Off-line Handwriting Recognition from Forms

Michael D. Garris, James L. Blue, Gerald T. Candela, Darrin L. Dimmick,
Jon Geist, Patrick J. Grother, Stanley A. Janet, and Charles L. Wilson
National Institute of Standards and Technology,
Building 225, Room A216
Gaithersburg, Maryland 20899
USA

Publishded in IEEE Int. Conf. on Systems, Man, & Cyb. Vol. 3, pp. 2783-2788, Vancouver, October 1995.

ABSTRACT

A public domain optical character recognition (OCR) system has been developed by the National Institute of Standards and Technology (NIST) to provide a baseline of performance on off-line handwriting recognition from forms. The system's source code, training data, and performance assessment tools are all publicly available. The system recognizes the handprint written on Handwriting Sample Forms as distributed on the CD-ROM, *NIST Special Database 19*. The public domain package contains a number of significant contributions to OCR technology, including an optimized Probabilistic Neural Network (PNN) classifier that operates a factor of 20 times faster than traditional software implementations of this algorithm. The modular design of the software makes it useful for training and testing set validation, multiple system voting schemes, and component evaluation and comparison. As an example, the OCR results from two versions of the recognition system (each using a different method of form removal) are presented and analyzed. It is shown that intelligent form removal can improve lowercase recognition by as much as 3%, but this *net* increase in performance is insufficient to understand the impact on the recognition. A method of analysis is provided, whereby the local changes in performance (both gains and losses) of the system are automatically determined. This involves analyzing the distribution statistics of corresponding character confusion pairs between the two systems. As off-line handwriting recognition technology continues to improve, sophisticated analyses like this are necessary to reduce the errors remaining in complex recognition problems.

1. INTRODUCTION

A public domain off-line handwriting recognition system has been developed by the National Institute of Standards and Technology (NIST) [1]. This standard reference system has been designed to evaluate optical character recognition (OCR) from forms. The recognition system is written in C and has been successfully compiled and tested on a host of UNIX workstations including computers manufactured by Digital Equipment Corporation, Hewlett Packard, IBM, Silicon Graphics Incorporated, and Sun Microsystems.[1]

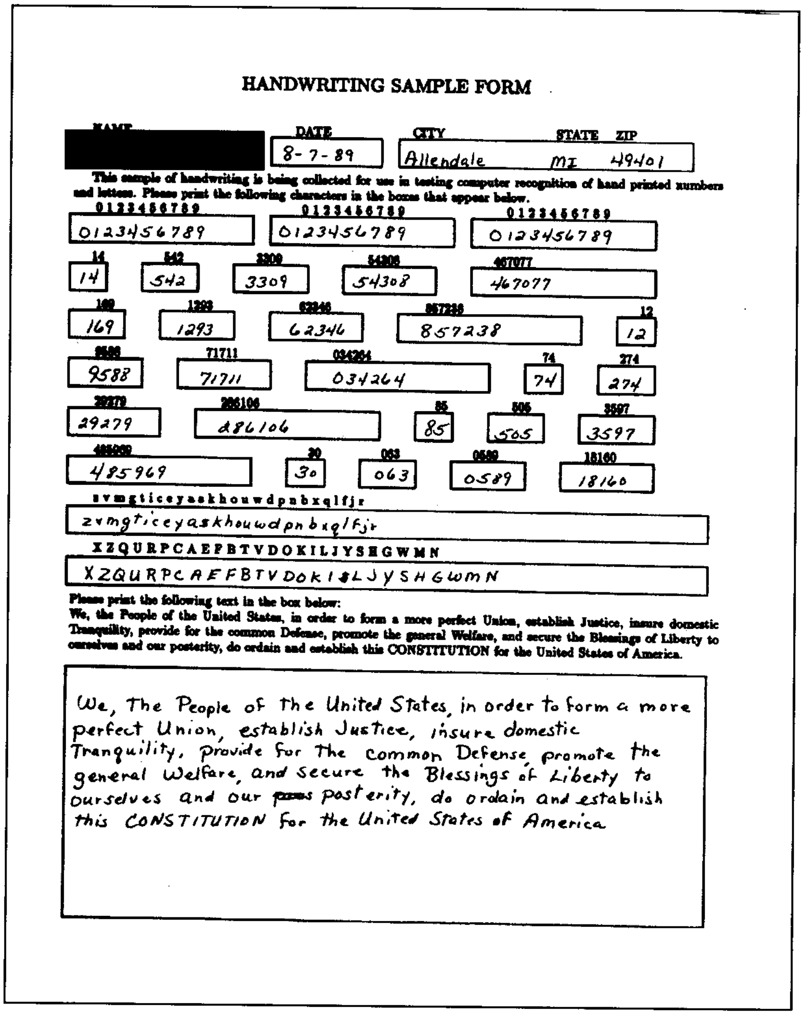
The package has been developed around an *open application*; the system's source code, training data, performance assessment tools, and types of forms processed are all publicly available. Software is provided to retrain the neural network classifier on a completely different set of images (for example, machine print characters), and a robust training set of 168,365 segmented and labelled handprint characters is included. The recognition system processes the Handwriting Sample Forms (HSF) distributed with *NIST Special Database 19* (SD19) [2]. This database contains full page binary images scanned at 12 pixels per millimeter (300 pixels per inch) completed by 3,699 different writers including permanent Census field representatives and high school students. An example of one of the forms is shown in Figure 1.

Section 2 gives a brief overview of the components comprising the NIST public domain off-line handwriting recognition system. Through its modular design, the recognition system can be used in a number of different ways. Its architecture and software organization is completely documented for those interested in technology integration, and any portion of the recognition system may be used without restriction in commercial applications. The system can be used for training and testing set validation. The software can be retrained and tested in a controlled way so that the impact of different training set profiles can be compared, and a training set that provides maximum robustness can be determined. Developers may find that the techniques used are complementary to their own systems, in which case using the public domain system in a voting scheme will improve overall recognition performance.

This paper demonstrates another use for the public domain system, in that it provides developers a baseline of performance on an end-to-end off-line handwriting recognition application, facilitating component testing and comparison. A component may be easily replaced by an alternative algorithm, the same set of input data can be run through the augmented system, and performances between the original and the augmented systems can be compared. This provides the ability to integrate and test new image processing and recognition technologies without having to start from scratch in developing an entire system.

Section 3 lists some global performance statistics achieved by the system. Unfortunately, these statistics tell us very little about the recognition system. There are obviously deficiencies in the system that cause recognition problems. How can they be identified and effectively overcome? If the system is modified to correct these problems, how do we know the new *improvements* to the system do not inadvertently introduce new sources of error? The global performance statistics report only the *net* impact of any modification. There needs to be a way of computing *local* performance statistics (for example, on character confusions) so both the losses and the gains in recognition accuracy can be analyzed. As system developers pursue lower and lower error rates, more sophisticated analyses are needed to understand the performance of the recognition system.

[1] Specific hardware and software products identified in this paper were used to adequately support the development of the technology described in this document. In no case does such identification imply recommendation by the National Institute of Standards and Technology.



HANDWRITING SAMPLE FORM

NAME [redacted]

DATE 8-7-89

CITY Allendale STATE MI ZIP 49401

This sample of handwriting is being collected for use in testing computer recognition of hand printed numbers and letters. Please print the following characters in the boxes that appear below.

0123456789 0123456789	0123456789 0123456789	0123456789 0123456789		
14 14	542 542	3309 3309	54308 54308	467077 467077
169 169	1293 1293	62346 62346	857238 857238	12 12
9588 9588	71711 71711	034264 034264	74 74	274 274
29279 29279	286106 286106	85 85	505 505	3597 3597
485969 485969	30 30	063 063	0589 0589	18160 18160

zvmgticeyaskhouwdpnbxqlfjr

zvmgticeyaskhouwdpnbxqlfjr

XZQURPCAEFBTVDOKILJYSHGWMN

XZQURPCAEFBTVDOKILJYSHGWMN

Please print the following text in the box below:

We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

We, The People of the United States, in order to form a more
perfect Union, establish Justice, insure domestic
Tranquility, provide for The common Defense, promote the
general Welfare, and secure the Blessings of Liberty to
ourselves and our posterity, do ordain and establish
this CONSTITUTION for the United States of America

Figure 1. Completed Handwriting Sample Form from SD19.

With this in mind, a component study using the public domain system along with an automated method for analyzing the character confusions between two systems is presented. The current version of the public domain system uses image subtraction to remove the form from the input image. An older version used histogram projections to locate the boxes on the form. Both techniques cause characters that overlap the form to be clipped, especially descenders on lowercase characters. These techniques also rely heavily on prestored masks and zone templates (the form's geometry) to guide the system. Using these techniques requires strict adherence to form printing and reproduction specifications, and it requires considerable effort when adding new forms or modifying existing forms in the system.

A new method of form removal was developed to improve the accuracy of character recognition while helping reduce the dependence on the geometric details of the form. The method takes a loosely zoned binary subimage of a field and detects and removes all *dominant* horizontal lines, while preserving the character strokes that overlap with the lines [3]. Any field in which the writer is provided a horizontal line to enter a response can be processed by this method. Even if the new form removal method does not improve character recognition accuracy, it is still a significant improvement because the requirement of *a priori* knowledge of the form's geometric details has been greatly reduced.

To evaluate the impact of the new form removal technique on OCR accuracy, the public domain system was modified, and the character confusion matrices of the old and new systems were statistically compared. Section 4 discusses the statistical method and the recognition results are analyzed. This demonstrates one of the many ways the public domain system can be used to evaluate OCR technology.

2. SYSTEM COMPONENTS

The functional architecture of the off-line handwriting recognition system is illustrated in Figure 2. This diagram represents the processing of handprinted fields that contain all digits, fields that contain all lowercase or all uppercase letters, and fields that contain a text paragraph of mixed upper and lowercase letters. As can be seen from the figure, there is a large overlap in the functional components used across these different types of fields.

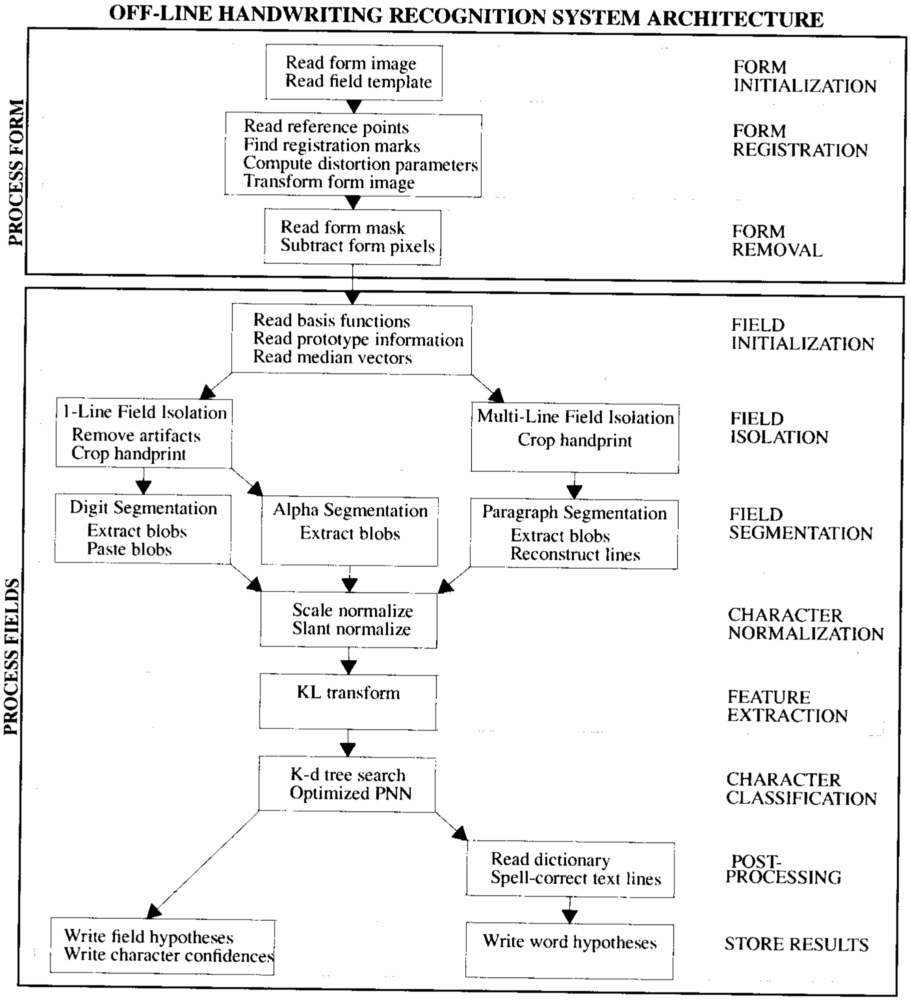


Figure 2. Functional architecture of the NIST public domain OCR system.

The system uses histogram projections to locate registration points within an image of an HSF form. These points are aligned to a set of reference registration points using a Linear Least Squares fit, and the image is transformed, removing any global distortions in rotation, translation, scale. The public domain system currently uses a blank form as a mask, erasing any pixels that are a part of the form (its boxes and instructions). Handwriting within each field is segmented using connected component labelling, and each component is size and slant normalized. The Karhunen Loève (KL) transform [4] is computed on each segmented character image, and a vector of KL coefficients are produced. These coefficients form a feature vector that is classified using a Probabilistic Neural Network (PNN) [5]. Our k-d tree implementation [6] classifies a factor of 20 times faster than more conventional software implementations of the PNN algorithm. The recognition system stores the character's assigned identity along with a confidence value. A complete description of each component is provided in the public domain system's documentation [1].

3. GLOBAL PERFORMANCE STATISTICS

NIST has developed a recognition system testing methodology that has been implemented as the NIST recognition system scoring package [7],[8]. The scoring package has been developed to measure the performance of character recognition systems and automated form processing systems by reconciling the system's *hypothesized* field contents (what the system read) to *reference* field contents (what is really in the field).

In a sample of the first 500 writers from SD19, the standard reference system achieves a character output accuracy (with no rejection) of 92.9% out of a possible 63,830 handprinted digits. The system achieves a character output accuracy of 75.3% from 12,766 lowercase letters and 84.5% out of a possible 12,766 uppercase letters. The system recognizes 79.1% of 13,748 numeric fields completely correct, and the system correctly recognizes 60.5% of the 25,532 words in the Constitution fields (using a limited dictionary of 38 words).

4. COMPONENT TEST AND STATISTICAL EVALUATION

This section presents a component study as an example of how one might use the NIST public domain recognition system to evaluate off-line handwriting recognition technology. A new method for line detection and removal was developed by NIST, and once performing satisfactorily on a small number of test cases, a large test was desired to evaluate the overall impact of the new method. It was noted that the new method would have the greatest influence on lowercase characters with descenders, such as g, j, p, q, and y. These characters, when handprinted, frequently intersect and pass through the line along which they are written.

A test consisting of the first 2,100 randomly-ordered lowercase alphabets in SD19 was chosen to compare two versions of the recognition system. The old version of the recognition system uses a form removal technique that chops off (or disconnects) these descenders causing inter-character ambiguities that decrease the performance of the system by inflating the number of substitutional errors. For example, a *q* with its descender chopped off looks like an *a*.

The new version of form removal uses the Hough line transform [9] to automatically detect all the *dominant* lines in the image. The lines are intelligently removed while simultaneously preserving overlapping character strokes by computing line width statistics and keying off of certain visual cues [3]. An example from using this technique is shown in Figure 3. All the other components (connected component character segmentation, size and slant normalization, KL feature extraction, and optimized PNN classification) are the same between the two systems.

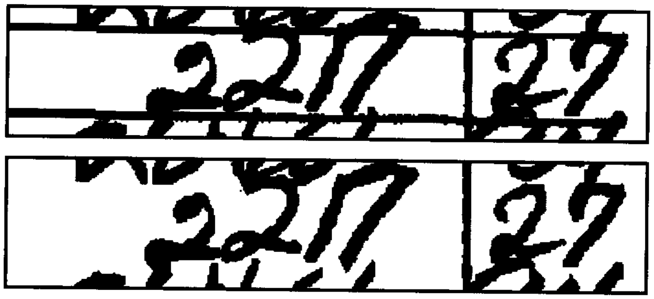


Figure 3. Results of intelligent (horizontal) line removal.

To assess the overall impact of the new form removal technique, the OCR results from the old and new systems were computed and then scored using the NIST scoring package. Global performance statistics were automatically calculated and reported. The old system achieved a character output accuracy of 76.9% out of a possible 54,340 lowercase letters; the test actually consisted of 2,090 lowercase fields. The new system achieved a character output accuracy of 80.1% on the same fields. Using the new line detection and removal techniques on the lowercase alphabet fields improved the recognition about 3%. This is reasonable, as there are only 5 out of the possible 26 letters in the lowercase alphabet that have significant descenders, and from observation, we know these 5 letters are not always written such that they overlap the line on the form. If on average one of the 5 letters touches the line during the printing of the alphabet, then this would account for 3.8% (1/26) of the letters. We also know this would be an upper bound (given our assumptions), because not all of the touching characters would cause an incorrect classification. So, a 3% improvement in recognition is *reasonable*.

Statistical Analysis of Confusion Matrices

The 3% increase in recognition accuracy as a result of using intelligent line removal is in fact an improvement, but this global performance statistic tells us very little about *how* the recognition was really impacted. Global performance measures are helpful, but in practice their usefulness is limited. There needs to be a way of computing *local* performance statistics on both the gains and the losses in recognition accuracy. For example in this study, we knew the improvements to the recognition system should have their greatest impact on lowercase letters (particularly, those with descenders), and we just happened to have a database of lowercase characters on which to test. We have shown a 3% improvement in overall lowercase recognition accuracy, and yet have the changes to the recognition system inadvertently caused some recognition problems? If so, what are these degradations? Is there something that can be done to minimize them so accuracy continues to improve towards some upper bound?

A technique for determining the significant difference in local performance statistics between two OCR systems has been developed. The method analyzes the confusion matrices generated by the two recognition systems. The testing set is partitioned into *n* equal subsets (in this study, $n = 10$). The substitution errors incurred by each system on each testing subset are collected into separate confusion matrices. This produces a set of *n* confusion matrices for the first system, and a set of *n* confusion matrices for the second system. The confusion matrices from each system provide *n* samples for each possible confusion pair. These samples are used to compute distribution statistics (a mean and a standard deviation). A mean and standard deviation pair is computed for each cell in the confusion matrix for the first system (μ_1, σ_1) and for each cell in the confusion matrix for the second system (μ_2, σ_2).

Given the distribution statistics of each corresponding confusion pair between the two systems, a Student's *t* test can be used to determine how similar the distributions are to each other [10]. The difference between two distributions is measured as

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{n}}} \qquad (1)$$

which is in units of root mean square standard error. Given the normalized distance *t*, a probability ρ is derived either numerically or via table look-up for each cell in the confusion matrix [11]. This is the probability that $|t|$ could be at least this large by chance, so the smaller the value of ρ, the less likely the two distributions are the same. Low values of ρ indicate the difference between the two systems for a particular confusion pair is significant. The table in Figure 4 was produced by thresholding ρ at 2%, in which case we are 98% sure the difference did *not* happen by random chance. The confusion pairs with ρ less than 2% were sorted on their corresponding value of *t* and reported in the table.

The first two columns in the table report each confusion pair determined to be statistically different. The first character, labeled (R)ight, is the reference character the system should have recognized. The second character, labeled (W)rong, is the hypothesis character the system incorrectly assigned to the letter. For example, the first line in the table is reporting statistics on the system incorrectly classifying p's as o's.

PAIR		SYSTEM 1		SYSTEM 2		MEAN Δ	STUDENT'S t	
R	W	μ_1	σ_1	μ_2	σ_2	$\mu_1 - \mu_2$	t	$\rho \times 10^2$
p	o	12.7	1.4	1.8	0.5	10.9	23.7	0.0
g	a	25.6	4.1	6.2	1.6	19.4	13.9	0.0
q	a	33.8	4.3	12.6	3.4	21.2	12.3	0.0
j	l	27.6	3.9	11.3	2.9	16.3	10.6	0.0
y	v	27.6	5.1	7.6	3.7	20.0	10.0	0.0
q	o	8.3	2.0	2.5	1.0	5.8	8.2	0.0
y	u	10.3	2.5	3.2	1.2	7.1	8.2	0.0
b	h	9.7	2.4	3.3	1.4	6.4	7.4	0.0
j	i	17.0	2.6	9.9	2.9	7.1	5.7	0.0
p	n	4.8	2.2	1.1	0.5	3.7	5.2	0.0
p	b	3.1	1.4	0.8	0.3	2.3	5.0	0.0
q	u	1.1	0.5	0.4	0.0	0.7	4.7	0.1
g	n	2.3	1.4	0.5	0.0	1.8	3.9	0.3
w	b	0.6	0.3	0.2	0.0	0.4	3.8	0.4
g	o	6.6	2.2	3.6	1.3	3.0	3.7	0.2
j	k	0.7	0.5	0.2	0.0	0.5	3.4	0.8
y	m	0.6	0.5	0.1	0.0	0.5	3.4	0.8
o	n	3.8	1.2	2.0	1.2	1.8	3.2	0.5
p	d	1.9	1.4	0.6	0.0	1.3	2.9	1.7
e	p	4.0	1.5	2.5	0.7	1.5	2.9	1.3
p	f	2.6	0.9	1.8	0.0	0.8	2.9	1.8
j	e	0.4	0.3	0.1	0.0	0.3	2.9	1.9
p	h	0.3	0.3	0.0	0.0	0.3	2.9	1.9
j	c	0.5	0.3	0.2	0.0	0.3	2.9	1.9
y	f	0.4	0.3	0.1	0.0	0.3	2.9	1.9
c	j	0.4	0.3	0.1	0.0	0.3	2.9	1.9
q	d	1.2	0.6	0.6	0.3	0.6	2.9	1.3
q	w	0.6	0.3	0.3	0.0	0.3	2.9	1.9
j	n	0.7	0.0	0.4	0.3	0.3	2.9	1.9
f	s	1.4	0.5	0.9	0.3	0.5	2.7	1.4
g	u	0.4	0.3	0.9	0.5	-0.5	-2.7	1.4
g	m	0.2	0.0	0.5	0.3	-0.3	-2.9	1.9
q	x	0.3	0.0	0.6	0.3	-0.3	-2.9	1.9
w	x	0.4	0.0	0.7	0.3	-0.3	-2.9	1.9
q	g	18.9	5.2	25.1	4.3	-6.2	-2.9	1.0
y	q	1.7	0.9	2.9	0.9	-1.2	-3.0	0.7
y	g	2.7	1.4	5.4	2.2	-2.7	-3.3	0.5
d	t	0.2	0.0	0.6	0.3	-0.4	-3.8	0.4
b	o	0.9	0.5	1.6	0.3	-0.7	-3.8	0.1
j	v	2.4	0.8	3.9	0.9	-1.5	-4.0	0.1
r	g	0.2	0.0	0.7	0.3	-0.5	-4.7	0.1

Figure 4. Statistical analysis reporting all significant (98% confident) changes in confusion errors between the old system that chopped off characters and the new system that removed lines while preserving character stokes.

The next two columns, labeled SYSTEM 1, contain the performance statistics from the old recognition system. The first of these columns lists the average number of errors μ_1 incurred for the corresponding confusion pair across the 10 test partitions. The second column lists the standard deviations σ_1 associated with these errors. The second system's error statistics are listed in the next two columns labeled SYSTEM 2. These are the errors from using the new line detection and removal techniques.

The column in the table, labeled MEAN Δ, is the difference in the mean accumulated errors between the two systems. With p's classified as o's, there were on average 10.9 fewer errors made by the second recognition system. Keep in mind there were 2,090 lowercase alphabets in the test, and these fields were divided into 10 equal partitions. As a result, there were 209 examples of each character in each partition, so 10.9 fewer p's called o's is an average decrease of 5.2% (10.9 / 209). The largest significant improvement was in q's called a's, where there was a 10.1% decrease in these types of errors.

The last two columns in the table list the results of the Student's t test. The first column lists the normalized distance t between the first and second systems' distribution of errors for the corresponding confusion pair. The second column lists the corresponding value of ρ, the probability that the measured difference between the two distributions occurred by chance.

The table is divided vertically in two parts. The top portion lists all those confusion pairs in which the new system improved, making fewer errors. This is represented in positive mean Δ's and in positive values of t. The bottom portion of the table lists all those confusion pairs in which the new system did worse than the old system. In this case, the mean Δ's and t values are negative. These represent the significant trade-offs of introducing the new form removal method into the recognition system. Despite the losses in performance, there are considerably more improvements with the new line detection and removal techniques, and their net impact on performance greatly outweighs the losses (3%).

A closer look at the table shows the statistics are well behaved. Looking at the top and bottom of the column of ρ values, one can observe that at the ends, ρ is very low so the confidence is very high that the difference between the two systems is significant. As we move to the middle of the table, the absolute values of t decrease and the values of ρ increase. This should occur because, as a general rule, the closer two distributions are to each other the more likely they come from the same underlying distribution. It is interesting to also note that, in general, the mean Δ's follow this same trend. However, there are some exceptions giving support to the fact that, without some measure of statistical significance, a single measured sample may be misleading.

In light of these observations, the confusion pairs at each end of the table are statistically most significant. Using the intelligent line detection and removal techniques, the most significant improvements occur with confusion pairs (p, o), (g, a), (q, a), (j, l), (y, v), (q, o), and (y, u). The first (reference) character in these pairs all have descenders, and if you remove the descenders you are left with a partial character that closely resembles the second character in each pair. As we expected, the intelligent removal of lines does significantly improve recognition, especially lowercase characters with descenders. The statistical analysis of the confusion matrices was used to assess the impact of intelligent form removal on off-line handwriting recognition, and by the predictable nature of the experiment, the OCR test results have served to validate the statistical method.

On the other end of the table, significant decreases in performance have occurred with (r, g), (j, v), and up the list a bit further (q, g). These are cases where introducing the new method of line removal actually increased confusion among specific pairs of characters. The new line removal technique attempts to preserve as much of the handwritten character as possible. In so doing, some of the lowercase characters now have considerably longer descenders, which in the old system were clipped. The consistent chopping off of descenders, while bad for classification in general, actually avoids certain inter-character ambiguities. By preserving the descenders, these naturally occurring ambiguities are reintroduced into the recognition problem and errors among these confusable characters increase. For example, j's that were once cut off and highly confusable as l's now have their descenders preserved, so at times their tails curve up sufficiently to be confusable with handprinted v's.

Some of the other confusion pairs, having a negative impact on the new recognition system, are harder to explain. It is our experience that the size normalization used in the preprocessing of the character image can cause unexpected, yet consistent, patterns. For example, as the descenders on handprinted g's become longer and longer, the dominance of the top loop on the letter, after size normalization, becomes smaller and smaller to the point that it closes. At this point, the normalized character image does become confusable with some handprinted r's. The point is there was no way to predict what the impact would be when reintroducing these naturally occurring ambiguities into the system. Through this statistical analysis, the significant system degradations (in addition to the improvements) have been automatically identified.

5. CONCLUSIONS

This paper has discussed the usefulness of the NIST public domain form-based handprint recognition system for evaluating off-line handwriting recognition. The modular design of the system makes it useful for OCR benchmarking, off-line training and testing set validation, and multiple system voting schemes. Another important use of the system is to provide a baseline of performance on an end-to-end application for system developers, facilitating component testing and comparison.

A practical example of component testing was presented, where two versions of the public domain system were compared. Each version used a different form removal method, and it was shown the new intelligent line removal technique did in fact improve the recognition of lowercase letters by 3%. To understand this *net* improvement, a statistical analysis of the two systems' confusion matrices was conducted. This analysis involved computing distribution statistics for each confusion pair, and then using a Student's t test to determine which of the corresponding confusion pair distributions between the two systems were significantly different. As the error rates in recognition systems continue to decrease, it becomes increasingly important to apply sophisticated analyses like this in order to understand the performance of the recognition system.

Distributions of the standard reference recognition system can be obtained free of charge on an ISO-9660 format CD-ROM by sending a letter of request to the primary author. Any portion of this system may be used without restrictions. The system software was produced by NIST, an agency of the U.S. government, and by statute is not subject to copyright in the United States. Recipients of the standard reference recognition system assume all responsibilities associated with its operation, modification, and maintenance.

6. REFERENCES

[1] M. D. Garris, J. L. Blue, G. T. Candela, D. L. Dimmick, J. Geist, P. J. Grother, S. A. Janet, and C. L. Wilson, "NIST Form-Based Handprint Recognition System," NIST Internal Report 5469 and CDROM, July 1994.

[2] P. J. Grother, "Handprinted Forms and Character Database, *NIST Special Database 19*," Technical Report and CD-ROM, National Institute of Standards and Technology, March 1995.

[3] M. D. Garris, "Method and Evaluation of Character Stroke Preservation on Handprint Recognition," NIST Internal Report, *to be published.*

[4] P. J. Grother, "Karhunen Loève Feature Extraction for Neural Handwritten Character Recognition," In *Proceedings: Applications of Artificial Neural Networks III*, Vol. 1709, SPIE, Orlando, April 1992, pp. 155-166.

[5] D. F. Specht, "Probabilistic Neural Networks," *Neural Networks*, Vol. 3(1), 1990, pp. 109-119.

[6] P. J. Grother, G. T. Candela, and J. L. Blue, "Fast Implementations of Nearest-Neighbor Classifiers," NIST Internal Report, National Institute of Standards and Technology, *to be published.*

[7] M. D. Garris, "Methods for Evaluating the Performance of Systems Intended to Recognize Characters from Image Data Scanned from Forms," NIST Internal Report NISTIR 5129, February 1993.

[8] M. D. Garris and S. A. Janet, "NIST Scoring Package User's Guide, Release 1.0, *Special Software 1*," NIST Internal Report 4950 and CDROM, October 1992.

[9] P. V. C. Hough, "Methods and Means for Recognizing Complex Patterns," U.S. Patent 3,069,654, 1962.

[10] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, 8th Edition, pp. 53-57, Iowa State University Press, Ames Iowa, 1989.

[11] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes, The Art of Scientific Computing (FORTRAN Version)*, pp. 466-467, Cambridge University Press, Cambridge, 1989.