NISTIR 5105

Machine-Assisted Human Classification of Segmented Characters for OCR Testing and Training

R. Allen Wilkinson
Michael D. Garris
Jon Geist

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Computer Systems Laboratory
Advanced Systems Division
Gaithersburg, MD 20899

NIST

NISTIR 5105

Machine-Assisted Human Classification of Segmented Characters for OCR Testing and Training

R. Allen Wilkinson
Michael D. Garris
Jon Geist

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Computer Systems Laboratory
Advanced Systems Division
Gaithersburg, MD 20899

December 1992

U.S. DEPARTMENT OF COMMERCE
Barbara Hackman Franklin, Secretary

TECHNOLOGY ADMINISTRATION
Robert M. White, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

Machine-Assisted Human Classification of Segmented Characters For OCR Testing and Training

R. Allen Wilkinson, Michael D. Garris, and Jon Geist,
National Institute of Standards and Technology,
Gaithersburg, MD 20899

Abstract

NIST needed a large set of segmented characters for use as a test set for the First Census Optical Character Recognition (OCR) Systems Conference. A machine-assisted human classification system was developed to expedite the process. The testing set consists of 58,000 digits and 10,000 upper and lower case characters entered on forms by high school students and is distributed as Testdata 1. A machine system was able to recognize a majority of the characters but all system decisions required human verification. The NIST recognition system was augmented with human verification to produce the testing database. This augmented system consists of several parts: the recognition system; a checking pass; a correcting pass; and a clean up pass. The recognition system was developed at NIST. The checking pass verifies that an image is in the correct class. The correcting pass allows classes to be changed. The clean-up pass forces the system to stabilize by accepting images with verified classifications while rejecting all others.

In developing the testing set we discovered that segmented characters can be ambiguous even without context bias. This ambiguity can be caused by over-segmentation or by the way a person writes. For instance, it is possible to create four ambiguous characters to represent all ten digits. This means that a quoted accuracy rate for a set of segmented characters is meaningless without reference to human performance on the same set of characters. This is different from the case of isolated fields where most of the ambiguity can be overcome by using context which is available in the non-segmented image. For instance, in the First Census OCR conference, one system achieved a forced decision error rate for digits of 1.6% while 21 other systems achieved error rates of 3.2% to 5.1%. This statement cannot be evaluated until the same set of characters is presented one at a time and without context to humans and then the performance is measured.

1 Introduction

It was required that NIST develop a large set of segmented characters for using in testing OCR systems for the First Census OCR Systems Conference [1]. A training set of over 300,000 segmented characters was already available. The training set was produced from

the 2100 forms in Special Database 1 (SD1) [2] with a potential of 382,200 characters and is distributed as Special Database 3 (SD3) [3]. The testing set was produced from 500 different forms with a potential of 91,000 characters and is distributed as NIST TestData 1 (TD1) [4].

Each form is segmented into field images and then into character images. The character images were labelled with reference classifications. At the time the training set was produced no recognition system existed at NIST, i.e., all classification had been done by one laboratory worker. This process turned out to be very slow (approximately 6 months) and tedious. The procedure used is fully described in section 2.

When it was time to produce the testing set, a complete hand printed character recognition system was available. If the recognition system was used to assign reference classifications to the test images, then the accuracy of the test set classifications would be poor, about 90% correct. A much higher accuracy was needed, so human interaction became necessary. Machine-assisted human classification of segmented characters can be faster than human-only classification. This process is described in section 3. One drawback with both methods is that some classifications are viewer-dependent due to character ambiguities. This effect is described in section 4.

2 Creating Special Database 3

The creation of SD3 was a long and time consuming process. The 30 numeric and alphabetic fields, with a potential of 182 character images each, on the 2100 forms in SD1 were first isolated and then segmented using a histogram segmenter which incorrectly segmented 11.2% of the characters. No machine recognition was used in this procedure. The images were assigned a classification by matching the order in which they were segmented to a reference string. Since the segmenter could over-segment or under-segment the fields, the reference string-based classification was often incorrect. Therefore, images had to be verified for correct classification by a human before they could be distributed. This was done with a slow view-and-verify program.

2.1 Segmenter

The segmenter used in creating SD3 was the third-generation, knowledge-based, histogram segmenter of Wilkinson [5]. The segmenter uses knowledge about hand-print to improve the accuracy of segmentation. This approach has several restrictions. These restrictions include: inability to separate connected characters; inability to find all segmentation cuts; and over-segmenting characters with a vertical void.

2.2 Verification

The verification process for SD3 was very slow and tedious because each character was individually displayed along with the class assignment derived from the order of segmentation

as compared to the reference string. If the class was correct for the image, then the image was saved with that class. If the image was of another class, then the image was reclassified and saved. If the image did not belong to any of the classes available, then it was rejected completely. Time constraints prevented the use of this process in creating the test data required for the First Census OCR Conference.

3 Creating Testdata 1

For the First Census OCR Conference a testing database of about one quarter of the size of SD3 was desired. TD1 was created for this purpose and a very different and more efficient approach was used in its creation. This approach took advantage of the NIST character recognition system [6]. Figure 1 shows the overall system flow for the segmentation, recognition and verification process used in creating TD1.

The character recognition system is used by the *Run Recognition System* module. The *Build Tree* module prepares the output from the recognition system for use in a post recognition human verification scheme. The checking pass consists of three modules: *Build Checking Sessions*, *Check Sessions*, and *Update Tree.* This is followed by the correcting pass with three modules of similar names: *Build Correcting Sessions*, *Correct Sessions*, and *Update Tree.* The next module terminates the human verification process. When the operator is satisfied with the results of running the six previous modules, then the system exits to the sanity pass. The sanity pass is really a complete run of the checking pass where items that are either accepted or rejected, which forces all items to be put in a terminal state. All of the accepted images and their verified classifications are included in the test database.

3.1 Character segmentation and Recognition

The segmentation and recognition of individual characters was done using a character recognition system developed at NIST [6]. The system isolates a field with data, segments it, and then recognizes the segmented images. These images and the recognition results still rquire verification by a human observer. This verification allows the quality and accuracy of the data to be maintained at the high level needed to preserve the integrity of any test conducted with the data.

3.2 Verification

The verification system uses four states, *accepted*, *rejected*, *to-be-checked*, and *to-be-corrected.* The *accepted* and *rejected* states are terminal while the *to-be-checked* and *to-be-corrected* states are intermediate. Each image being verified must be in one of these states. Upon completion of the recognition process, all images are either *accepted* or *rejected.* Before any *checking* or *correcting* sessions can be done, the items in the *accepted* state must be moved to the *to-be-checked* state and the items in the *rejected* state to the *to-be-corrected* state.

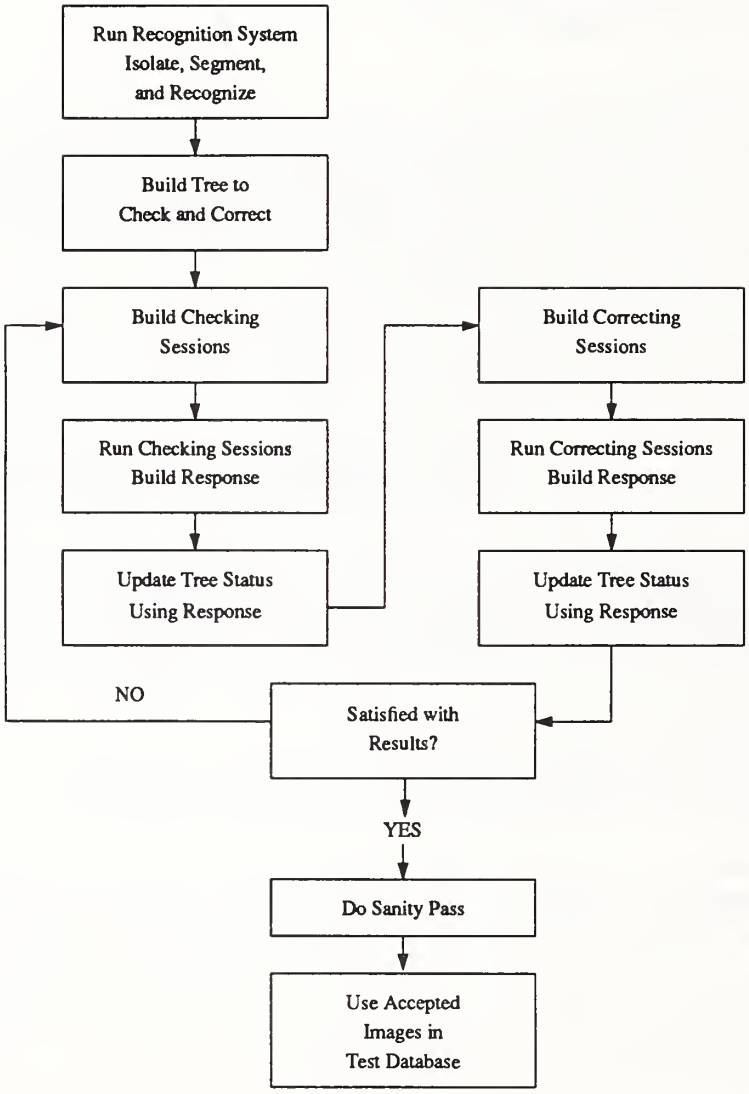

Figure 1: Flow chart for database verification/creation

Verification begins by running a *checking* session which is described in greater detail in section 3.2.1. The *checking* session moves items in the *to-be-checked* state to the *accepted* state if correct and to the *to-be-corrected* state if not.

After the *checking* sessions are all finished, the *correcting* sessions begin. The *correcting* sessions are described in detail in section 3.2.2. A *correcting* session moves items from the *to be corrected* state to the *to be checked* state if the item is recognizable and to the *rejected* state if it is not.

A number of different individuals were involved in both the checking and correcting stages of the creation of TD1. Therefore, it is possible to have items oscillate between the *to-be-checked* and *to-be-corrected* states. This is remedied by a sanity pass which checks everything in the *to-be-checked* state and then moves all items in the *to-be-corrected* state to the *rejected* state. This will get everything into either the *accepted* or *rejected* state as desired.

3.2.1 The Checking Pass

Sessions are built for the *checking* pass. Each session consists of up to 1024 images of the same class. This class is assigned initially by the recognition system and subsequently by the human verifier. After the sessions are built, all images in the session are displayed at the same time. Those items that do not belong to the given class are marked. These items are moved to the *to-be-corrected* state while all other items are moved to the *accepted* state. Once an item is in the *accepted* state it cannot be moved out. The *checking* pass is a yes-no process. Images can either be accepted or rejected. For an image to be reclassified, it must first be rejected in the *checking* pass, then in the *correcting* pass it can be reclassified. After the image is reclassified, it will be returned to the *checking* pass to check if the reclassification is correct. It is possible to cause the image to oscillate between these two passes. This will occur if the reclassification continues to be incorrect.

3.2.2 The Correcting Pass

Sessions are also built for the *correcting* pass. These sessions consist of up to 200 images of the same class. Each image in a session is viewed separately from the others. The only options the verifier has are to accept, reject, or reclassify the image. Accepting the image means that the image belongs to the class being viewed, and that it is moved to the *to-be-checked* state. A rejected image is one that does not belong in any of the classifications allowed. For this work, only alphanumerics were allowed. A rejected image is moved to the *rejected* state and cannot be moved out. Images receiving new classifications from the human verifier are moved to the *to-be-checked* state. The *correcting* pass allows images to be reclassified while the *checking* pass does not.

4 Ambiguity of Segmented Characters

4.1 Writer Ambiguity

Without considering the possibility of a more general definition, writer ambiguity for uniform pixel size, binary, handprint images is defined as follows: If two images that can be made identical through translation, rotation, or shear transformations do not depict the same character according to the intention of the person or persons who printed them, then they are writer ambiguous over the set of writers who printed them and over any writer superset of that set. A set of images of writer-ambiguous characters is a writer-ambiguous set. In order for an OCR system to correctly classify every image in a writer-ambiguous set, it must assign different classifications to the same images at least once. This requires either information about the images beyond that contained in the individual images or luck. Neither of these possibilities need be considered in the context of a scientific study of the OCR of segmented handprint.

4.2 Reader Ambiguity

It can be argued from a philosophical point of view that reader ambiguity is irrelevant to the OCR of segmented handprint, since all that should matter is what the writer intended, and not how some reader interprets it. However, there are practical reasons associated with the use of machine-assisted classification that make it necessary to consider reader ambiguity as well.

First, writers sometimes interchange characters when copying them from a template as was done in the generation of TD1, and sometimes they write incorrect or ambiguous characters on purpose for reasons that have to do with being human. This means that we cannot rely on the writer-assigned classifications, which is the reason that human checking is needed.

At least two types of reader ambiguity that are likely to occur during the machine-assisted human checking carried out on TD1 can be identified. The first type is called context ambiguity. Consider the *checking* pass described above. A screen containing 1024 images of characters that had all been classified, for example, as *fours* by the NIST OCR system was viewed by a human. It is possible that some of the characters on the screen would have been classified as *nines* if viewed in isolation by the same person, but were passed as *fours* due to the context bias induced by the screen full of *fours*. Figure 2 is an example of this.

The second type is called of reader ambiguity subjective ambiguity. Consider the *correcting* pass. It is very unlikely that all of the images would have been classified the same way if the different subsets were corrected by different individuals. For instance the character in the upper left hand corner is subjectively ambiguous enough to result in classification by different readers as either a *four* or a *nine* when viewed in isolation. But, when viewing that character as part of a screen full of *fours* or screen of *nines*, most readers would probably accept the assigned classification due to context ambiguity.

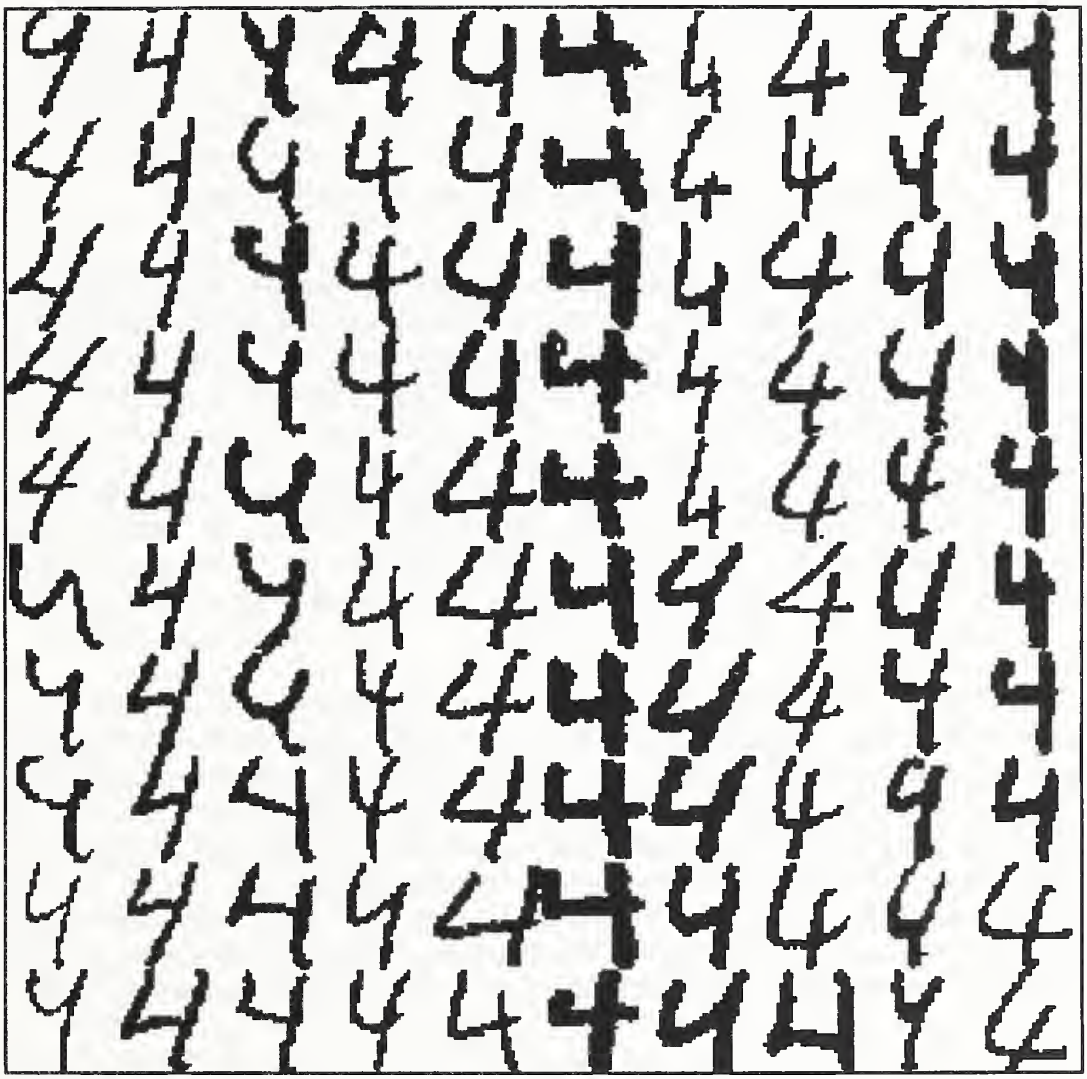

Figure 2: Example of several hand written characters

Figure 3: Example of the ultimate ambiguous digit set.

All types of reader and writer ambiguity are grouped together and refered to collectively as Writer/Reader (WR) ambiguity. Figure 3 shows the four ambiguous characters that can represent all ten digits. For example, image A can be a seven (7) or a one (1), image B can be a zero (0), two (2), six (6), or eight (8), image C can be a three (3) or five (5), while image D can be a four (4) or nine (9).

It is possible to test any given set of handprint images for writer ambiguity entirely by machine, but classifications under different conditions by a number of different people are required to test for reader ambiguity. No tests of TD1 were carried out for either type of ambiguity. Therefore, it is likely that TD1 suffers from a least two types of reader ambiguity, and it is not possible to rule out writer ambiguity.

It is not clear that it is desirable, much less practical, to remove all WR ambiguous images from an test set. First, we have no proof that there are any images that are WR unambiguous over a large enough set of readers and writers. However, it is likely that the majority of images of handprint characters are WR unambiguous over any reasonable set of readers and writers; handwriting has been under pressure to evolve in the direction of unambiguity since the invention of writing. There is a more important reason why any but the most preposterous WR ambiguous images should be retained in an image test set: they do occur at a certain frequency in real samples of handprint. Moreover, ideal OCR performance should recognize WR ambiguous images as WR ambiguous and treat them as such, as described below.

4.3 Ideal Confidence Values with Reader Ambiguity

For every image in a set of images of handprint digits, there exists a set of ten numbers $p_{WR}(i,j), j = 0,\ldots,9$ and $i = 1,\ldots,I$, where I is the number of images in the set, that give the probability over a given set of writers and readers that the i^{th} image represents the digit j. Thus

$$\sum_{j=0}^{9} p_{WR}(i,j) = 1. \tag{1}$$

For ideal OCR performance, the i^{th} image is classified as the value of j for which $p_{WR}(i,j)$ has the greatest value, or any one of the greatest in case of a tie. This defines a function $j(i)$. The WR probability that $j(i)$ is the correct classification of the i^{th} image is defined to be $p_{WR}(i,j(i))$. If the images in the set are now reordered from least to greatest $p_{WR}(i,j(i))$, they are in an optimum order for rejecting the least probable classifications while retaining the most probable for any given choice of rejection fraction.

Let $N_{WR}(i)$ be the number of digits j for which $p_{WR}(i,j)$ is non-negligible for a given i. For the digit test of TD1 the range of $N_{WR}(i)$ includes 1, 2, and 10, corresponding to WR unambiguous images, images that are WR ambiguous between two digits, and images that are completely unintelligible as digits. We do not know if this exhausts the range of $N_{WR}(i)$ for TD1 or other practical sets of segmented characters.

If $N_{WR}(i) = 1$, then $p_{WR}(i) = p_{WR}(i,j(i)) = 1$. However, if $N_{WR}(i) > 1$, we know nothing about the $p_{WR}(i,j)$ and their distribution with respect to j except that $p_{WR}(i,j(i)) \geq 1/N_{WR}(i)$. Experience tells us that $N_{WR}(r) = 1$ (no WR ambiguity) for most of the images in sets of practical significance, and that $N_{WR}(r) = 2$ for the majority of the remaining images, but it tells us little more. This prevents us from making any *a priori* statements about the functional form of $p_{WR}(i)$. Nevertheless, it is clear that an ideal OCR system should be able to recognize the WR ambiguous characters in a set of segmented characters and assign them the appropriate WR probabilities as confidences rather than any other confidence measure, although real systems may not be capable of this level of performance.

5 Conclusions

NIST needed a large set of segmented characters for use as a test set for the First Census OCR Systems Conference. A machine-assisted human classification system was developed to expedite the process. The NIST recognition system was augmented with human verification to produce the test database. The verification system gave each segmented character a status which could be one of four values. These four values were *accepted*, *rejected*, *to-be-checked*, and *to-be-corrected*. The output of both the recognition and the verification systems was either *accepted* and *rejected* values. To run the verification system all items marked *accepted* by the recognition system were relabelled *to-be-checked*, and all labelled *rejected* were relabelled *to-be-corrected*. As determined by the previous system the items marked *to-be-checked* were viewed in large groups of the same class by a human. This produces context bias. This means that the person viewing the character is biased towards the current classification. There is also context bias when the items marked *to-be-corrected* are viewed for the same reasons. From the *to-be-checked* staus an item can move to *accepted* or *to-be-corrected*, while from the *to-be-corrected* status it can move to *rejected* or *to-be-checked*.

Accurate databases are needed for testing and training of OCR systems. Machine recognition which is only 90% accurate, generates databases with too many errors to be useful for testing or training purposes. If machine recognition were 100% accurate there would be no reason to test the OCR technology, and human interaction for verification would not be needed. Recognition of the complete database by using a human observer is too time consuming.

Machine-assisted human classification uses the best of both methods and produces accurate databases in a short amount of time. This is done by allowing context biasing to aid the human verifier. Machine-assisted human classification assumes that a machine recognition system can produce high confidence recognition before human interaction verifies recognition classifications.

References

[1] R. A. Wilkinson, J. Geist, S. Janet, P. J. Grother, C. J. C. Burges, R. Creecy, B. Hammond, J. J. Hull, N. J. Larsen, T. P. Vogl, and C. L. Wilson. The First Optical Character Recognition Systems Confernce. Technical Report NISTIR 4912, National Institute of Standards and Technology, August 1992.

[2] C. L. Wilson and M. D. Garris. Handprinted character database. Technical Report Special Database 1, **HWDB**, National Institute of Standards and Technology, April 1990.

[3] M. D. Garris and R. A. Wilkinson. Handwritten segmented characters database. Technical Report Special Database 3, **HWSC**, National Institute of Standards and Technology, February 1992.

[4] R. A. Wilkinson. Handprinted Segmented Characters Database. Technical Report Test Database 1, **TST1**, National Institute of Standards and Technology, April 1992.

[5] R. A. Wilkinson. Segmenting of Text Images with Massively Parallel Machines. In D. P. Casasent, editor, *Intelligent Robots and Computer Vision*, volume 1607, pages 312–323. SPIE, Boston, 1991.

[6] M. D. Garris, C. L. Wilson, J. L. Blue, G. T. Candela, P. Grother, S. Janet, and R. A. Wilkinson. Massivelly parallel implementation of character recognition systems. In *Conference on Character Recognition and Digitizer Technologies*, volume 1661, pages 269–280, San Jose California, February 1992. SPIE.