patterns generated through the PCA technique (m = 3 in Eq.(8))

4 SVM Learning by Using Virtual Examples

In this chapter, we describe a method to learn SVM by using virtual examples which are generated by the methods described in Chapters 2 and 3. This is based on the method proposed by Miyao et al.[13].

For handwritten Hiragana recognition, we use SVM. SVM is a learning method based on the margin-maximization principle. SVM performs a binary classification by finding the optimal separating hyperplane in the feature space. Suppose that a set of training examples, $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ $y_i \in \{-1, 1\}$, is given; the SVM classifies the input $\mathbf{x}$ based on the function

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) - b \tag{9}$$

where $K(\mathbf{x}, \mathbf{y})$ is a kernel function which defines the inner product in the feature space. The coefficients α_is are non-zero only for the subset of the input data called support vectors.

The performance of SVM depends on the kernel. We use the RBF (Gaussian) kernel, which outperformed other commonly used kernels in preliminary experiments. The Gaussian kernel is given as

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2) \tag{10}$$

In our work, as an input to the RBF, we used directional feature patterns that have been commonly and successfully used for handwritten character recognition[6]. Each image is normalized to 64×64 pixels. Then, the contour of the normalized image is partitioned into 8×8 blocks. Four patterns emphasizing four directions (vertical, horizontal, left slant and right slant) at every block are detected. As a result, 256 features ($8 \times 8 \times 4$) are extracted. They are treated as components of a 256-dimensional feature vector.

To apply SVM (binary classifier) to a multiclass character recognition problem, we use the one-versus-rest (1vr)-type method. Suppose that we are dealing with an n-class problem. In the 1vr-type method, n SVMs f_i $(i = 1, \cdots, n)$, each of which classifies a single class from the other classes, are learned from examples. The resultant class c is determined as $c = \arg\max_i f_i(\mathbf{x})$. We used SVM^{light} [3] to train SVMs.

In our method, the affine transformation is applied to each stroke to synthesize artificial character patterns. SVM classifiers are learned on the basis of the generated patterns. If the effect of the stroke-wise affine transformation is too little, the generated patterns are not likely to contribute to improve the SVM. On the other hand, if the effect of the transformation is inappropriately large, the resultant patterns tend to act as *noise* in SVM learning. In this method, a preliminary SVM is trained by using the original data set. Then, to avoid the synthesis of inappropriate training samples, for each generated pattern, its *effectiveness* is examined by the absolute value of the SVM output. After this data selection process, using the augmented training samples, the final SVM is trained.

5 Experimental Results

To examine the effectiveness of the proposed method, classification experiments were carried out. In the experiments, we used the character database **HANDS-nakayosi_t-98-09**[12]. We selected the 10 Japanese Hiragana characters shown in Figure 3 and trained 1vr-type SVMs. A character data set written by 50 people was used for training. Another data set written by another 50 people was used for testing. For the performance test of the classifiers, 500 patterns for each character class were used.

For each class, the seeds were 50 real handwritten training samples and 50 samples generated according to (8) based on the real samples. Virtual examples were synthesized by applying the affine transformation to the 100 seeds. The parameters of the transformation $(t_x, t_y, \theta, \varepsilon_x, \varepsilon_y)$ used in the experiment were

$$|t_x|, |t_y| < 30, \quad |\theta| < 5(\text{degree}), \quad |\varepsilon_x|, |\varepsilon_y| < 0.3.$$

The training samples for the learning SVMs consist of the seeds and the virtual examples.

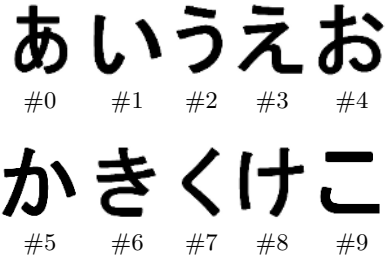

Fig. 3. Target Japanese Hiragana characters

Figure 4 shows a comparison of the recognition rates of SVMs trained using training samples based on a different number of eigen vectors m in (8). In this figure, 'without PCA' represents the results of the SVMs trained by using only real samples and virtual examples synthesized by applying the affine transformation to the real samples (i.e., this is the result for the traditional method[13]). This figure shows that the recognition rates of the proposed method for $m = 5$ are almost the same as those of the SVMs trained with real samples.

Our pattern generation method described in (8) depends on the choice of the base sample. The selection of the base sample may have an effect on the overall performance of the SVMs. To restrict the influence due to the selection of the base sample, we tried to increase the number of base samples. In the experiments, 5 base samples were taken from 50 real training samples, 10 seeds were generated from each base sample through PCA, and a total of 50 seeds were generated. The recognition results using SVMs trained by them are shown in Figure 5. This shows that the recognition rates with $m = 3$ for the proposed method outperformed the traditional method.

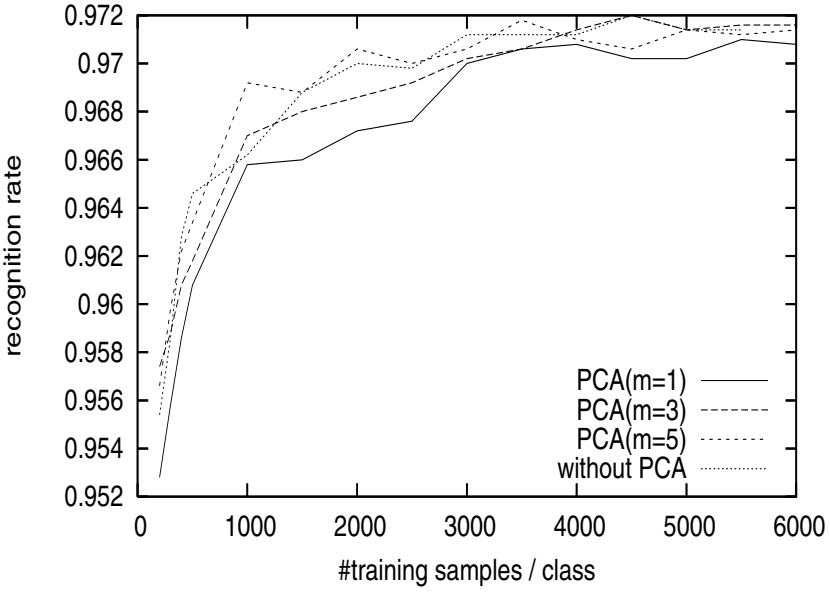

Fig. 4. Comparison of the recognition rates of SVMs trained using training samples based on different m in Eq. (8)

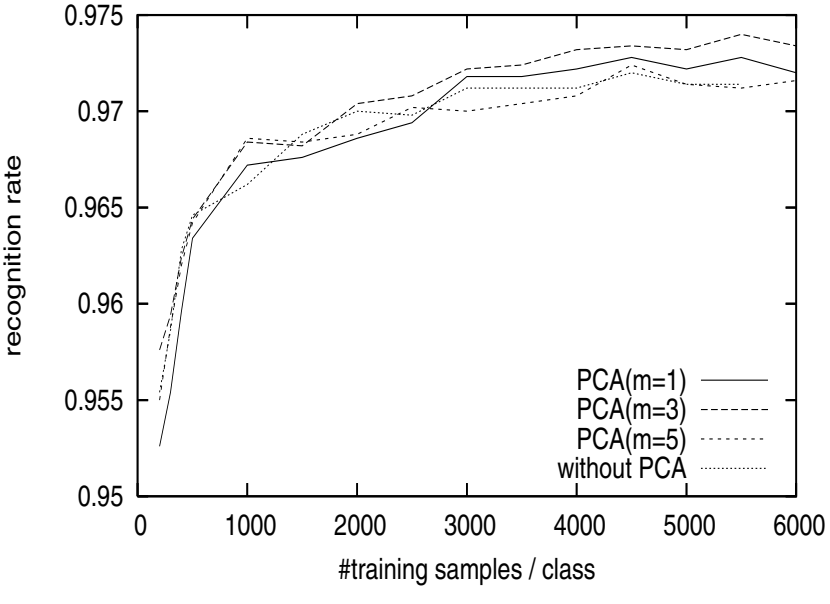

Fig. 5. Recognition rates of SVMs using seeds generated from 5 base samples

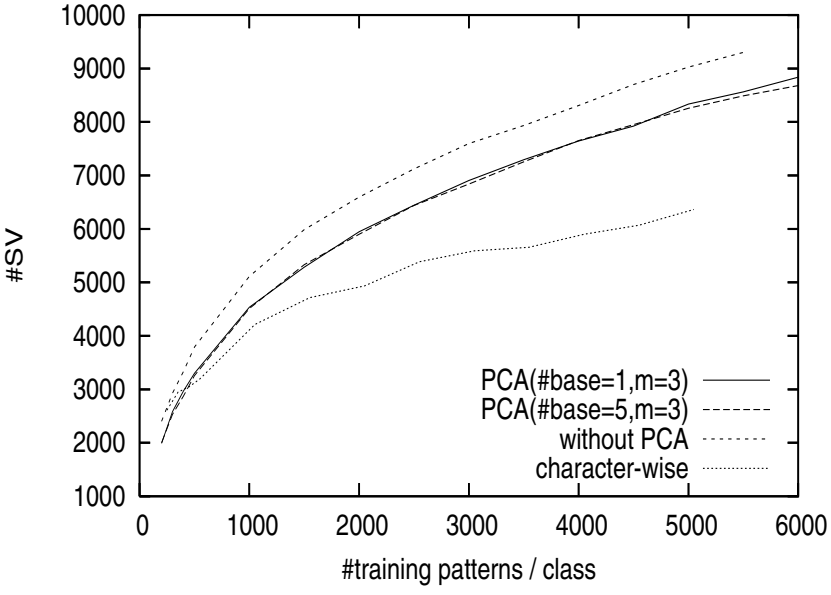

Fig. 6. Relationship between #training samples and #SV

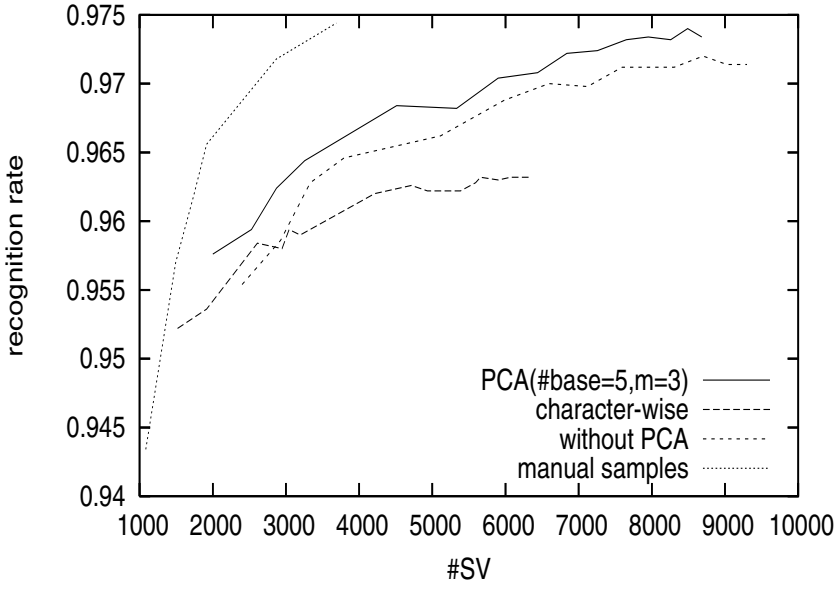

Fig. 7. Performance of generated SVMs

We have shown that the recognition rate of the SVM can be improved by using the artificially generated training samples. Usually, with an increase in the

number of training samples, both the recognition rate and the number of SV (support vectors) also increase. The increase of the number of SV leads to slow recognition speed. Ideally we should generate artificial patterns that result in both a high recognition rate and fast recognition speed (i.e., small number of SVs). We examined the total number of SVs of the trained 10 SVMs for the training samples shown in Figure 6. In this figure, 'character-wise' represents a simple perturbation method which applies the affine transform to character images. As the figure shows, the increase rate of #SV for the traditional method denoted by 'without PCA' is higher than that for the proposed method. This figure also shows that the number of base sample does not affect the increase rate of #SV.

To examine the effectiveness of the proposed method, we show the relationship between #SV and the recognition rates in Figure 7. In this figure, the curve denoted by 'manual samples' represents the performance of SVMs trained by only real handwritten samples. These training samples were written by the same (50) people, who also wrote the seed patterns of artificial samples used in the other methods. Preferably, the performance of SVMs trained by virtual examples should come close to that by SVMs trained by real samples. Although the proposed method is not satisfactory, the figure shows that our method outperformes the previously proposed method [13].

6 Conclusion

In this paper, to improve the performance of SVMs for off-line character recognition, we proposed a method to synthesize virtual examples for learning SVMs. Virtual examples were synthesized from on-line characters by the following two steps : (1) applying affine transformation to each stroke of "real" characters, and (2) applying affine transformation to each stroke of artificial characters, which were synthesized on the basis of PCA. SVM classifiers were trained by using training samples containing artificially generated patterns and real characters. We examined the effectiveness of the proposed method with respect to recognition rates and number of support vectors of SVM through experiments involving handwritten Japanese Hiragana character classification.

Our results indicated that, if the value of m (i.e., the number of eigen vectors) and the base samples are appropriately chosen, the recognition rates for the proposed method are higher than or equal to those by the method without PCA-based pattern augmentation and the classification time for the proposed method is also faster, since the number of support vectors of SVMs is further reduced. In our experiments, the best results were obtained when we used the parameter $m = 3$ and 5 base samples.

Future work includes :

- Determining why this method can reduce the number of support vectors more than the method without PCA.
- Designing a method to choose an appropriate number of eigen vectors (i.e. m) and the base samples so that SVM can have higher performance.

References

1. D.Ghosh and A.P.Shibaprasad, "An analytic approach for generation of artificial hand-printed character database from given generative models", Pattern Recognition, Vol.32, pp. 907 – 920 (1999).
2. T.M.Ha and H.Bunke, "Off-line, handwritten numeral recognition by perturbation method", IEEE Trans. PAMI, Vol. 19, No. 5, pp. 535 – 539 (1997).
3. T.Joachims, "Making large-scale SVM learning practical", In *Advances in kernel methods*, Chapter 11, MIT Press (1999).
4. K.Maruyama, M.Maruyama, H.Miyao and Y.Nakano, "A method to make multiple hypotheses with high cumulative recognition rate using SVMs", Pattern Recognition, Vol. 37,No. 2, pp. 241–251 (2004).
5. E.Miller, N.Matsakis and P.Viola, "Learning from one example through shared densities on transformation", Proc. CVPR2000, Vol.1, pp.464–471 (2000).
6. S. Mori, C. Y. Suen and K. Yamamoto, "Historical review of OCR research and development", Proc. IEEE, Vol.80, No.7, pp.1029–1058 (1992).
7. P.Niyogi, F.Girosi, and T.Poggio, "Incorporating prior knowledge in machine learning by creating virtual examples", Proc. IEEE, Vol.86, No.11, pp.2196 – 2207 (1998).
8. V.Vapnik, "Statistical Learning Theory", Wiley, New York (1998).
9. O.Velek, C.-L. Lieu, S.Jaeger and M.Nakagawa, "An improved approach to generating realistic Kanji character images from on-line characters and its benefit to off-line recognition performance" Proc. ICPR 2002, Vol.1, pp. 588 – 591 (2002).
10. O.Velek, S.Jaeger and M.Nakagawa, "A new warping technique for normalizing likelihood of multiple classifiers and its effectiveness in combined on-line/off-line Japanese character recognition", Proc. IWFHR 2002, pp. 177 – 182 (2002).
11. R. Schölkopf and A.J.Smola, "Learning with Kernels", The MIT Press (2002).
12. `http://www.tuat.ac.jp/~nakagawa/ipdb/`
13. H.Miyao, M.Maruyama, Y.Nakano and T. Hananoi, "Off-Line Handwritten Character Recognition by SVM based on the Virtual Examples Synthesized from On-Line Characters" Proc. ICDAR 2005, Vol.1, pp. 494 – 498 (2005).