2012 10th IAPR International Workshop on Document Analysis Systems

Expanding Recognizable Distorted Characters Using Self-Corrective Recognition

Masaki Tsukada, Masakazu Iwamura, and Koichi Kise
Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuencho, Naka, Sakai 599-8531 Japan
Email: tsukada@m.cs.osakafu-u.ac.jp, {masa, kise}@cs.osakafu-u.ac.jp

***Abstract*—Large datasets are always demanded for better recognition performance. However, it is not easy to produce them because costly and slow human operators have been necessary for labeling. In the current paper, in order to resolve the problem on yielding large datasets, we propose a scenario for automatic labeling based on the self-corrective recognition algorithm. The strong point of the proposed method is the capability of expanding recognizable distorted characters unlike existing methods. In the experiments, we show a possibility to realize automatic labeling by the method.**

***Keywords*-character recognition; self-corrective recognition; semi-supervised learning; transductive transfer learning; large dataset; affine distortion**

I. Introduction

There is a growing demand for new applications and services in reading scene texts, which is motivated by wide-spread availability and improvement of mobile devices having cameras (e.g., a camera phone). There are some commercial services available on reading text captured with a camera. Google Goggles[1] is a smartphone application to read texts captured with a built-in camera in addition to search the web using a photo. Evernote[2] is an application to help the user retrieve his/her data including scene texts with a keyword by indexing them. Tangochu[3] is a service to extract words from pictures taken by users.

However, these existing applications and services are not perfect; characters they can recognize are quite limited in comparison with those human beings can. For example, the characters in scene images have more various characters than those on documents as shown in Fig.1. This variety makes character recognition difficult [1]. Moreover, it is known that recognition of camera-captured character images by the computer is spoiled by some extent of perspective distortion, lighting artifacts, occlusion, low resolution, out of focus and so on. Therefore, recognizing character images suffering from various disturbances is quite challenging.

A feasible approach to recognize characters suffering from these disturbances is the memory-based approach [2]. In the approach, a lot of templates are stored in the database in the training phase, and the closest template to the query character image is searched to determine the output of the classifier in the recognition phase. It is obvious that the larger the number of templates stored in the database becomes, the more the recognition performance increases. Therefore, the biggest problem to take the approach is how to collect a lot of templates. That is, the problem of datasets.

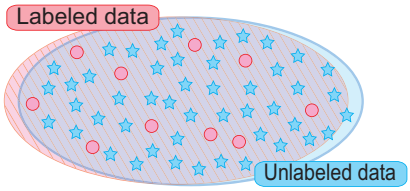
Figure 1. Examples of various character images in scenes.

In the field of camera-based character recognition, we are facing serious lack of large datasets. The largest one currently available is NEOCR containing 5,238 words which are labeled manually by one person in three months [3]. A larger dataset which might be available soon is 1 million digits dataset of house numbers originated from Google Street View; these digits are automatically extracted and recognized, and then verified by human operators. Human operators are involved in the datasets. Needless to say that hiring human operators is costly and time-consuming; while a cheaper solution (e.g., use of Amazon Mechanical Turk[4] [4]) might relax the problem, it is not the complete solution to yield further larger datasets. Therefore, we need a better way beyond manual labeling.

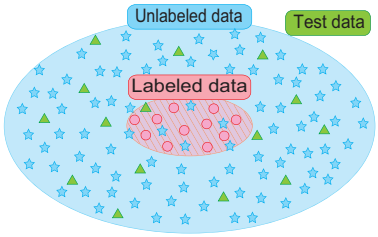
A promising approach to yield enormous datasets automatically is the self-corrective recognition algorithm which trains or adapts a classifier using unlabeled data [5], [6]. Since the algorithm has a capability to improve the classifier without labeled data, there is a possibility to realize a dreamlike automatic labeling system. Imagine a positive cycle based on the algorithm that the classifier continues to acquire the ability to recognize new images suffering from different degradation — finally, we can automatically obtain a clever classifier which can recognize various degraded character images and a large dataset containing the degraded character images! The largest bottleneck of the scenario is the assumption of the algorithm that labeled and unlabeled data come from the same source as shown in Fig.2a. In other words, they must have the identical distribution in the feature space. Since images suffering from different

[1] http://www.google.com/mobile/goggles/
[2] http://www.evernote.com/
[3] http://tangochu.jp/en/
[4] https://www.mturk.com/mturk/

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.37



(a) Distributions of general semi-supervised learning.



(b) Distributions of our research.

Figure 2. Assumption of distributions of labeled and unlabeled data.

distortions have different distributions as shown in Fig.2b, relaxing the assumption is crucial to realize the scenario.

In the current paper, following the scenario, we present an experimental study on expanding recognizable distorted characters dissimilar to the labeled data (initial templates) used to train the classifier. The problem setting is as follows. We use the same algorithm as the original paper [5] but a different assumption on data; in the training phase a standard labeled character image of each class is stored in the database, and in the recognition phase test images suffering from affine distortions (test data) are recognized. Since there is a big gap between the labeled data and test data, we have to make an effort to interpolate them. In the experiments, we evaluate possible ideas to realize it and show possibility that various characters can be recognized.

II. Related Work

The self-corrective recognition algorithm is regarded as self-training in the context of semi-supervised learning [7]. These methods use a small amount of labeled data and a large amount of unlabeled data. In handwriting recognition, a series of papers using the self-training are published by the same group (e.g., [8]). Graph-based semi-supervised learning approach uses a similar idea to the proposed method; they interpolate the labeled data by unlabeled data [7], [9]. However, all the existing methods in the semi-supervised learning assume the distributions of the labeled data and unlabeled data are very similar as shown in Fig.2(a). On the other hand, our research assumes the distributions of them are different as shown in Fig.2(b). This is because the purpose of our research is to recognize various characters having various distortions.

Transductive transfer learning is a framework using labeled and unlabeled data having different distributions [10].

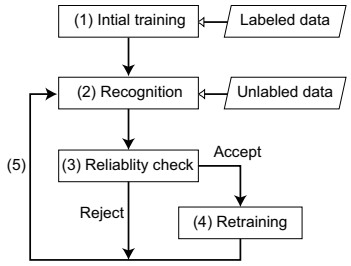


Figure 3. Overview of the training phase of the self-corrective recognition.

The framework prepares two domains: D_S and D_T. D_S and D_T consist of labeled and unlabeled data, respectively. The aim of the framework is to extract the knowledge from D_S and transfer the knowledge to D_T. Our research is close to the framework because both satisfy the conditions above. However, there is difference; the difference between the framework and our research is the number of labeled data. The former requires a lot of labeled data to extract the knowledge. But, the latter assumes a few labeled data. Thus, existing methods in the framework are not directly applicable to the task in this paper.

III. Self-Corrective Recognition for Data from Different Sources

An overview of the training phase of the self-corrective recognition algorithm is shown in Fig. 3. The process consists of (1) train the initial classifier using labeled data, (2) predict class labels of unlabeled data by recognizing the unlabeled data with the current classifier, (3) check the reliability of the labels, (4) retrain the classifier with data having reliable labels predicted, (5) repeat the items (2)-(4). As mentioned above, we use the same algorithm as the original paper [5] but a different assumption on data; there is a big gap between labeled data and test data.

In such a case, the classifier can initially recognize only characters having similar shapes to the labeled datum for training. That is to say, it cannot recognize heavily distorted ones. In order to recognize these characters, we have to make an effort to interpolate unlabeled data. If the unlabeled data are similar to the labeled datum, the classifier can be trained correctly and its performance increases. Conversely if the unlabeled data are not similar, the classifier misrecognizes and is trained incorrectly. As a result, performance declines.

Therefore, a possible idea to avoid failure on training is to order unlabeled data according to Euclidean distances between the features of the unlabeled data and the labeled datum. The effectiveness of the ordering is examined without performing reliability check (i.e.,without rejection) in the first series of experiments. In the second series of experiments, the effect of rejection is examined.

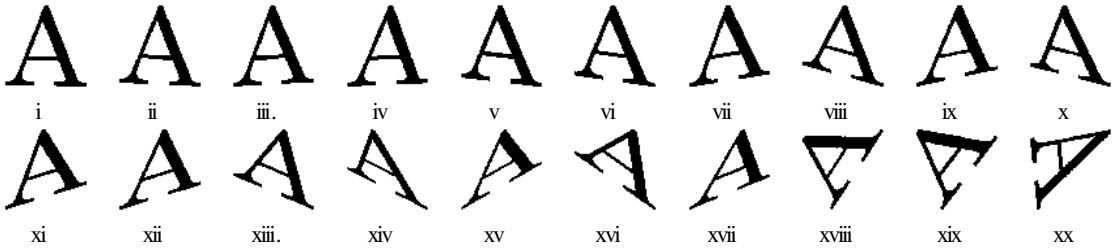


Figure 4. Similarity groups: examples of distorted images of 'A' belonging to the groups.

IV. Distorted Character Images

A. Affine transformation

Affine transformation is a common 2D graphic geometric transformation. Excluding a translation, an affine transformation matrix is represented by

$$\boldsymbol{T} = \begin{pmatrix} \beta & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & \tan\phi \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & \frac{1}{\alpha} \end{pmatrix}, \tag{1}$$

where β, θ, ϕ and α are the parameters of scale, rotation, shearing, and independent scaling, respectively. The matrix $\boldsymbol{T}$ projects a coordinate $\boldsymbol{x}$ to another one $\boldsymbol{y}$ by $\boldsymbol{y} = \boldsymbol{T}\boldsymbol{x}$. In this paper, β and α were always set to 1 because images were normalized after affine transformation. θ and ϕ were selected from $[-70, -69.75, -69.5, \ldots, +70]$ and $[-50, -49.75, -49.5, \ldots, +50]$, respectively. By combining them, we applied 224,961 affine transformations for each character (224,961 comes from 561 $\times$ 401).

B. Similarity groups of character images

In the experiment, we calculate Euclidean distances between the features of affine distorted character images and the labeled datum. The features were calculated as follows. Affine transformations were applied to the character images. The images were normalized to 64 64 pixels and binarized. From a binarized image, a 4096-dimensional binary vector was extracted, each element of which corresponded to a pixel. The principal component analysis was applied to the vectors and 40-dimensional real-value vectors were obtained.

We categorized the images into 20 groups according to distances; as shown in Fig. 4, these groups are called similarity groups that were equally divided between the maximum and minimum of distances. For example, group i consists of images having smaller distances, group xx consists of images having larger distances.

V. Experiments

We performed two series of experiments. In the first series of experiments, we control the order of unlabeled data in order to examine the effectiveness of the ordering. The order of unlabeled data was determined by the similarity groups. In order to group unlabeled data correctly, we assume that the ground truth of unlabeled data is known. However, in practice, the ground truth of unlabeled data is unknown. Therefore, in the second series of experiments, unlabeled data for retraining were selected using reliability check on the condition that the ground truth is unknown.

A. Experimental Settings

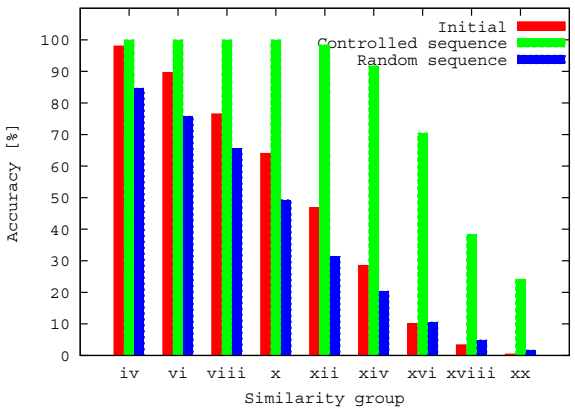
In order to realize the memory-based method, we used a nearest neighbor classifier based on a hash-based approximate nearest neighbor search [11]. The reason why a hash-based method was used is storing new data is quite faster than tree-based methods.

Capital alphabet letters of Century and Arial fonts were used. Since both results were similar, we show only the results of Century. Affine transformations were applied to the images of the letters of 120pt. The same feature vectors used for grouping in Section IV-B were used.

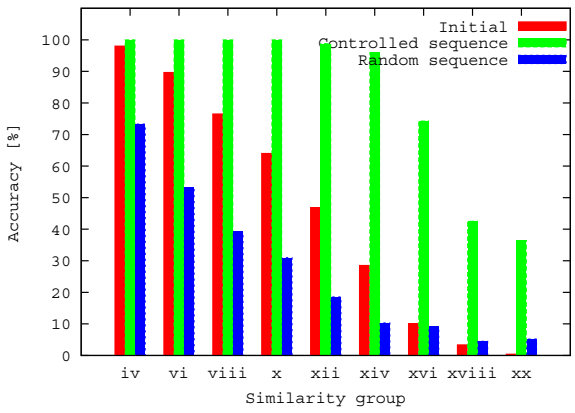
B. Experiment 1

In the first series of experiments, the effectiveness of the ordering is examined on the one-path retraining. Reliability check was not performed (i.e., without rejection). For the sake of that, we compared two sequences of unlabeled data in different orderings. One is the sequence being taken into account the ordering, which is called *controlled sequence*; unlabeled character images belonging to group i were used for retraining first and those belonging to group xx were used later. The images within a group were shuffled. The other is the sequence without being taken into account the ordering, which is called *random sequence*. Two sets of unlabeled images having different numbers of images were selected randomly, (a) 77,220 and (b) 1,909,180. The reason for preparing two sets is that we make sure the difference of the accuracies by the number of unlabeled data to train. In order to evaluate performance of the classifier, 100 images per group iv, vi, ..., xx were selected as test data. The reason for not using group i, ii and iii was that the number of character images of those groups were insufficient.

The recognition results are shown in Fig.5. By comparing the controlled sequence and random sequence, the former outperformed the latter. The accuracy of the former improved as compared with the accuracy of the initial. On the other hand, the accuracy of the latter decreased



(a) Retrained with 77,220 unlabeled images.



(b) Retrained with 1,909,180 unlabeled images.

Figure 5. Relationship between similarity groups and accuracy.

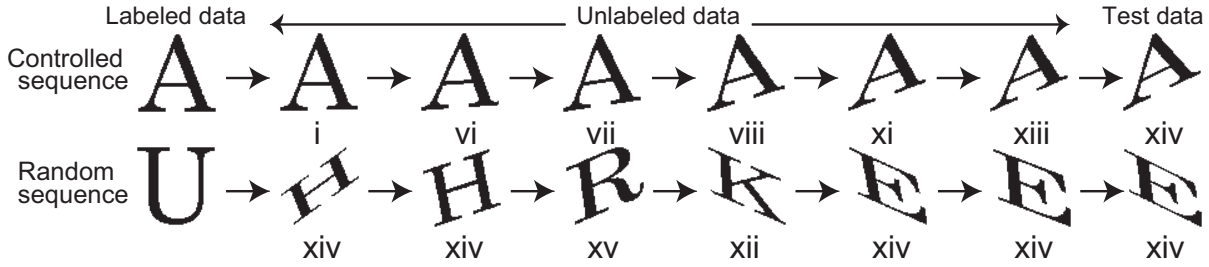


Figure 6. Genealogy of retraining. This is an example to show that there is a chance that wrong retraining can occur in the random sequence in an early stage. Each image at the left end is the labeled datum. Each image at the right end is a test datum. The remaining are unlabeled data. Firstly, the second from the left end was used for retraining earlier and its approximate nearest neighbor was the labeled datum. Secondly, the third from the left end was used for retraining and its approximate nearest neighbor was the second from the left end and so on. Finally, the approximate nearest neighbor of the test datum was the second from the right end. The upper row represents a successful case in controlled sequence (true class: 'A') and the lower row does a failure case in random sequence (true class: 'U'). Each numeral under an image represents a similarity group.

Table I
THE TRAINING PRECISION C, THE FALSE TRAINING PRECISION M AND THE REJECTING RATE R OF CONTROLLED AND RANDOM SEQUENCES IN EXPERIMENT 1.

		C[%]	M[%]	R[%]
Controlled sequence	(a)	75.61	24.39	-
	(b)	70.90	29.10	-
Random sequence	(a)	31.34	68.76	-
	(b)	11.95	88.05	-

except groups xvi, xviii and xx. This result represents we need to control the order of the unlabeled data to recognize distorted characters. For further investigation, the training precision $C = \frac{N_c}{N_c+N_m+N_r}$, the false training precision $M = \frac{N_m}{N_c+N_m+N_r}$, and the rejecting rate $R = \frac{N_r}{N_c+N_m+N_r}$ are defined, where N_c, N_m and N_r are the number of correctly labeled characters, mislabeled characters and rejected characters during retraining, respectively. C, M and R using the controlled sequence and the random sequence are shown in Table I. They show the importance of the ordering of unlabeled data. By comparing Figs. 5(a) and 5(b), Fig. 5(b) achieved better performance. This shows that the larger number of unlabeled data help improvement of recognition performance. One reason might be distances between unlabeled data used for retraining were closer in Fig. 5(b) than Fig. 5(a).

Fig. 6 represents the genealogy of retraining. The upper row represents a successful case in the controlled sequence and the lower row does a failure case in the random sequence. In the random sequence, the performance was degraded. This also shows that it is necessary to control the ordering of unlabeled data so as to avoid failure on retraining.

C. Experiment 2

In the second series of experiments, the effect of rejection is examined on the loop-path retraining. From the results of the previous experiments, it is obvious that controlling the ordering of unlabeled data is crucial to avoid failure on retraining. However, the previous assumption is not realistic because information on the true class is needed in the process of the ordering. Therefore, in the second

Table II
THE TRAINING PRECISION C, THE FALSE TRAINING PRECISION M, THE REJECTING RATE R AND THE NUMBER OF UNLABELED DATA USED FOR RETRAINING IN EXPERIMENT 2.

	Threshold	C[%]	M[%]	R[%]	# of unlabeled data used for retraining
fixed	1.5	16.05	31.63	52.32	910,224
	2.0	7.23	7.73	84.97	286,921
	2.5	1.28	1.00	97.72	43,611
	3.0	0.29	0.16	99.55	8,521
dynamic		53.37	46.62	0.00	1,909,180

series of experiments, we carry out experiments in a realistic condition. That is, using the random sequence in the previous experiments and introducing rejection.

As a criterion for rejection in the reliability check,

$$\frac{d_2}{d_1} > \text{threshold} \tag{2}$$

is defined, where d_1 and d_2 are the distances between an unlabeled datum and its first and second approximate nearest neighbors, respectively. Only if the inequality is satisfied, the unlabeled datum is accepted. Even if the unlabeled data do not satisfy the inequality (2), the unlabeled data may satisfy the inequality after retraining other unlabeled data which satisfy the inequality. Therefore, the retraining process is repeated so as to train the classifier with a lot of data.

In this section, two kinds of experiments were carried out: fixed and dynamic thresholding. The purpose of the fixed thresholding experiments is to examine the effectiveness of rejection in different values of thresholds: 1.5, 2.0, 2.5 and 3.0. 1,909,180 unlabeled images used in the experiments were selected randomly. The same test data as in the first series of experiments were used.

The recognition results of fixed thresholding are shown in Fig. 7. The smaller threshold becomes, the larger the number of unlabeled data used for retraining increases. Therefore, in the case of smaller threshold (e.g., 1.5), the result was similar to the previous experiments. On the other hand, in the case of larger threshold (e.g., 2.5 and 3.0), the result after retraining was closer to that of the initial state because a limited number of unlabeled data were used for retraining. The cases in the middle (e.g., 2.0) achieved better recognition performance. It was important to keep high training precision as compared with the false training precision and less rejecting rate as shown in Table II.

The purpose of the dynamic thresholding experiments is to simulate a realistic solution, repeating the retraining process with decreasing the threshold. If the inequality

$$\frac{N_{c-1}}{N_c} < 0.90 \tag{3}$$

satisfied, the threshold was decreased by 0.2. Here, N_c is the sum of new training characters in the cth cycle. Even if the number of new training characters is insufficient, the threshold is decreased to train the classifier with more

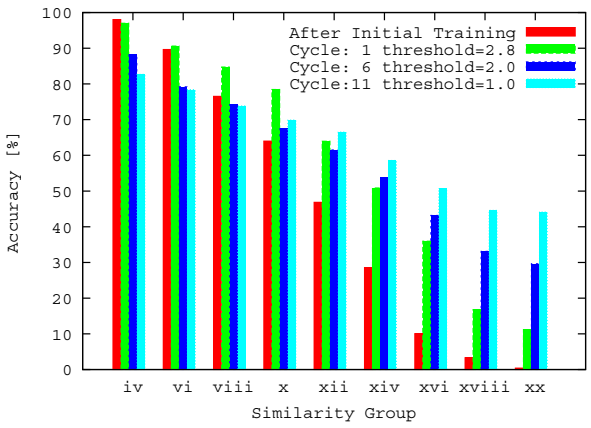


Figure 8. Relationship between accuracy and similarity groups in the dynamic thresholding.

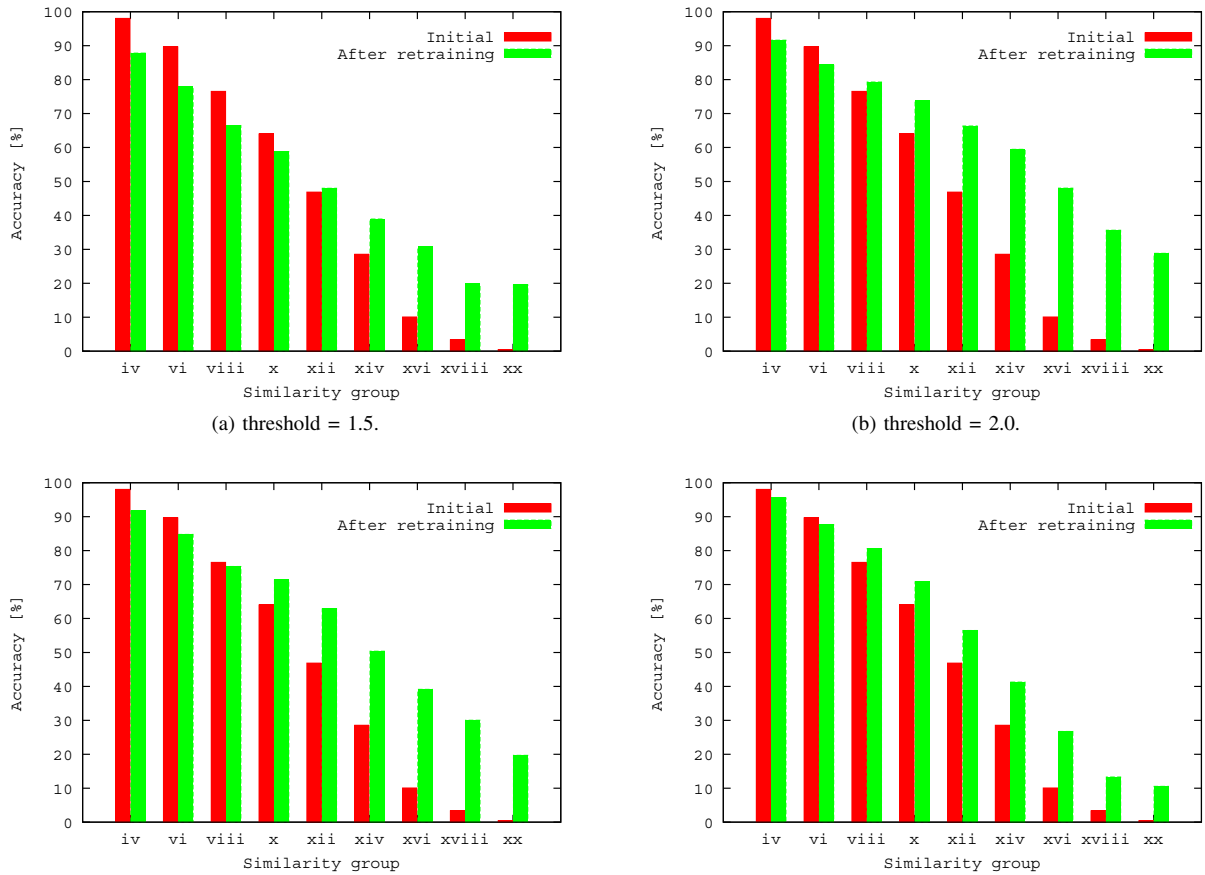
characters keeping high precision. Fig. 8 shows the result in the case that the initial threshold was 2.8. Finally, the threshold was decreased down to 1. This means all unlabeled data were used for retrained. C, M and R are shown in Table II. This result shows that the dynamic thresholding strategy outperformed the fixed thresholding. Since C was high and R was low, the classifier was well trained with a lot of unlabeled data.

VI. CONCLUSION

In the current paper, we proposed a scenario of labeling for large datasets based on the self-corrective recognition algorithm. The method improves the classifier so as to recognize distorted characters having different distribution from labeled data. The strong point of the proposed method is the capability of expanding recognizable distorted characters. Therefore, the method has a possibility to realize a system which can recognize a large variety of characters.

In the experiment 1, we showed that to control the order of unlabeled data enabled to recognize distorted characters which are dissimilar to the labeled datum. In the experiment 2, we selected the unlabeled data using reliability check to deal with real cases. In this paper, we proposed the methods of the fixed and dynamic thresholding. Those results showed the dynamic thresholding was better than the fixed thresholding.

Future work consists of two things. One of them is to cope with characters which can be transformed into the same distribution. For example, 'L' and 'V' can be transformed into the same shape by applying an affine transformation. Therefore, we need to propose a method that similar characters belonging to different classes such as 'L' and 'V' are classified. The other is to apply our method to degradation other than affine distortion in order to verify variously degraded characters in Fig.1 can be recognized.



(a) threshold = 1.5.

(b) threshold = 2.0.

(c) threshold = 2.5.

(d) threshold = 3.0.

Figure 7. Relationship between accuracy and similarity groups when threshold was changed.

ACKNOWLEDGMENT

The authors appreciate Profs. George Nagy and Henry Baird for their advices and encouragement to us. This work was supported in part by CREST project from JST, Foundation of Informational Science Advancement, and the Grant-in-Aid for Young Scientists (B) (21700202) from Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey," *IJDAR*, vol. 7, no. 2+3, pp. 83–104, Jul. 2005.

[2] M. Iwamura, T. Tsuji, and K. Kise, "Memory-based recognition of camera-captured characters," *Proc. DAS2010*, pp. 89–96, Jun. 2010.

[3] R. Nagy, A. Dicker, and K. Meyer-Wegener, "NEOCR: A configurable dataset for natural image text recognition," *Proc. CBDAR2011*, pp. 53–58, Sep. 2011.

[4] K. Wang and S. Belongie, "Word spotting in the wild," *Proc. ECCV2010: Part I*, pp. 591–604, Sep. 2010.

[5] G. Nagy and G. L. Shelton, "Self-corrective character recognition system," *IEEE Trans. Information Theory*, vol. IT-12, pp. 215–222, Apr. 1966.

[6] G. Nagy and H. S. Baird, "A self-correcting 100-font classifier," *Proc. IS&T/SPIE Symp. on Electronic Imaging: Science & Technology*, Feb. 1994.

[7] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, Sep. 2009.

[8] V. Frinken, A. Fischer, and H. Bunke, "Improving handwritten keyword spotting with self-training," *Proc. the 2011 ACM Symposium on Applied Computing*, pp. 840–845, Mar. 2011.

[9] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, Sep. 2006.

[10] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[11] T. Sato, M. Iwamura, and K. Kise, "Fast approximate nearest neighbor search based on improved approximate distance," IEICE Technical Report PRMU2011-67, Sep. 2011, written in Japanese.