A New Warping Technique for Normalizing Likelihood of Multiple Classifiers and its Effectiveness in Combined On-line/Off-line Japanese Character Recognition

OndIej Velek, Stefan Jaeger, Masaki Nakagawa
Tokyo University of Agri. & Tech. 2-24-16 Naka-cho, Koganei-shi, Tokyo 184-8588, Japan
E-mail: velek@hands.ei.tuat.ac.jp, stefan@hands.ei.tuat.ac.jp, nakagawa@cc.tuat.ac.jp

Abstract

We propose a new technique for normalizing likelihood of multiple classifiers prior to their combination. Our technique takes classifier-specific likelihood characteristics into account and maps them to a common, ideal characteristic allowing fair combination under arbitrary combination schemes. For each classifier, a simple warping process aligns the likelihood with the accumulated recognition rate, so that recognition rate becomes a uniformly increasing function of likelihood. For combining normalized likelihood values, we investigate several elementary combination rules, such as sum-rule or max-rule. We achieved a significant performance gain of more than five percent, compared to the best single recognition rate, showing both the effectiveness of our method for classifier combination and the benefit of combining on-line Japanese character recognition with stroke order and stroke number independent off-line recognition. Moreover, our experiments provide additional empirical evidence for the good performance of the sum rule in comparison with other elementary combination rules, as has already been observed by other research groups.

1 Introduction

Exploiting complementary information by combining different classifiers for the same classification problem has been a research field actively pursued in pattern recognition during the recent years. In handwriting recognition, classifier combinations are of particular interest since they allow bridging the gap between on-line and off-line handwriting recognition. An integrated on-line/off-line recognition system can exploit valuable on-line information while off-line data guarantees robustness against stroke order and stroke number variations, which is usually the main disadvantage of on-line systems. The different nature of on-line and off-line data, however, complicates their combination within the classification step itself. In fact, there are currently no convincing approaches for combining on-line and off-line information directly in the classification engine. Hence most approaches combine both types of information either during pre-processing; i.e. feature computation [1-3], or, like this paper, in post-processing [4-6]. Combining on-line and off-line recognition in post-processing basically means combination of two independent on-line and off-line recognizers. Though this requires access to on-line as well as off-line data and requires training of two recognizers, it seems to be the most straightforward way for on-line/off-line combination.

In this paper, we combine several on-line and off-line handwriting recognizers to improve on-line Japanese character recognition rates. The main focus lies on the new normalization technique we present in order to normalize likelihood provided by different recognizers and ensure sound classifier combination.

Section 2 and 3 describe the general theoretical framework of our work reported here. Section 4, 5, and 6 describe respectively the databases, recognizers, and combination schemes we use in our experiments. Section 7 contains the main results of our work, and Section 8 concludes this paper with a general discussion.

2 Comparability of different classifiers

Let A be a classifier that maps an unknown input pattern x to one of m possible classes ($\omega_1, \ldots, \omega_m$), and returns values $a_i = A(x, \omega_i)$ related to the probability that x is a member of class ω_i. For an ideal classifier, each return value a_i corresponds with the true probability of ω_i given x; i.e., the a-posteriori probability $P_i(\omega_i/x)$ with

$0 \le a_i = P_i(\omega_i/x) \le 1$. Accordingly, the k-best candidates are defined by the k-best a-posteriori probabilities among $P_i(\omega_i/x)$, $i = 1, \ldots, m$, from which the best value usually determines the classification result. In real practice, however, the output values a_i can merely be approximations of the correct a-posteriori probabilities. Many classifiers do not even output approximations but only values related to

the a-posteriori probabilities. For instance, very often $a_i = A(x, \omega_i)$ is a distance between an input pattern x and a class ω_i in a high-dimensional space given a specific metric; with $a_i \in [0;\infty]$. In this case, the best candidate is not the maximum value among all a_i, but their minimum; i.e., the class with the shortest distance. Also, the scale of output values is generally unknown. For example, from $a_p = 0.5\ a_r$, one cannot predict that class ω_r is twice as likely as class ω_p.

These inadequacies generally pose no problem for a single recognizer that needs to find the k-best candidates. However, for combining several classifiers $\{A, B, C\ldots\}$ we need some relation among candidates stemming from different classifiers $\{a_i = A(x, \omega_i), b_i = B(x, \omega_i), \ldots\}$ in order to compare them and to select the best class. One necessary, but of course not sufficient, requirement is that all values are within the same range.

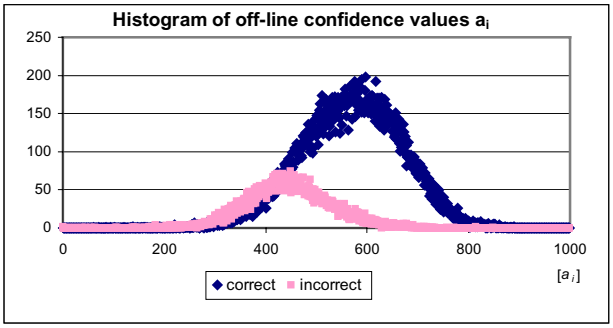
The next section describes our proposed normalization of a-posteriori estimates to ensure a common standard of comparison and a fair combination of multiple classifiers.

3 Characteristic function

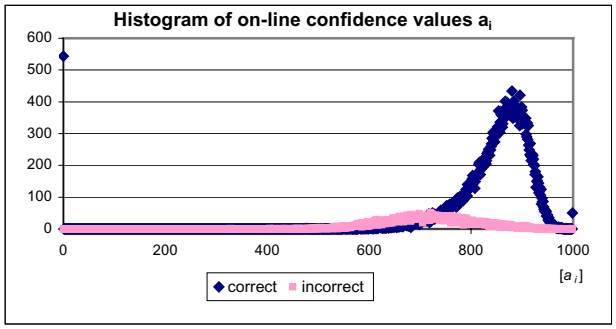
To better describe the output of classifiers, we define a characteristic function for each classifier. In order to do so, we first count the number of correctly and incorrectly recognized samples for each likelihood value: $n_{correct}(i)$ and $n_{incorrect}(i)$ respectively. Graph 1 and Graph 2 show two exemplary histograms of these numbers for on-line and off-line recognition respectively. The ideal classifier returns only correct answers with likelihood values covering the whole range of possible values. Most practical classifiers will, of course, return also incorrect answers. For a single classifier system, only the recognition rate; i.e., the number of correctly recognized patterns $\sum_i n_{correct}(i)$ divided by the number of overall patterns, is of importance. However, for combining multiple classifiers, not only the recognition rate, but also the distribution of $n_{correct}(i)$ and $n_{incorrect}(i)$ is of interest. If some classifiers provide better recognition rates for several sub-intervals of the likelihood range than the best single recognition rate, then we can suppose that by combining multiple classifiers the combined recognition rate will outperform the single best rate.

Graph 1 and Graph 2 illustrate that for the off-line classifier most of the correct output is around a =600, while the on-line classifier has most of its correct answers around a=900. The peak of $n_{correct}(i)$ differs from $n_{incorrect}(i)$ in both classifiers. Also, both classifiers use only a small range of the output interval [0;1000] intensively.

Graph 3 and Graph 4 show the corresponding recognition rates for each likelihood value. Our goal is to normalize the output of each classifier so that it better reflects the actual classifier performance for each output value and



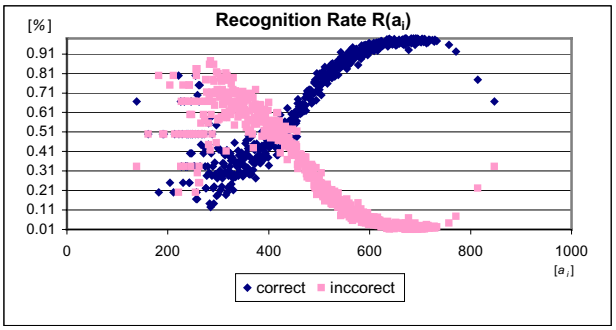
Graph 1: Off-line confidence values.



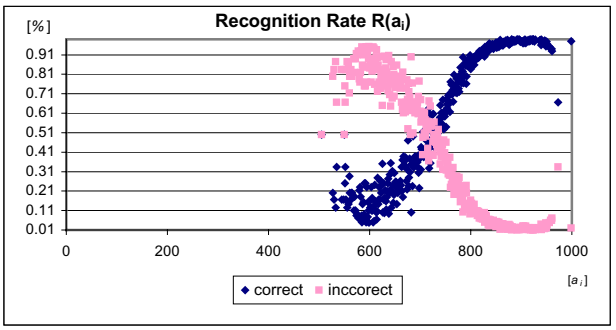
Graph 2: On-line confidence values.

allows direct comparison and/or combination with outputs from other classifiers. The main idea is to turn likelihood into a monotone increasing function depending on the recognition rate. After normalization, the highest output should still define the most likely class of an unknown input pattern x; $a_i > a_j$ should imply that $P(\omega_i/x) > P(\omega_j/x)$. To normalize classifier outputs, we align the likelihood with the accumulated recognition rates by applying a warping process. This process expands and compresses local subintervals of the horizontal x-axis (recognizer output) to better match the practical recognition rates achieved. In Graph 3 and Graph 4 we can see that recognition rate is neither a continuous nor a monotone growing function over classifier output. This is caused mainly by statistical errors due to insufficient training data. In subintervals with enough training data, however, recognition rate may nonetheless be continuous and monotonically increasing.

The presented work is partly motivated by ideas introduced in Reference [7], where likelihood values of a single classifier system are calibrated according to the actual performance of the classifier for a particular likelihood value. In this paper, however, our main intention is not to improve the recognition rate of a single classifier system but to allow a fair combination of multiple classifiers. In fact, the recognition rate of a single classifier system is not affected by our method. Our normalization method does not consider the single recognition rate for a specific output value, but instead, the accumulated recognition rate for an interval $[0;a_i]$, see Graph 5 and Graph 6.



Graph 3: Likelihood-dependent off-line recognition rates.



Graph 4: Likelihood-dependent on-line recognition rates.

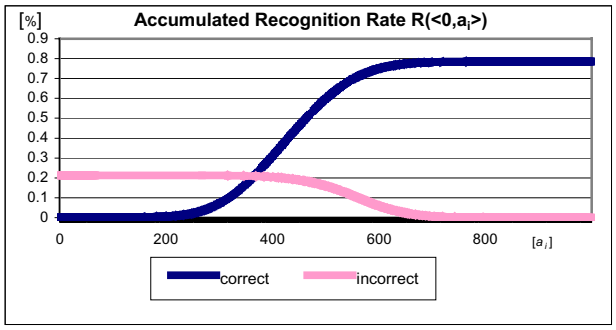
The accumulated recognition rate is a continuous, monotone growing function over classifier output. The error rate depicted in Figure 5 and Figure 6 shows the remaining percentage of misclassified patterns for likelihood values higher than a particular value a_i.

We normalize the output of each single recognizer so that the accumulated probability function R(<0,a_i>) becomes a function proportional to the classifier output. Accordingly, we normalize each classifier output by adding an adjustment: $a'_i = a_i + charf(a_i)$ with

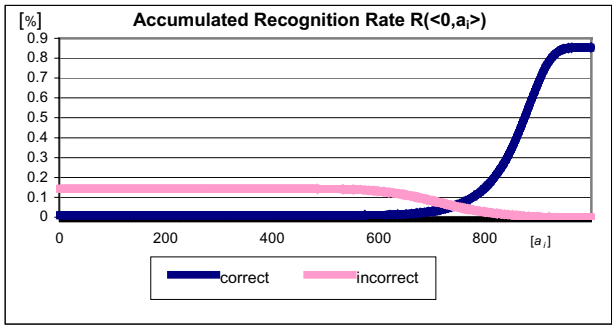
$$charf(a_i) = a_{\max} * R_i - a_i = a_{\max} * \frac{\sum_{k=0}^{i} n_{correct}(k)}{N} - a_i,$$

where $a_{\max}$ is the maximum possible output of a classifier ($a_{\max}$ = 1000 in our experiments), N is the number of overall patterns, and R_i stands for the partially accumulated recognition rate. We call this classifier-specific adjustment the characteristic function [$charf_i$] of a classifier. To compute the [$charf_i$] of a classifier, we need another sample set, which should be independent from the training set.

Accordingly, we used data independent from the training set to ensure a proper evaluation, though Graph 7 shows only minor differences between characteristic functions derived from different data files: While the NTT characteristic - derived from a database collected by

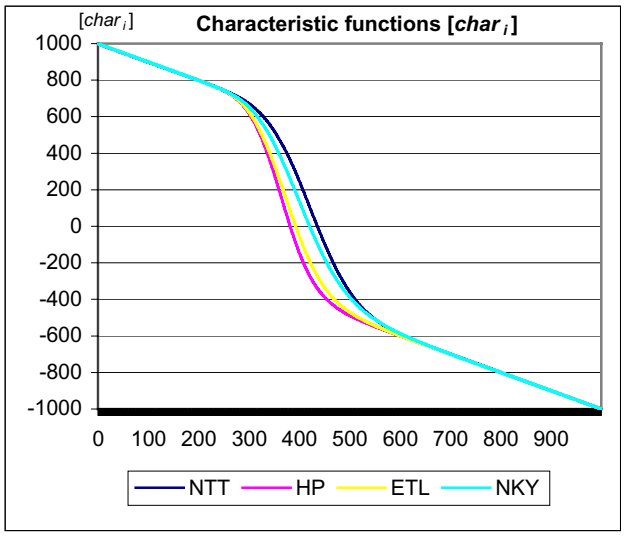


Graph 5: Accumulated off-line recognition rates.

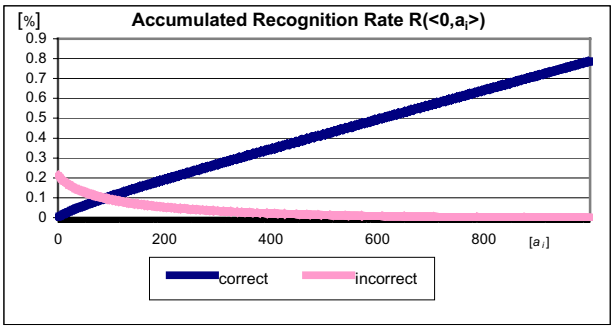


Graph 6: Accumulated on-line recognition rates.

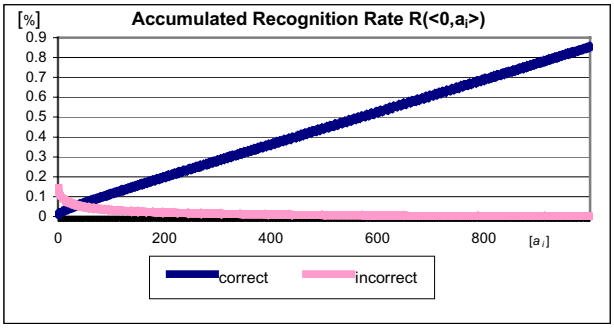
NTT-IT - was computed on the test set, the HP characteristic – HP-JEITA database - is based on the training set. Using the same set for training and normalizing is thus possible, if there is not enough data available for the independent data set. According to the characteristics depicted in Graph 7, and several experiments on other databases, we conclude that a characteristic function depends mostly on the recognizer from which it has been derived, and that it is relatively independent from the set used for normalization.



Graph 7: Characteristic functions for the Off-line recognizer: MQDF256/train set=HP

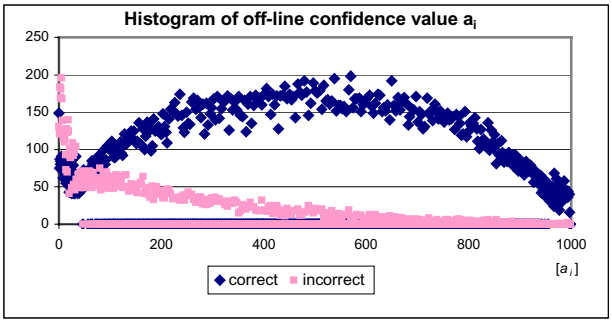


Graph 8: Accumulated off-line recognition rates after normalization.

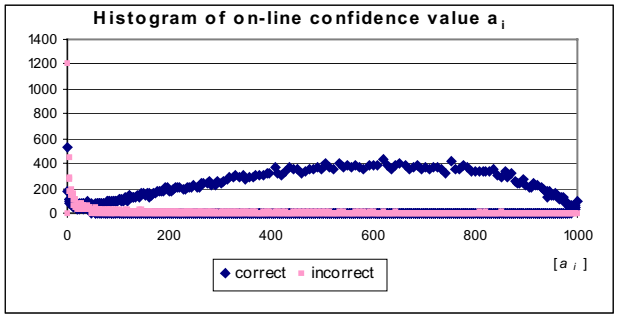


Graph 9: Accumulated on-line recognition rates after normalization.

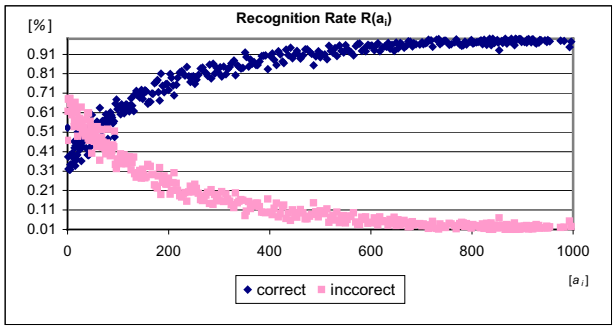
Let us compare graphs 1 to 6 with the corresponding graphs 8 to 13 generated after normalization. Graph 8 and Graph 9 show the normalized accumulated recognition rates for off-line and on-line recognition.



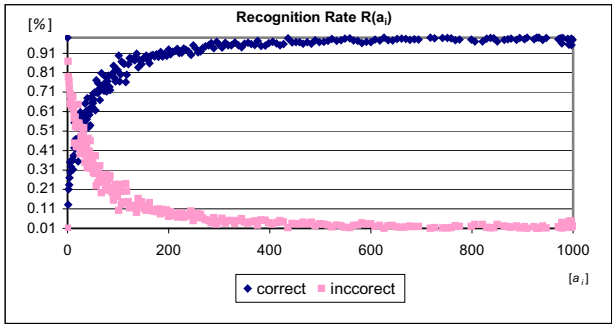
Graph 10: Off-line confidence values after normalization.



Graph 11: On-line confidence values after normalization.



Graph 12: Likelihood-dependent off-line recognition rates after normalization.



Graph 13: Likelihood-dependent on-line recognition rates after normalization.

In Graph 10 and Graph 11, we see that, after normalization, the whole output spectrum is used for correct answers. Moreover, incorrect answers concentrate in the left, low-value part with a peak near zero. Since both off-line and on-line recognizers show the same output behavior after normalization, this provides us with a standard measure for classifier combination

Graph 12 and Graph 13 show that $P_{correct}(a_i) > 50\% > P_{incorrect}(a_i)$ for $a_i \in [100;1000]$. Again, both off-line and on-line classifiers behave similarly here.

4 Databases used in our experiments

We use four handwritten Japanese character databases, two off-line databases and two on-line databases. The off-line databases are ETL9B (4000 writers/3,036 categories/ 607,200 patterns) and HP-JEITA (580/3,214/1,917,480). For on-line recognition we use the Nakayosi database (163/4,438/1,695,689) collected at TUAT [8], and NTT-AT HIT (51/1,237/63,087).

From the on-line databases, Nakayosi and NTT-AT, we generate dual off-line versions using our method for generating realistic Kanji character images from on-line patterns [9]. This method combines on-line patters with a calligraphic stroke shape library, which contains genuine off-line patterns written with different writing tools. Since the artificially generated off-line images are combinations of actual on-line and off-line patterns they look very natural and realistic. Thanks to this duality, the two on-line data-

bases can be trained and tested by off-line as well as on-line methods [10].

We use the NTT-AT database as the test set for our experiments. Accordingly, all recognition rates reported in this paper are based on the NTT-AT database. The NTT-AT database contains data written by elderly people with an average age of 70 years, with the oldest writer being 86 years old. The patterns contained in NTT-AT are casually written, very often with an untypical stroke order. We chose this database because it is difficult to recognize for many recognizers. We also consider it a very good test bed for underpinning our conjecture saying that combining on-line classifiers with stroke-order independent off-line classifiers leads to better overall recognition rates.

5 Classifiers used in our experiments

Altogether we use seven classifiers in our experiments, four off-line and three on-line classifiers. Three of the four off-line recognizers represent each character as a 256-dimensional feature vector. They scale every input pattern to a 64x64 grid by non-linear normalization and smooth it by a connectivity-preserving procedure. Then, they decompose the normalized image into 4 contour sub-patterns, one for each main orientation. Finally, they extract a 64-dimensional feature vector for each contour pattern from their convolution with a blurring mask (Gaussian filter). A pre-classification step precedes the actual final recognition. Pre-classification selects the 50 candidates with the shortest Euclidian distances between the categories' mean vectors and the test pattern. The final classification uses a modified quadratic discriminant function (MQDF2) developed by Kimura [11] from traditional QDF. By training these classifiers with three different training sets, namely HP (with 450 patterns per category), ETL (201), and NKY (163), we obtain three different recognizers. The fourth off-line recognizer was also trained with ETL, but its feature vector is only 196-dimensional; i.e., 49 features for each orientation. For classification, it utilizes the MCE training method (minimum classification error) proposed by Juang and Katagiri [12], which is a widely known LVQ algorithm.

The three on-line recognizers used in our experiments are small variations of the same on-line recognizer described in [13]. We trained each on-line recognizer with the NKY database but changed a parameter controlling the underlying structured character representation for each of them. All on-line recognizers employ a two-step coarse classification based on easily extractable on-line features in the first step and four-directional features in the second step. An efficient elastic matcher exploiting hierarchical pattern representations performs the main classification.

6 Combination strategies

We investigate basically three different combination strategies for combining our on-line and off-line recognizers: majority vote, max-rule, and sum-rule. Majority vote chooses the class receiving the most votes from all classifiers. The max-rule takes the class with the maximum output value among each classifier, while the sum-rule sums up the output for each class and selects the one with the highest sum. These rules are motivated by a paper by Kittler[5], which gives a theoretical explanation for the frequently observed superiority of the sum-rule. In addition to the combination strategies discussed in [5], we apply an additional rule based on majority vote that falls back on the sum rule to resolve ties. This rule actually yields the best recognition rates, even better than the plain sum rule, as shown below.

7 Experimental results

Table 1 shows the n-best recognition rates for each classifier after being tested on the NTT database. The off-line recognizer trained with the HP database achieves the best overall recognition rate, namely 89.07% for the first best candidate.

Table 1: The recognition rate for each individual classifier

	Off-line classifiers				On-line classifiers		
	LVQ192	MQDF/256					
	etl	hp	etl	nky	On1	On2	On3
1	78.78	**89.07**	86.99	86.63	85.67	84.66	84.55
2	86.07	93.69	92.21	92.15	88.90	88.36	88.27
3	88.78	95.09	93.92	93.94	89.89	89.47	89.40
5	90.89	96.24	95.30	95.44	90.53	90.19	90.13
10	91.58	97.16	96.53	96.64	90.65	90.32	90.27

Table 2 shows the recognition rates for combining respectively only on-line classifiers, only off-line classifiers, and both on-line as well as off-line classifiers with four different combination schemes. The first column and second column contain the theoretical minimum and maximum for the majority vote and the max-rule: The first column lists the percentage of test patterns that were recognized correctly by all classifiers, while the second one contains the percentage of patterns that were recognized by at least one classifier. Note that we do not need any transformation of output values here if we combine only on-line or only off-line recognizers since the output values are in the same range and have the same scale. For combining the on-line and off-line recognizers in Table 2, we used a simple linear range mapping. The maximal improvement compared to the best single recognition rate of Table 1 is almost three percent (2.93%). This first result supports our assumption that combined on/off-line recognition yields improved recognition rates.

Table 2: Recognition rates for different combination schemes

	AND	OR	Majority	Max	Sum	Mj-Sum
On-line	82.60	87.37	85.25	85.08	84.01	84.62
Off-line	72.51	93.96	82.49	88.25	87.47	87.48
On+Off	63.91	97.22	90.46	91.75	**92.00**	91.95

Table 3 contains the same recognition rates, but after normalization. It shows a significant improvement of more than two percent compared to Table 2 due to normalized output values. Together with the improvement in Table 1, this sums up to an overall improvement of more than five percent (5.07%). For combined on-line and off-line recognition rates, the sum-rule outperforms the max-rule on both non-normalized and normalized values in Table 2 and Table 3. Anyway, we achieve our best recognition rate with majority vote when we resolve ties with the sum rule. With this method, we reached 94.14% recognition rate as our overall best performance.

Table 3: Recognition rates for different classification schemes after normalization

	AND	OR	Majority	Max	Sum	Mj-Sum
On-line	82.60	87.37	85.25	85.01	84.68	85.38
Off-line	72.51	93.96	82.49	88.79	89.59	89.15
On+Off	63.91	97.22	90.46	91.84	93.77	**94.14**

8 Discussion

The purpose of this paper is twofold – to prove the effectiveness of our technique for normalizing likelihood, and to show the benefit of combining on-line character recognition with off-line recognition. In order to do so, we generated three on-line and four off-line Japanese character recognizers using different training sets. We normalized their output values according to our technique described above, and combined them with several combination schemes. The main idea of our proposed normalization method is to align likelihood with accumulated recognition rate by applying a warping process prior to combination. This method normalizes the likelihood values of multiple classifiers while still allowing application of arbitrary combination schemes.

In summary, we achieved about three percent improvement by combining on-line and off-line recognition and another two percent by normalizing, which sums up to more than five percent overall improvement compared to the best single recognition rate. The sum-rule outperforms the max-rule, which is in accordance with previous observations from other research groups. However, we achieved the best overall performance (94.14% on the NTT data set) by majority vote with ties being resolved with the sum rule. Our proposed method has two important features: It implicitly weights classifiers according to their domain-specific recognition rates, and does not affect the performance of a single classifier system.

Acknowledgment

In our experiments, we used 4,283,456 patterns from four databases altogether. We would like to express our gratitude to all the people who made these patterns available.

References

1. M. Hamanaka, K. Yamada, J. Tsukumo, On-Line Japanese Character Recognition Experiments by an Off-Line Method Based on Normalization-Cooperated Feature Extraction, Proc. 2nd ICDAR (1993) 204-207
2. M. Okamoto, A. Nakamura, K. Yamamoto, Direction-Change Features of Imaginary Strokes for On-Line Handwriting Character Recognition, 14th ICPR (1998) 1747-51
3. S. Jaeger, S. Manke, J. Reichert, A. Waibel, On-Line Handwriting Recognition: The Npen++ Recognizer, IJDAR 3(3) (2001) 169-180
4. H. Kang, K. Kim and J. Kim; A Framework for Probabilistic Combination of Multiple Classifiers at an Abstract Level, EAAI 10 (4) (1997) 379-385
5. J.Kittler, M.Hatef, R.Duin, J.Matas, On Combining Classifiers, *IEEE PAMI 20(3)* (1998) 222-239
6. H. Tanaka, K. Nakajima, K. Ishigaki, K. Akiyama, M. Nakagawa, Hybrid Pen-Input Character Recognition System Based on Integration of On-Line – Off-Line Recognition, 5th ICDAR (1999) 209-212
7. M. Oberländer, German Patent DE 44 36 408 C1 (in German), 1995, on-line available on the website of the German Patent and Trade Mark Office: www.depatisnet.de
8. M. Nakagawa, et al., On-line character pattern database sampled in a sequence of sentences without any writing instructions, *Proc. 4th ICDAR* (1997) 376-380
9. O. Velek, Ch. Liu, M. Nakagawa, Generating Realistic Kanji Character Images from On-line Patterns, *Proc. 6th ICDAR* (2001) 556-560
10. C. Viard-Gaudin, P.M. Lallican, S. Knerr, P. Binter, The IRESTE ON-OFF IRONOFF Handwritten Image Database, 5th ICDAR (1999) 455-458
11. F. Kimura, Modified quadratic discriminant function and the application to Chinese characters, *IEEE PAMI 9(1)* (1987) 149-153
12. B.-H. Juang, S Katagiri, Discriminative learning for minimization error classification, *IEEE Trans. Signal Processing 40(12)* (1992) 761-768
13. M. Nakagawa, K. Akiyama, L.V. Tu, A. Homma, T. Higashiyama, Robust and Highly Customizable Recognition of On-Line Handwritten Japanese Characters, Proc. 13th ICPR (1996), volume III, 269-273