

A 100-Font Classifier

Henry S. Baird
Reid Fossey

AT&T Bell Laboratories
600 Mountain Avenue, Room 2C-557
Murray Hill, New Jersey 07974 USA

Abstract

We present an engineering study of a classifier (image recognition algorithm) for at least 100 font styles of the printable ASCII character set. The fonts have been chosen for maximum coverage of 20th Century American publications. Training and testing stages have been largely automated using a systematic image database which was pseudo-randomly generated using a parameterized model of imaging defects. Engineering benchmark tests have been run on one million images. We have measured four aspects of classifier performance: accuracy, uniformity, generalization, and compression. The classifier achieves better than 99.7% top choice correct on printed English books, using dictionary context. The technology generalizes strongly, indicating that the classifier should perform equally well on many more than the 100 fonts tested.

Keywords: Classification, printed text, polyfont, omnifont, OCR, character recognition.

1. Introduction

We give a detailed performance analysis of an experimental classifier for isolated characters from the Latin alphabet printed in any of 100 fonts. The underlying technology has been described in [Bai88a], and a smaller-scale trial (on ten fonts) is reported in [LB87].

The ability to recognize characters printed in any of a large number of fonts, with no on-line training or prior specification of fonts, is essential to any general-purpose optical character recognition (OCR) machine. Such a *polyfont*[†] capability has been claimed for several commercially-available page readers in recent years (manufacturers include Xerox, Daimler-Benz, Calera, and Toshiba). To our knowledge no systematic performance statistics substantiating these claims has been published. In much of the research literature, character recognition methods have been tested on data sets so restricted that it is difficult to judge whether or not they will scale up successfully. For these reasons, a detailed account of a large-scale polyfont classifier may be a useful contribution. In addition, we believe that engineering issues arising in such large-scale trials are worthy topics for basic research, for several reasons:

1. Certain properties of classifiers, including *generalizing power* and *compression*, are

[†] Some authors use the term “omnifont,” which in this context should never be taken in its literal sense of “all fonts.” No existing OCR machine performs equally well, or even useably well, on all of the hundreds of fonts used by modern typesetters.

interesting on theoretical grounds, but are usually discussed only qualitatively. The rich variety of shapes in large-scale polyfont experiments offers an opportunity to assess these quantitatively.

2. Virtually all pattern recognition methods, when reduced to practice, require either simplifying assumptions (to fit theoretical models), or shortcuts in their implementation (to reduce cost), or both. As a result, their time or space requirements may grow faster than small-scale trials may suggest, especially under pressure of ambitious accuracy goals. We will report on the scaling characteristics of our technology as it has coped with larger and larger numbers of fonts.
3. Since all recognition methods must fail on images that are sufficiently distorted, no account of classifier performance is complete without a discussion of the margins of performance under a quantifiable model of image defects. As a step towards this goal, we have integrated an explicit defect model into the trials.
4. Future page readers must be versatile, easily adapting to different languages and writing systems. Evolution towards this goal is critically dependent on greater automation in the construction of classifiers. This experimental trial involves a highly-automated, language-independent inference procedure.

The trial was carried out in three phases:

1. Selection of 100 font styles to achieve broad coverage of printed and typewritten 20th Century American books, magazines, journals, newspapers, and business correspondence.
2. Design of an experimental trial to ensure that classifier performance statistics are unbiased and uniformly fair over a range of sizes and commonly-occurring imaging defects.
3. Analysis of accuracy and uniformity over a range of sizes, symbols, and imaging defects. In particular, we discuss the scaling characteristics of the technology, including generalization and compression.

Some background on the classification technology is given in Section 2. The selection of fonts is motivated in Section 3. The engineering design of the trial is given in Section 4. Accuracy and uniformity of the resulting classifier are discussed in Section 5. Generalizing power and compression are described in Section 6. Appendix I discusses the evidence supporting the choice of fonts. Appendix II illustrates the fonts included in the trial.

2. The Classifier Technology

The classifier technology used here is described in [Bai88a], with a few modifications described below. Briefly, it extracts local geometric shapes from the input image, maps this diverse collection of shapes into a feature vector with binary components, and then infers/classifies using a single-stage Bayesian classifier under an assumption of class-conditional independence among the features. Thus it represents a hybrid of structural shape analysis algorithms with statistical decision theory. The geometric shapes are derived from moments of area and boundary analysis, and include connected components, holes, edges, locally-maximal convex and concave arcs, and intrusions from the convex hull (this differs from the set used in [Bai88a]: in particular, we no longer vectorize). The feature identification mapping is not specified manually, but is determined automatically by the distribution of geometric shapes in the training set.

The Bayesian classifier is inferred using conventional supervised learning. Classifier runtime is $O(CF + \log C)$ and space is $O(CF)$, where C is the number of classes and F the number of binary features. There may be more than one class for each symbol in the alphabet (details in Section 4).

The use of a quantitative, parameterized model of imaging defects [Bai90a] permits us to build these classifiers with a minimum of manual effort. The model includes parameters for size, digitizing resolution, blur, binarization threshold, pixel sensitivity variations, jitter, skew, stretching, height above baseline, and kerning. It has been calibrated on image populations occurring in printed books and typewritten documents. Associated with it is a pseudo-random defect generator that reads one or more sample images of a symbol and writes an arbitrarily large number of

distorted versions with a specified distribution of defects.

3. The Selection of Fonts

We wished to choose a set of fonts to cover as much as possible of 20th C. American publications, including books, magazines, newspapers, and typewritten material. We were interested primarily in *body text* typefaces, used for reading matter, and so we have ignored display faces, used for preliminary pages, part and chapter titles, running heads, and sometimes sub-heads [CMS82]. We also neglect decorative and advertising designs such as script, swash, black-letter, and extreme Egyptian faces.

We use the term *font* in the sense usual among computer typesetters, to mean a design that is distinguishable by shape rather than by size. Thus we do not count each point size of text as a different font, as letterpress typographers often do [Bro83]. Also, we consider Times Roman and *Times Italic* to be different fonts even though they belong conventionally to the same typeface family. We will not distinguish moderate variations in weight (light/bold) and width (condensed/expanded), suitable for use in body text, since these are represented by deformations in the pseudo-randomly generated training set. Thus, our 100 fonts represent approximately 50 complete body-text typeface families.

Font usage evidence is discussed in Appendix I, and the 100 fonts are illustrated in Appendix II.

4. Design of the Experimental Trial

For each of the 100 fonts, for each of the 94[†] symbols in the printable ASCII set —

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789 .,:;"'``*^  
?!@#$%&/_\~-_+= <>(){}[]
```

— and for each of 10 point sizes in the range [5,14], we generated 25 samples using the image defect model, at a digitizing resolution of 300 pixels/inch (ppi). Half of this set (the odd point sizes {5,7,9,11,13}) was used to train the classifier, and the other half (the even sizes {6,8,10,12,14}) to test it. Each sample image was labeled with baseline location and nominal text size (in points), so that it could be tested in complete local geometric context. Nominal text sizes, assigned by type designers, vary unsystematically from font to font: these irregularities were compensated for using the method described in [Bai88b].

The training procedure (described in [Bai88a]) is almost entirely automatic: the only manual step required is *splitting* of symbol classes into *variant* classes. For example, we have found that, in order to achieve high accuracy on all 100 styles of the symbol /a/, we must split them into six variant classes:

- 1) **a a a a a a** etc.
- 2) **a a a a a a** etc.
- 3) **Q a a a Q a** etc.
- 4) **a a**
- 5) **a**
- 6) **a**

Variants (1) and (2) represent two grossly different styles of /a/. The other four are required to represent variations on (2) which are only subtly different from some styles of /o/. Note that in variants (1), (3), and (4), roman and italic styles are successfully combined in one variant class.

For some symbols, such as /=/, no splitting is required; for others, such as /J/, as many as 18 variants are needed. On average, among all the symbols, six variants are required (see Section 6). In addition to having a strong effect on accuracy, splitting determines the time and space demands

[†] Symbols `_~^"{}<>=+@` were unavailable in half of the fonts.

of the classifier (see Section 6 below).

The necessity of splitting appears to be due to our use of a single-stage Bayesian linear classifier, which may perform poorly when class distributions in feature space are not unimodal and well-separated. Several attempts to automate splitting (using clustering, *etc*) have so far failed to yield results competitive with expert judgement. It can be argued that this is a weakness in the approach. Other classification methods, such as multiple-stage neural networks trained by back-propagation [LBD90], are capable of learning classes that are not unimodal, at least in principle. However, learning subtle distinctions of the kind illustrated above might require unacceptably long training times. We are not aware of any successful application so far of such a method to large-scale polyfont classification problems.

It is tempting to view splitting as a straight-forward engineering trade-off between computing resources and accuracy. When splitting is maximized, putting each font-symbol in a separate variant class, then time and space demands will be greatest, but accuracy will presumably also be the best possible. This is equivalent to running a complete set of font-specific classifiers in parallel, what is sometimes called *multi-font* classification. It is important to note that the high accuracy of multi-font classification may only be realized on the fonts explicitly included in the training set. Ideally, one wants a classifier that performs equally well on fonts that are “similar” but not identical to these. Such a generalizing ability is enhanced when several slightly different font-symbols are combined in a single variant: the resulting statistical distribution may generalize to embrace other font-symbols similar to these. So in fact there are two motives for minimizing splitting: to reduce time and space demands, and to improve generalization.

5. Accuracy and Uniformity

Testing on pseudo-randomly generated samples has advantages and disadvantages. One advantage is that such a test is, at least in principle, replicable, and so permits comparisons with competing classification methods. Another perhaps more important advantage is that the performance results are uniformly-fair over the cross-product of fonts, symbols, and sizes, and under a consistent set of image defects: as a result, predictions of relative performance among fonts, symbols, and sizes can be made with high confidence.

In this study, our principal interest is in the classifier’s uniformity, that is in relative rather than absolute accuracies. Thus, when computing statistics, we will assume that all 100 fonts and all 94 symbols occur equiprobably, and that image defects are distributed as specified in the image defect model. Here are the test results, displayed by text size.

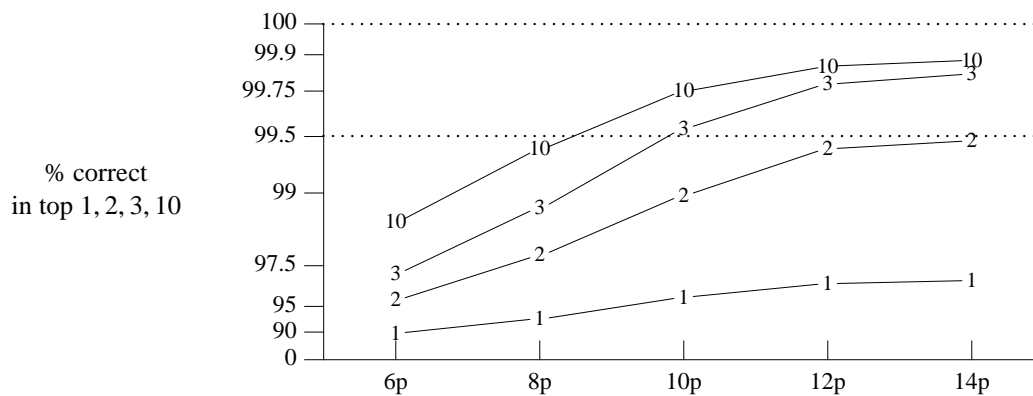


Figure 1. Accuracy as a function of text size (in points at 300 ppi), averaged over all 100 fonts and 94 symbols, assuming all fonts and symbols are equiprobable. Per cent correct in top 1, 2, 3, and 10 choices are plotted separately. The vertical scale is smoothly distorted to reveal details above 99.5%. (About 200,000 samples per datum.)

Top-1 statistics estimate the expected performance on symbols in isolation (that is, in the absence of any known contextual constraints). Top-3 statistics predict the performance of fast, shallow

data-driven contextual analysis using English dictionary look-up and punctuation rules. Top-10 statistics indicate upper bounds on performance, since it is unlikely that any fast method for data-driven contextual analysis could improve on them.

Note that accuracy improves monotonically on larger images: this is of course to be expected in all classifier technologies. In designing this test, we chose a range of sizes that would reveal the “critical size threshold” of the classifier technology: that is, the size of text below which most errors are due to coarse spatial quantization. This can be thought of as a measure of the noise immunity of the technology.

The data in Figure 1 suggest that the critical size is approximately 9 point at 300 ppi (or, equivalently, about 19 pixels per x-height): above this threshold, top-3 accuracy exceeds 99.5%, and furthermore soon flattens out. It is interesting to note that, above the critical size, only marginal improvements occur in top- k accuracy for $3 < k < 10$: in other words, symbols can be recognized either correctly in top-3, or not at all, with high confidence.

The errors remaining above critical size are distributed among the symbols as shown in the following figure.

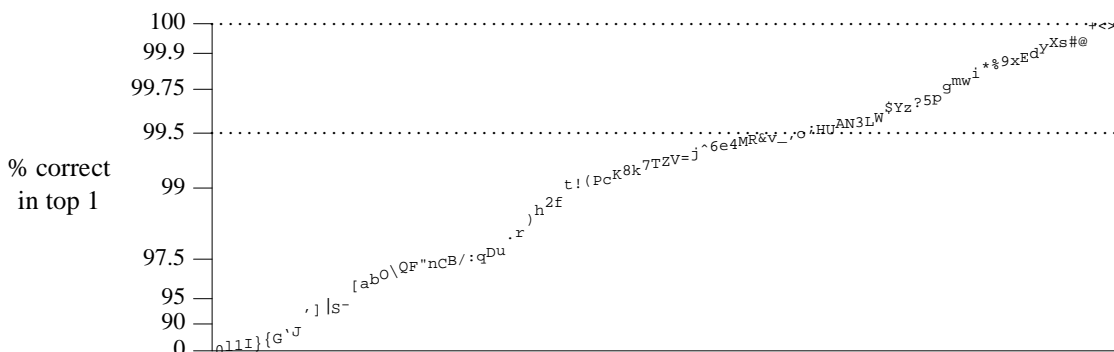


Figure 2. Top-choice accuracy of each of the 94 symbols, averaged over all 100 fonts and over the sizes 10p, 12p, and 14p (at 300 ppi). The 94 symbols are plotted in ascending order of accuracy. (About 7000 samples per datum; fewer for the symbols not present in all fonts.)

Seventy-five per cent of all top-1 errors occur on only 13 (14%) of the symbols:

0 1 I I { { G ' J '] | S

Not unexpectedly, /0/ (numeric zero) is often mistaken for /O/ (alphabetic "oh"), /G/ for /C/, /' for /'/, and /S/ for /5/; the rest are all similarly-shaped “vertical-stroke” symbols: 1 1 I I { { }. Of course, many of these confusions occur *across* fonts, and are arguably inevitable in any large-scale polyfont trial. Some inevitable confusions occur within a single font, where the typeface designer has neglected to vary the shapes: for example, in Times Roman, one and “ell” appear as 1 l, and in Futura Roman, “ell” and “eye” appear as 1 l.

Next we examine the uniformity of the classifier’s performance over the range of fonts. For this purpose, we use top-3 accuracies.

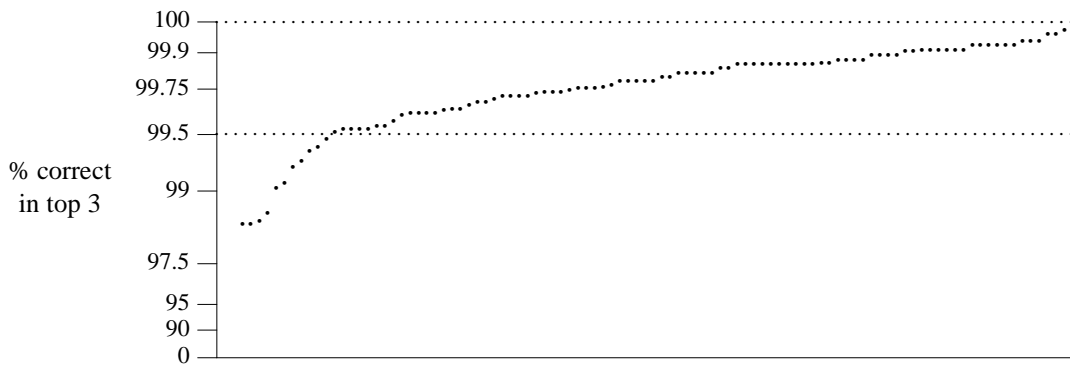


Figure 3. Top-3 accuracy of each of the 100 fonts, averaged over sizes 10p, 12p, and 14p (at 300 ppi). The fonts, represented by bullets, are plotted in ascending order of accuracy. (About 5000 samples per datum.)

Note the remarkably consistent performance across the fonts. The most difficult fonts, with accuracies below 99.5% (computed as in Figure 3), are shown below, with their worst (top-3) symbols:

98.57	Gillsans Roman	l I /
98.57	Gillsans Italic	l I a /
98.61	Galliard Italic	J O \ { }
98.74	<i>Avant Garde Italic</i>	/ T \ / { }
99.05	Galliard Roman	J }
99.12	<i>Avant Garde Roman</i>	l l { }
99.27	Univers Italic	I ' l /
99.32	New Century Schoolbk Roman	l { } /
99.40	Caslon Old Face Italic	O l /
99.43	Weiss Italic	G I O ; / u

Sans serif fonts do somewhat worse than serified fonts overall, but the classifier does not favor roman over italic styles. Generally, designs with hairline strokes (such as Caslon Old Face) do worse than average. Case confusions are rare, and difficult punctuation pairs such as /./,/ and /:/,/ are easily distinguished in most fonts. Also, the most complex and variable shapes, such as ampersands /&/ and at-signs /@/, present few special problems.

In summary, the test shows that almost all non-uniformities in classifier performance across fonts are due either to coarsely-digitized images (x-heights smaller than 18 pixels), or to simplistic designs (vertical-stroke characters).

This is good news, since it suggests that the technology will be able to cope with many more than 100 font styles, while maintaining this level of performance, with no changes to the algorithms. Also, with only a few localized algorithm improvements, higher overall accuracy may be possible to achieve, by making feature-extraction less sensitive to digitizing errors (perhaps through greater smoothing), and by adding a few new features such as relatively subtle boundary shapes occurring among vertical-stroke characters.

The main disadvantage of testing on artificially-generated data is of course that the results may not predict performance on actually-occurring images. We have attempted a rough calibration of the classifier's absolute accuracy. As one component of a complete page reader [Bai90b] that uses shape-directed resegmentation and exploits a dictionary, the classifier has been tested on original copies of several English books, printed in the Times and Garamond typeface families in 8, 9, and 10 point sizes. Imaged at a digitizing resolution of 400 pixels/inch, each book exhibited a final character recognition accuracy exceeding 99.7%. This accuracy is partly due to factors that we did not study in the pseudo-random trial, such as non-uniform occurrence of symbols.

6. Generalization and Compression

The rich variety of font styles offered an opportunity to measure certain properties of the recognition technology that are often discussed qualitatively but seldom quantitatively. One of these is *generalizing power*, the ability of a system to extrapolate automatically from a design set, and so to perform well on fonts not explicitly trained on. We observed that, in this domain, the recognition technology generalizes by about a factor of four (more precisely, 3.9): that is, training on 1/4 of the font-symbols available in the design set was sufficient to achieve high accuracy on all of them. This strong generalization is due of course to both the classifier technology and the nature of the application domain. Within this domain, however, this suggests that the classifier will perform comparably on many more than the 100 fonts that we have tested.

Another property of classifiers, interesting on theoretical grounds, is *compression*, the conciseness of the representation within the classifier. In the classifier, each class is represented by a single statistical record, containing first-order statistics of a few scalar features, and a vector of log-ratios of Bayesian *a priori* binary-feature probabilities. The number of these classes is a good measure of the resources consumed by recognition: both runtime and space requirements of the classifier are linear in this number (for a fixed number of features). There is always at least one class per symbol, but more than one class may be needed to represent all font style variations. Compression, then, can be computed as the ratio of font-symbols to classes: a compression of one meaning that each font-symbol requires its own class. Our classifier exhibits a compression of about 15.9: that is, averaged over the symbol set, about 16 different font styles are represented by one class. This is achieved in spite of variations due to imaging defects. Such high compression has practical implications: it suggests that classifiers for larger numbers of fonts can be constructed with only a fractional increase in computing resources.

7. Summary

We have been careful to design an experiment that is replicable by other researchers, at least in principle. This is why we have constrained ourselves, for the purposes of this report, to the ASCII set, to commercially-available fonts, and to a precisely-specified image defect model. This is also the motivation for a number of simplifying statistical assumptions, such as uniform distributions on symbols and fonts, top-*k* accuracy measures, and no forgiven confusions. We encourage other researchers to carry out comparable large-scale polyfont trials, using other classifier technologies.

We can now look back over five years of experiments with polyfont classifiers, using an essentially stable technology. Starting in 1986 [LB87] with cleanly printed images of ten fonts of 70 symbols, we have progressed to over 100 diverse fonts of full ASCII, whose images are distorted by a wide range of defects. Due to the increased difficulty, the critical size threshold has increased from 16 pixels/x-height to 19 pixels/x-height. However, three key results have remained nearly constant:

1. Top-1 accuracy, averaged over all fonts and symbols, rarely exceeds 97% at any size, unless either some confusions are forgiven or a non-uniform distribution of symbols is imposed.
2. Top-3 accuracy exceeds 99.5% on almost all fonts, above the critical size.
3. Top-3 accuracy, measured on isolated characters, predicts top-1 accuracy achievable on English books using dictionary and punctuation context.

The first result appears to be inevitable given modern typographical practice: therefore we feel that claims of ASCII polyfont accuracy in excess of 97%, to be credible, should always be qualified by specifying a minimum text size, a distribution on the alphabet, and a list of forgiven confusions. The second and third results suggest that our technology should be capable of extending to several hundreds of fonts with good results.

In fact, we have already experimented on non-ASCII symbols such as ligatures, digraphs, diacrits, and mathematical notation in more than 150 fonts, as well as on several non-Latin writing systems (for examples, see [Bai90b]). To date, the technology continues to scale up gracefully, in the way suggested by the experimental results we have reported.

8. Acknowledgements

Lorinda Cherry and Richard Drechsler helped provide machine-legible font descriptions. Tom Killian's fast and versatile bitmap printers have been invaluable. David Ittner, Doug McIlroy, and Tim Thompson offered helpful comments on an earlier draft.

9. References

- [Bai88a] Baird, H. S., "Feature Identification for Hybrid Structural/Statistical Pattern Classification," *Computer Vision, Graphics, & Image Processing* **42**, 1988, pp. 318-333.
- [Bai88b] Baird, H. S., "Global-to-Local Layout Analysis," *Proceedings, IAPR Workshop on Syntactic and Structural Pattern Recognition*, Pont-à-Mousson, France, 12-14 September, 1988.
- [Bai90a] Baird, H. S., "Document Image Defect Models," *Proceedings, IAPR 1990 Workshop on SSPR*, Murray Hill, NJ, USA, June 13-15, 1990.
- [Bai90b] Baird, H. S., "Anatomy of a Page Reader," *Proceedings, IAPR Workshop on Machine Vision Applications*, Tokyo, Japan, November 28-30, 1990.
- [Bro83] Brown, B., *Brown's Index to Photocomposition Typography*, Greenwood Publishing (Somerset, 1983).
- [Big90] Charles Bigelow, P.O. Box 1299, Menlo Park, CA 94026, July, 1990, personal communication.
- [Bun90] Ned Bunnell, Adobe Systems, 1585 Charleston Rd, Mountain View, CA 94039, July, 1990, personal communication.
- [Bur90] Elise Burroughs, American Society of Newspaper Editors, P.O. Box 17004, Washington, DC, July 1990, personal communication.
- [CMS82] *The Chicago Manual of Style*, The University of Chicago Press, Chicago, 1982.
- [Glu90] Nathan Gluck, American Institute of Graphics Arts, 1059 3rd Ave, New York City, NY, July 1990, personal communication.
- [KPB87] Kahan, S., T. Pavlidis, and H. S. Baird, "On the Recognition of Printed Characters of any Font or Size," *IEEE Trans. PAMI*, Vol. **PAMI-9**, No. 2, March, 1987.
- [Knu86] Knuth, D. E., *Computer Modern Typefaces*, Addison-Wesley, Reading, 1986.
- [LBD90] LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, and H. S. Baird, "Constrained Neural Network for Unconstrained Handwritten Digit Recognition," *Proceedings, Int'l Workshop on Frontiers in Handwriting Recognition*, Montreal, 2-3 April, 1990.
- [LB87] Lam, S. and H. S. Baird, "Performance Testing of Mixed-Font, Variable-Size Character Recognizers," *Proceedings, 5th Scandinavian Conference on Image Analysis*, Stockholm, SWEDEN, June 2-5, 1987.
- [Law89] Lawson, A., *The Anatomy of a Typeface*, Godine (Boston, 1989).
- [Law90] Alexander Lawson, 1601 East DelWebb Blvd, Sun City Center, FL 33573, July 1990, personal communication; formerly taught at Rochester Institute of Technology.
- [Leh90] Bruce Lehnert, Linotype Company, 425 Oser Ave, Hauppauge, NY 11788, July 1990, personal communication.
- [Lin89] *Mergenthaler Type Library Handbook*, Linotype AG, Eschborn, Germany, 1989.
- [Pro90] Archibald Provan, Rochester Institute of Technology, Rochester, NY, October 1990, personal communication.
- [Rom90] Frank J. Romano, Type World Newsletter, P.O. Box 170, Salem, NH 03079, personal communication.
- [Sch90] John Schappler, P.O. Box 170, Salem, NH 03079, July 1990, personal communication.
- [Sey90] Jonathan Seybold, Seybold Reports, Media, PA, July 1990, personal communication.

[Tyt90] Peter Tytell, Tytell Typewriter Company, 116 Fulton St, NYC, NY, October 1990, personal communication.

Appendix I. 20th C. American Font Usage

In the attempt to identify 100 fonts with the broadest coverage, we have consulted modern type designers [Big90] [Sch90], commercial type distributors [Bun90] [Leh90], academic historians of typography [Law90] [Pro90], associations of book and newspaper publishers [Glu90] [Bur90], industry watch publications [Sey90] [Rom90], and typewriter identification experts [Tyt90]. They were unanimous in the opinion that no careful study of the statistics of 20th C. American font usage has been published. Dr. Alexander Lawson, author of *Anatomy of a Typeface* [Law89], an authoritative history of typography, told us, “the rapid proliferation of fonts has overwhelmed attempts to track font usage.”

Nevertheless, by piecing together evidence from a variety of sources, it is possible to identify the most influential typeface designs. Lawson organizes his history around thirty typeface families, from which we selected eighteen body-text families: Baskerville, Bembo, Bodoni, Caledonia, Caslon, Century Schoolbook, Cheltenham, Futura, Garamond, Galliard, Goudy, Ionic, Janson, Optima, Palatino, Sabon, and Times. We complemented these with historically-important variations: the sans-serif faces Avant Garde, Eurostile, and Helvetica, the newspaper fonts Corona and Excelsior, and the slab-serif fonts Trade Gothic, Rockwell, and Serifa.

Perhaps the most reliable guide to typographical practice in 20th C. American trade press is the AIGA’s annual Best Books list [Glu90], published annually 1925-1980: our analysis of these lists suggested the addition of Bookman, Cloister, Gill Sans, Memphis, Plantin, Spartan, Trump Mediaeval, Univers, Weiss, and Zapf Book. Typewriter faces are represented by Courier (in two variations), Typewriter Pica, Typewriter Elite, Letter Gothic, Prestige Elite, and Print Out. A few more serified bookfonts were suggested by Chuck Bigelow [Big90], Bruce Lehnert [Leh90], and John Schappler [Sch90]: Aster, Breughel, Clearface, Frutiger, Leamington, Lucida, Melior, Meridien, Souvenir, Textype, and Walbaum.

A few strongly-recommended typefaces were unavailable to us in a convenient computer-legible format: Bell*, IBM Bookface, Bulmer*, Centaur*, Cochin*, Electra, Estienne*, Fournier*, Granjon, Lutetia*, Original Old Style*, Old Style*, Oxford*, Perpetua, Scotch, Snell, and Stymie. It would be an interesting exercise to add all 75 fonts of Donald Knuth’s Computer Modern typeface family [Knu86].

Truncating the list at 100 is of course somewhat arbitrary. It would be straightforward to extend it to about 200, using the sources that we have referenced, after which evidence of wide usage becomes relatively sparse.

As discussed in Section 3, we count the Roman and Italic variations within a single “font family” separately, as though they were distinct fonts. Moderate variations in weight (light/bold) and width (compressed/expanded) are represented through the effects of the image defect model.

Appendix II. The 100 Fonts Used in the Trials

These are Trademarks of Linotype AG [Lin89], unless shown otherwise in square brackets.

Aster Roman	Letter Gothic Roman [Adobe]
Aster Italic	Letter Gothic Slanted [Adobe]
Avant Garde Book Roman (ITC)	Lucida Roman [Adobe]
<i>Avant Garde Book Oblique (ITC)</i>	Lucida Italic [Adobe]
Bembo Roman	Melior Roman
Bembo Italic	Melior Italic
Bodoni Roman	Memphis Medium Roman
Bodoni Italic	Memphis Medium Italic
Bookman Light Roman [ITC]	Meridien Roman
Bookman Light Italic [ITC]	Meridien Italic
Breughel Roman	New Baskerville Roman [ITC]
Breughel Italic	New Baskerville Italic [ITC]
Caledonia Roman	New Century Schoolbook Roman
Caledonia Italic	<i>New Century Schoolbook Italic</i>
Caslon Old Face # 2 Roman	Optima Roman
Caslon Old Face # 2 Italic	Optima Italic
Cheltenham Roman	Palatino Roman
Cheltenham Italic	<i>Palatino Italic</i>
Clearface Regular Roman [ITC]	Plantin Light Roman
Clearface Regular Italic [ITC]	Plantin Light Italic
Cloister Roman	Prestige Elite Roman
Cloister Italic	Prestige Elite Italic
Corona Roman [Adobe]	Print Out Roman
Corona Italic [Adobe]	Rockwell Light Roman
Courier 10 Roman [Bitstream]	Rockwell Light Italic
Courier Twelve [Monotype]	Sabon Roman
Eurostile Roman	Sabon Italic
Eurostile Italic	Serifa Roman
Excelsior Roman [Adobe]	Serifa Italic
Excelsior Italic [Adobe]	Souvenir Medium Roman [ITC]
Frutiger #55 Roman	Souvenir Medium Italic [ITC]
Frutiger #56 Italic	Spartan Book Roman
Futura Book Roman	Spartan Book Italic
Futura Book Italic	Texttype Roman
Galliard Roman [ITC]	Texttype Italic
Galliard Italic [ITC]	Times Roman
Garamond # 3 Roman	<i>Times Italic</i>
Garamond # 3 Italic	Trade Gothic Roman
Gill Sans Roman	Trump Mediaeval Roman
Gill Sans Italic	Trump Mediaeval Italic
Goudy Old Style Roman	Typewriter Elite [Monotype]
Goudy Old Style Italic	Typewriter Pica [Bitstream]
Helvetica Roman	Univers #55 Roman
<i>Helvetica Italic</i>	Univers #56 Italic
Ionic Roman [Monotype]	Walbaum Roman
Ionic Italic [Monotype]	Walbaum Italic
Janson Text Roman [Adobe]	Weiss Roman
Janson Text Italic [Adobe]	Weiss Italic
Leamington Roman	Zapf Book Light Roman [ITC]
Leamington Italic	Zapf Book Light Italic [ITC]

Appendix III. Results by Font

List top-1, top-2, top-3 accuracy for each font, together with ten worst symbols.

Appendix IV. Generalization and Compression

Discuss generalization and compression as a function of the number of fonts.

Appendix V. Results on Entire Books

Summarize errors on complete printed books.
College, Twain, etc.