1-1-1993

The use of synthesized images to evaluate the performance of Ocr devices and algorithms

Frank Robert Jenkins
University of Nevada, Las Vegas

Order Number 1356028

The use of synthesized images to evaluate the performance of OCR devices and algorithms

Jenkins, Frank Robert, M.S.

University of Nevada, Las Vegas, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

The Use of Synthesized Images to Evaluate the Performance of OCR Devices and Algorithms

by

Frank R. Jenkins

A thesis submitted in partial fulfillment
of the requirements for the degree of

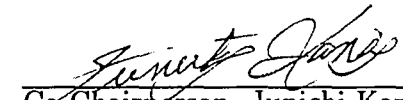
Master of Science

in

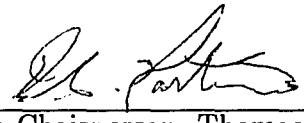
Computer Science

Department of Computer Science
University of Nevada, Las Vegas
August 1993

The Thesis of Frank R. Jenkins for the degree of Master of Science in Computer Science is approved.

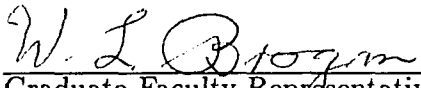


Co-Chairperson, Junichi Kanai, Ph.D

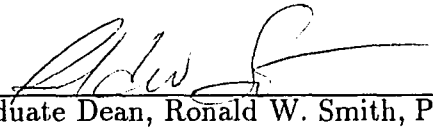


Co-Chairperson, Thomas Nartker, Ph.D

Examining Committee Member, Laxmi Gewali, Ph.D



Graduate Faculty Representative, William L. Brogan, Ph.D



Graduate Dean, Ronald W. Smith, Ph.D

University of Nevada, Las Vegas

August, 1993

ABSTRACT

This thesis will attempt to establish if synthesized images can be used to predict the performance of Optical Character Recognition (OCR) algorithms and devices. The value of this research lies in reducing the considerable costs associated with preparing test images for OCR research. The paper reports on a series of experiments in which synthesized images of text files in nine different fonts and sizes are input to eight commercial OCR devices. The method used to create the images is explained and a detailed analysis of the character and word confusion between the output and the true text files is presented. The synthesized images are then printed and scanned to mechanically introduce "noise". The resulting images are also input to the devices and analysis performed. A high correlation was found between the output from the printed and scanned images and the output from "real world" images.

Contents

List of Tables

List of Figures

Acknowledgements

The original idea that synthesized images might be useful in OCR research was suggested to me by Dr. Tom Nartker, the Director of the Information Science Research Institute and Dr. Junichi Kanai, my thesis advisor. I also owe them a debt of gratitude for their continuing assistance and encouragement throughout the project.

Professor George Nagy of Rensselaer Polytechnic Institute and Dr. Henry Baird of AT&T Bell Laboratories also provided encouragement and invaluable advice.

Steve Rice wrote many of the software programs that I used to analyze OCR device output, Kevin Grover helped with the arcane aspects of PostScript and Jim Porrazzo did much of the scanning.

The project was funded, in part, by a grant from the U.S. Department of Energy.

Chapter 1

Introduction

Background

This research began as an attempt to determine the effect of skew on the accuracy of Optical Character Recognition (OCR) devices and algorithms (hereafter referred to as "devices"). The desire to isolate skew as a controllable noise parameter led to experimentation with creating computer-generated document images. The skew experiments were interesting, but not as interesting as the images themselves. When I began experimenting with what I then called "ideal" images, it became clear that the flexibility they afforded could provide a valuable tool for OCR research. A number of experiments using those images were suggested. One of these suggestions was to use the images to determine the current state of the art in OCR.

It was assumed that if these computer-generated (now called "synthesized" or **S**) images were read by a number of devices, the upper level (highest possible character accuracy) of current OCR technology could be determined. The hypothesis was that a user of OCR devices could not expect any better accuracy from a device when it read digitized documents. The results of this experiment, on six devices, revealed some of the strengths and shortcomings of using such a procedure. The results were erratic. Some devices performed remarkably well on certain **S** images, but disappointingly on others. The results were interesting enough to warrant expanded testing and a deeper

analysis of the use of such images. Thus, this thesis is an initial exploration of the use of **S** images in OCR research. This author expects that **S** images will become a standard tool of OCR researchers in the next few years

Related Work

The author has surveyed over 1100 documents in the fields of Character Recognition and Document Analysis and could find no literature on the general subject of using computer-generated images to study OCR devices. The papers are listed in [Jenkins93a]. A few papers report on research into noise models. These are covered in a subsequent section of this thesis.

A report on this study was presented at a recent Symposium[1][Jenkins93b] Attendees volunteered the opinion that this study constitutes significant pioneering research.

[1]Second Annual Symposium on Document Analysis and Information Retrieval, April 26-28, 1993, Las Vegas Nev., Sponsored by the Information Science Research Institute and Howard R. Hughes College of Engineering, UNLV in cooperation with IEEE and IEEE Computer Society.

Chapter 2

Testing of OCR Devices

OCR Overview

There are a number of approaches to recognizing characters. The one described here is a typical one, and is used by one of the leading manufacturers of OCR products. The procedure is extracted from [Bokser92].

Scanning and Binarization

An OCR device reads a digitized image of a document.

A document image is a visual representation of a printed page. A digital document image is a two dimensional array representation of a document image obtained by optically scanning and digitizing a hard copy page. The process involves the sampling and conversion of light photons to electric signals, which are converted into pixel values. The image can be rendered in greyscale, in which each pixel is assigned a value representing a shade of grey; in color, in which the document is scanned as three grey level images with red, green and blue filters; or in binary form in which a thresholding operation assigns binary values to the pixels. Pixel values below the threshold are assigned black (usually the value 1), and those above are assigned white (usually 0) [Srihari86].

Segmentation

Before characters can be read from an image, they must be segmented from the rest of the document. This segmentation or zoning process is often done from the top-down, that is, first the areas of text must be separated from areas of pictures and other non-text. Then the text must be partitioned into columns (if appropriate), then lines, then words, and finally the individual characters.

There are a number of problems associated with zoning. Any skew that exists in the document, either locally or globally, must be corrected. Joining or touching characters must be separated, and noise must be identified and removed.

Character Recognition

If the final segmentation contains a properly segmented character image, then the output should be the character label that a human would assign to that image. The way a device tries to do this is through feature extraction and classification.

The features extracted from a character image are ideally those which preserve the properties that make an 'e', for example, different from other characters such as a 'c'. The images are divided into zones, and topological and geometric features such as horizontal and vertical lines, crossbars, curves, arcs, etc. are identified and extracted as feature vectors.

Classification algorithms are formed using training sets of data. The set should come from the variety of fonts and image quality that the device is expected to handle. Feature vector sets are assigned to characters and the training set is run through the classifier. The output is examined and feature sets adjusted accordingly. This iterative process is continued until the desired accuracy is achieved. When actual data is submitted, the classifiers will often assign some sort of *confidence level* to the output characters. If an image is unclassifiable, a *reject marker* will be assigned.

The words formed by the character streams can be checked for veracity by a number of means such as decision trees, n-grams or lexicons. Lexicons are of two types; built-in, which contain what the manufacturer considers to be common words, and user-defined, which are added to cover words peculiar to the user's application. Lexicons can use the *confidence levels* and *reject markers* in determining the most acceptable word formed by the characters.

Test Data

A number of types of input data can be used to determine the character accuracy of data entry systems: isolated characters, words, text-lines, text blocks or zones and complete pages. The majority of OCR performance results are reported for isolated characters. Such data may take either the form of test alphabets with an equal number of samples of each class (digits, upper case, lower case, punctuation, special symbols), or of characters extracted from a number of sample documents roughly corresponding to their frequency of usage.

During testing, the output characters are compared with the correct characters, and *character accuracy percentages* and *confusion data* are calculated. *Confusion data* are those characters that were misclassified, paired with the output which was produced; e.g. the 'c' was misclassified as an 'e' five times, the 'i' was misclassified as an 'l' twice, and so forth [Kanai93].

Word level is another level of testing. Here lexical techniques can be used and word level segmentation tested. Words that are joined, or a word that is segmented into two words can be studied.

By using text blocks it is possible to test text-line extraction methods. If a page containing text blocks is in columnar form, zoning capabilities can also be tested.

Acquiring Test Data

Analysis of the behavior of OCR devices is complicated by the large number of variables that affect the input. These variables can be classified as either *typesetting variables* or *noise variables.* Examples of *typesetting variables* are typeface, type size and type style.[1] Combinations of these variables must be used to study OCR devices that claim to have omni-font recognition capabilities. *Noise variables* are introduced by the printing, copying, and scanning processes. These variables cause image distortions.

Because of the large number of input variables, large scale experiments are a necessary part of OCR research. Many test images, along with tools to automate the experiments with these images, are needed. Since vendors do not disclose the workings of their devices, researchers must treat them as black boxes, making large scale experimentation even more necessary.

The acquisition of test data can be an expensive process. The major expense is involved with producing the *truth* data. This is the correct text file (usually in ASCII format) corresponding to the page image. It is compared with the device's output to determine accuracy. The text file must either be entered manually or produced by correcting an OCR device's output. In either case, careful editing must be done to insure that text as close to 100% accurate as possible is prepared. For one large U.S. Department of Energy project, Dickey estimated that it costs $3.79 per page to digitize and to prepare the corresponding ASCII text file, with an accuracy of 99.8%[Dickey91].[2]

Another way to create a test data set is to synthesize the images inside a computer

[1] For a discussion of typesetting variables, see [Rubenstein88].

[2] Some initial experiments demonstrated to the author the necessity for using actual words and not merely meaningless strings of characters as input to OCR devices. Many devices have lexicons that cannot be disabled, and these lexicons would try to force words from the character strings, corrupting the output.

starting with existing ASCII files. This approach has two major appeals. First, it is cost effective. Many **S** images can be created from a text file with virtually no manpower. Moreover, creating the associated *truth* representation is not necessary because the *truth* representation is the source file itself. Second, *typesetting variables* and *noise variables* are under the researcher's control.

Chapter 3

Noise Models

"Noise" is a concept that originated in communications theory. It is generalized to represent a number of nonideal circumstances [Schalkoff92]. Noise models can range from simple salt-and-pepper noise to sophisticated emulations of image distortions [Baird92]. Considerable work has been done in modelling image defects, but, to date, the validity and applicability of noise models for predicting performance in field conditions have not been extensively studied. Work is currently being done at the University of Washington to implement Baird's and a locally defined noise model [Kanungo92].

Baird discusses an image generator that simulates an imaging defect model. Application of the generator was limited to machine-printed text, but he points out that "An important factor ... is the use of training sets that are uniform in a strong sense: they should contain an equal number of samples of all symbols, over all fonts, and distorted by the same distribution of image defects."[Baird92] In a later article, he provides some specifics: "The input to the pseudo-random generator is an 'ideal' black and white image at high resolution: in practice, I use scalable outline descriptions purchased from typeface manufacturers."[Baird93] His testing was limited to a period and the digit symbols **0-9**. His use of the computer-generated images was limited to modelling, and did not include OCR device testing.

Neither of these efforts show that noisy **S** images can be used to predict the performance of OCR devices.

Chapter 4

Creating S Images

The end product of the synthesizing process should be an image in a form that can be read by OCR devices. The devices that were available to this researcher could all read the Tag Image File Format (TIFF), so this format was chosen.[1]

Another issue is resolution. When the bitmap of a page is synthesized by a computer program, the character shapes are limited by the program's image resolution and this becomes a limiting research parameter.[2] The standard in office applications today is 300dpi [Bayer92] and commercial OCR devices for the English language are designed to recognize page images digitized to at least 300 dpi, so the images in this study are synthesized at that resolution. Also, the environment at the Information Science Research Institute (ISRI), where this research was conducted, uses TIFF at 300dpi [Grover93]. Since ISRI also uses CCITT Group 4 compression, that is the compression used in this study.

The process also had to allow as much researcher control as possible over the images. Typeface, type font,type style, intercharacter and interword spacing, interlinear spacing (leading), and skew are all factors that the researcher may wish to control when creating images. The synthesizing process that was adapted for this

[1]For a description of TIFF, see [Aldus92].

[2]This is the primary reason the name of the images was changed from "ideal" to "synthesized", "ideal" carries the implication of infinite resolution.

study allows control over all of these except intercharacter and interword spacing.

One essential step was to automate the generation of **S** images from text files. A number of methods were tested including the use of desktop publishing systems. One of the most promising was *Microsoft's Word for Windows.* A number of tests were run, and this software could easily typeset complicated page layouts. Intercharacter and interword spacing and columnization could also be controlled [Micro91].

The problem with this software and other PC based systems was that the entire page image creation process could not be completely automated. Operator interaction was required to convert the output to a format that would be accepted by the OCR devices. Whenever such interaction is required, error-free operation cannot be guaranteed.

The process that was adopted involved writing batch routines and modifying existing public domain programs to control the creation of images. These programs are available on the UNIX operating system and it was possible to create batch processes in UNIX that would allow complete automation of typesetting of the ASCII files, bitmap creation, inputting the images to the devices, and gathering and processing device output. Figure 4.1 summarizes the process.

The first step is to typeset the ASCII text files. The LaTeX program was chosen for this purpose. LaTeX files are generated by concatenating sets of LaTeX commands and the text files. LaTeX allows various font types, styles and sizes to be introduced. For a description of how this is done, see [Lamport86].[3] The fonts chosen for this experiment are all in the public domain, as are all the programs used in the synthesizing process except for some batch files and text manipulation programs that were written by the author.

Next, to create a symbolic representation of the page, LaTeX output is translated

[3] LaTeX uses the magstep process to change font size. One magstep magnifies the base font size by 1.2. Thus a 14pt size generated from a 10pt size is actually 14.4pt.

into **PostScript**. The program **dvips**[4] is used to translate the LaTeX representation of the pages into the **PostScript** format.

S images are then generated from the **PostScript** files using the **PostScript** compatible program called **GhostScript.**[5] **GhostScript** generates the binary page images at a desired resolution from the **PostScript** files. The process of creating the binary image consists of rendering the **PostScript** language description of each character, which is found in a font dictionary, onto a raster output device by a process known as *scan conversion.* The device can be a printer, a computer screen, or, for our purposes, an image file in the portable bitmap (.pbm) format [Adobe90]. **GhostScript** implements the font type and font size that was introduced during the LaTeX process.[6]

The images in the portable bitmap format are converted into the **TIFF** format using the program **pnmtotiff**[7] and then compressed to Group 4.

It takes approximately two minutes for a Sun SPARCstation IPC to create each image tested in this paper using the above procedure.

---

[4]Radical Eye Software Ver. 5.497.

[5]Ver. 2.4.1, Developed by Aladdin Enterprise. Some modifications are required to use **GhostScript** in batch processing of files without interaction.

[6]The fonts chosen for this experiment are available in the **GhostScript** program.

[7]Derived by J. Poskanzer from ras2tiff.c, which is Copyright (c) 1990 by Sun Microsystems, Inc. The author is Patrick J. Naughton.

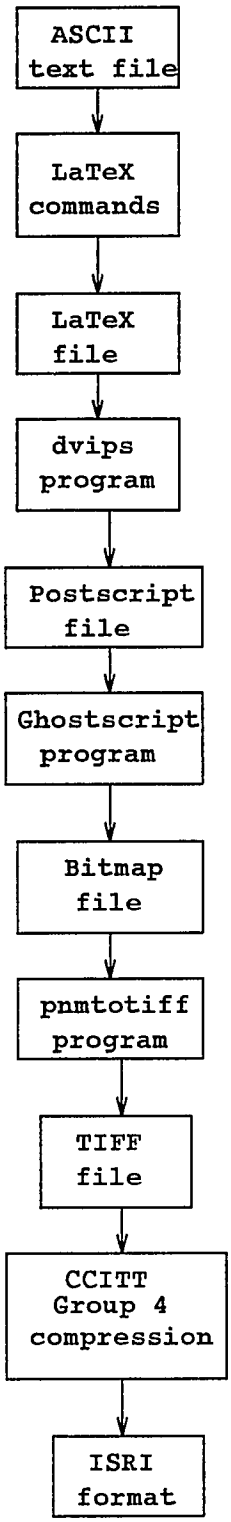


Figure 4.1: Flow diagram for creating synthesized images

Chapter 5

Experiments Using S Images

Hypothesis

Data obtained from **S** images are predictive of the data from "real world" or **RW** images. If this is proven, **S** images can be used to predict OCR performance.

Test Data and Procedures

To test the hypothesis, a set of **RW** images, preferably from a database that had already been tested is required. Such a database exists at ISRI[Bradford92]. The images used were 132 randomly selected text pages from 9300 pages of this database. The ASCII text *truth* representation of these images was used to create the **S** images.

The Courier, Helvetica, and Times typefaces were used in the experiment, and each typeface was synthesized in 10, 12 and 14 point sizes. Skew was kept at zero for all images and the type style was consistently roman. A total of 305 images were required to synthesize the 132 pages for each typeface and type size.[1] There were actually nine images created for each page, one for each of three font types in each of three font sizes, for a total of 2745 images.

[1]The reason for the large number of images is due to the fact that the original pages had to be divided into 242 single column zones. Many of these contained small (6 or 8 point) type sizes. When they were typeset into larger sizes and converted to images, some of the images were wider than 8.5 inches. Because some OCR devices cannot read images that wide, artificial line breaks were introduced. These line breaks caused some of the images to exceed 11 inches, and some devices would not accept this size. Artificial page breaks were introduced where necessary, resulting in a total of 305 pages to be converted to images.

There are 278,715 characters on these text pages.[2] Since each character was synthesized 9 times, a total of 2,508,435 characters were synthesized and tested.

The type of page property chosen for such experiments is important. Since this research is in its infancy, the author decided to examine only images of "main body", single column text and not tables, equations, graphs, headers, footers, captions, footnotes, etc.

The behavior of eight commercial OCR devices[3], two UNIX-based and six PC-based, was examined using the **S** images described above.

Since these are commercial devices, their inner workings are unknown. They must be treated as "black boxes", and researchers are limited to drawing inferences from the output.

A batch process was used to input each image in turn to all eight OCR devices simultaneously. Each device output a file containing its interpretation in ASCII format of each image. For the 2,745 images and eight devices there were 21,960 files created in this manner. Since each image was created from an ASCII text file, a comparison of that file with the file generated by a device indicated the device's accuracy when reading that image. Both character and word[4] accuracy were calculated.[5] Character accuracy was measured by counting the number of insertions, substitutions and deletions required to correct the device output to agree with the ASCII text. For c characters, i insertions, s substitutions and d deletions. Character accuracy is $\frac{c-i-s-d}{100c}$.

Word accuracy was calculated as: $\frac{cw}{tw}$, where cw is the number of words recognized correctly and tw the total number of words. The accuracies were also aggregated by

[2]There were 278,786 characters in the original pages. This included some tilde characters which were removed from the ASCII files prior to their conversion to **S** images.

[3]The devices, which will not be identified with the accuracy data in this report are: Caere OCR, Calera MM600, Cognitive Cuneiform, CTA TextPert DTK, ExperVision RTK, OCRON Recore, Recognita Plus DTK, and XIS ScanWorX API.

[4]A word is a sequence of one or more letters. Numbers and punctuation are not included in the word confusion reports.

[5]For a discussion of character accuracy, see [Rice93a].

font type (all three Courier sizes, all three Helvetica sizes and all three Times sizes) for each device and aggregated by all nine font style-size combinations.

The "system" lexicons were enabled on all devices. However, no "user defined" lexicon was used. No interactive "learning" modes were used, and no device received special training. Misrecognition of non-ASCII symbols was not counted against the devices.

The **RW** images of the files that were synthesized had previously been read by each of the devices, so a comparison between those accuracies and the accuracies from the **S** images could be made.

Results of the S Image Experiments

Tables 5.1 shows the character accuracy obtained when the devices processed the **S** images. Note that the highest accuracy was 99.97%. It was achieved by device #6 in the 10 point Courier font.

Table 5.1: Character accuracies for **S** images

Device	**Courier**			**Helvetica**			**Times-Roman**		
	10pt	*12pt*	*14pt*	*10pt*	*12pt*	*14pt*	*10pt*	*12pt*	*14pt*
1	99.93	99.92	99.95	99.90	99.90	99.90	99.90	99.91	99.92
2	99.93	99.92	99.90	99.62	99.32	99.54	99.90	99.89	99.82
3	99.84	99.78	99.71	99.60	98.91	98.91	99.89	99.88	99.68
4	99.82	99.88	99.91	99.71	99.61	99.70	99.78	99.92	99.88
5	95.56	99.30	99.17	99.82	99.83	99.50	99.92	99.86	99.59
6	99.97	99.78	99.95	99.81	99.74	99.35	99.94	99.95	99.95
7	99.80	99.85	99.83	98.25	98.77	99.83	99.60	99.89	99.66
8	99.71	99.63	99.74	99.78	99.66	99.61	99.50	99.68	99.83

Table 5.2 shows the character accuracy obtained when the devices processed the **RW** images. In almost every instance, the devices recognized the characters in the **S** images better than the characters in the **RW** images; however, no device was able to recognize all characters in any combination of typeface and size.

Table 5.2: Character accuracies for **RW** images

Device	**% Accuracy**
1	98.93
2	98.11
3	96.27
4	98.99
5	97.15
6	97.66
7	98.92
8	96.59

Table 5.3 shows the word accuracy obtained when the devices processed the **S** images. Note the low word accuracy (77.19%) for device #5 in 10 point Courier.

Table 5.3: Word accuracies for **S** images

Device	**Courier**			**Helvetica**			**Times-Roman**		
	10pt	*12pt*	*14pt*	*10pt*	*12pt*	*14pt*	*10pt*	*12pt*	*14pt*
1	99.77	99.87	99.93	99.70	99.74	99.79	99.74	99.75	99.81
2	99.68	99.67	99.62	99.60	98.04	98.71	99.66	99.71	99.66
3	99.80	99.66	99.52	98.73	96.78	97.59	99.60	99.85	99.54
4	99.23	99.80	99.80	99.33	99.54	99.64	99.57	99.83	99.83
5	77.19	99.68	98.60	99.14	99.38	99.15	99.78	99.49	99.89
6	99.88	98.86	99.86	99.47	99.26	98.42	99.85	99.87	99.84
7	99.66	99.54	99.38	98.57	99.48	99.54	98.22	99.54	98.84
8	99.80	99.74	99.66	99.13	98.52	98.43	99.50	99.50	99.47

Table 5.4 shows the word accuracy obtained when the devices processed **RW** images . As in the character accuracy tables, the devices did not always score higher on the **S** than **RW** images.

Table 5.4: Word accuracies for **RW** images

Device	% Accuracy
1	97.42
2	94.85
3	88.25
4	98.12
5	90.34
6	93.94
7	97.28
8	90.54

Analysis of the Results of Synthesized Image Experiments

Comparison with results from RW images

The fact that some devices achieved lower accuracies on certain font type and size combinations for the **S** images than for the **RW** images tends to question the validity of the **S** images. An analysis of the specific errors made by those devices, however, showed that in each case, the lower accuracy was attributable to chronic recognition errors, and that these errors were limited to that particular device and were not repeated by any of the other seven devices.

There were three such cases. First, in 10 point Courier, device #5 misclassified every 'c' (as an 'o'), every 'g' (as a 'q') and every 'N' (as an 'IV'). These were the reasons its character accuracy when reading **S** images was 95.56%, well below its 97.15% accuracy when reading the **RW** images. None of the other devices made similar errors. The word accuracy, which is generally dependent on the character accuracy, was also lower (77.19% vs. 98.92%).

The other two cases are device #7 reading 10 and 12 point Helvetica. It produced no output when the input character was ',' in 10 and 12 point, and it read every '9' as '0', and 'ti' as 'fl' in 10 point. No other device made these errors. This caused

its character accuracy (98.25% in 10 point and 98.77% in 12 point) when reading **S** images to be less than its **RW** accuracy (98.92%).[6] (See Appendix A for a list of chronic errors.)

Blank Errors

The most frequent error was the introduction of a superfluous blank character. That is, the breaking of a word into two or more words. These accounted for 24.7% of the errors over all the devices. This can only be interpreted as a segmentation problem. The fifth most frequent error was the joining of two words by not outputting the intervening blank character. This caused 5.94% of all errors. In one case (device #2 reading 12 point Helvetica), 1373 blanks were added and 275 missed. Had the blank errors not been made, the 99.32% accuracy would have been increased to 99.91%.

A summary of blank character errors by device is shown in table 5.5.

Table 5.5: Blank Character Errors - **S** Images

Device	added blanks errors	% of total errors	deleted blanks errors	% of total errors
1	326	15.1	576	26.8
2	2815	46.8	1432	23.8
3	5000	47.2	692	6.5
4	2436	49.6	119	2.4
5	2408	11.6	353	1.7
6	1990	46.0	190	4.4
7	503	4.0	322	2.5
8	1667	20.9	442	5.5

Predictive Ability of S Images

Overall, the accuracy demonstrated by the devices was disappointing. Considering that the images contained neither speckle, nor skew, nor touching characters,

[6]But the word accuracy was not lower. The reason is that punctuation marks and numbers are not considered in calculations of word accuracy

and since the inter-word and inter-line spacing was determined by software, a mean accuracy of 99.65% for the eight devices appears low (See Appendix B for aggregate accuracies). Only device #1 achieved accuracies of 99.90% or higher for all font types.

The ability of each of the nine font type and size combinations to predict the **RW** accuracies was also disappointing. It was examined by calculating the correlation between the **S** and **RW** accuracies for each device.[7] Character and word accuracy correlations for each font and size combination was found to be very low.

The correlations didn't improve much when the data was aggregated by size (the average of all three fonts for a given size), by font (the average of all three sizes for a given font), or aggregated by all nine font - size combinations, as table 5.6 shows.

Table 5.6: Correlation Coefficients between **S** images and **RW** images

Aggregate S image accuracy	Character Corr.	Word Corr.
All Fonts, 10pt	0.14	0.36
All Fonts, 12pt	0.44	0.88
All Fonts, 14pt	0.77	0.86
Courier Font, All 3 sizes	0.33	0.38
Helvetica Font, All 3 sizes	0.08	0.78
Times Font, All 3 sizes	0.33	-0.20
All fonts and sizes	0.38	0.59

Comparisons by Page Quality Groupings

An examination was done of device accuracy when page quality was a factor. Each of the 132 pages was sorted according to the median accuracy that was attained by six OCR devices when reading the pages [Rice92] and three groups were formed

[7]Correlation coefficients and best fit lines were calculated using formulas in [Walpole89].

containing approximately the same number of characters:

Best	-	39 pages containing 93,016 characters	(highest accuracy)
Middle	-	40 pages containing 93,586 characters	
Worst	-	53 pages containing 92,184 characters	(lowest accuracy)

Table 5.7 shows the **S** and **RW** image accuracy correlations using the same groupings. As the numbers show, the correlations are still low.

Table 5.7: Correlation Coefficients between **S** images and **RW** images considering Page Quality

Best **S** with Best **RW**	0.38	with All **RW**	0.40
Middle **S** with Middle **RW**	0.40	with All **RW**	0.36
Worst **S** with Worst **RW**	0.38	with All **RW**	0.39

Conclusions from S image experiments

The device accuracy obtained from **S** images is not predictive of the device accuracy obtained from **RW** images.

It was pointed out in the "Test Data" section that there were two general classifications of variables that affect images that are read by OCR devices: *typesetting variables* and *noise variables*. Since *noise* was eliminated in the creation of **S** images, the misclassifications of the characters in the **S** images must have been due to the *typesetting* variables. The high number of misclassifications in all typeface and size combinations indicates that none of the devices possess true omnifont capability.

The data also suggest that the devices were not trained on synthesized images. They may be designed to compensate for noise introduced by printers and scanners and are somehow at a disadvantage when trying to recognize synthesized characters.

Chapter 6

Experiments Using Printed and Scanned Images

Hypothesis

If noise of the type introduced by printers and scanners is added to the **S** images, the resulting images will simulate **RW** images sufficiently to be predictive of **RW** image output when read by OCR devices.

Test Data

To test this hypothesis, lacking a suitable noise model, the **S** images from each typeface and type size combination were rendered into **Postscript**, printed by a DEC LPS 20 laser printer (300 dpi), and each of the output pages was digitized using a Fujitsu M3096G scanner. Thus, the printer and scanner mechanically introduced real world noise to the **S** images. These images (called **PS** images) were then submitted to all eight OCR devices.

Results of Experiments Using PS Images

Table 6.1 shows the resulting character accuracy and table 6.2 the resulting word accuracy.

Table 6.1: Character accuracies for **PS** images

Device	**Courier**			**Helvetica**			**Times-Roman**		
	10pt	*12pt*	*14pt*	*10pt*	*12pt*	*14pt*	*10pt*	*12pt*	*14pt*
1	99.93	99.92	99.95	99.85	99.89	99.90	99.78	99.80	99.80
2	99.90	99.87	99.89	99.56	**99.55**	**99.62**	99.67	99.80	99.82
3	**99.89**	**99.88**	99.29	99.21	99.84	99.34	99.73	99.84	**99.78**
4	**99.88**	99.87	99.89	99.67	**99.70**	99.69	99.75	99.87	99.87
5	**99.86**	**99.80**	**99.26**	99.65	99.73	99.45	99.76	99.60	**99.77**
6	99.29	99.64	99.20	99.75	**99.85**	**99.71**	99.89	99.88	99.81
7	99.80	99.85	**99.84**	**99.60**	99.63	99.78	**99.72**	99.83	**99.86**
8	99.62	99.61	99.69	99.39	99.33	99.40	99.47	99.58	99.55

Note: There was one image that device #6 would not accept in 10pt Courier and one in 14pt Courier. This cost it 0.61% in Courier 10pt accuracy and 0.59% in 12pt.

Table 6.2: Word accuracies for **PS** images

Device	**Courier**			**Helvetica**			**Times-Roman**		
	10pt	*12pt*	*14pt*	*10pt*	*12pt*	*14pt*	*10pt*	*12pt*	*14pt*
1	**99.85**	**99.91**	**99.94**	99.55	99.74	99.78	99.28	99.43	99.13
2	99.66	99.63	**99.65**	98.42	**98.71**	**99.00**	99.17	99.44	99.58
3	**99.85**	**99.77**	98.69	97.21	95.87	**98.02**	99.00	99.53	99.51
4	**99.77**	99.79	**99.81**	99.08	99.39	99.54	99.52	99.78	99.81
5	**99.43**	99.60	98.60	98.15	98.97	98.90	99.20	98.08	99.43
6	98.94	98.17	98.52	99.44	**99.50**	99.28	99.80	99.79	99.28
7	99.40	**99.56**	**99.43**	**98.76**	99.20	99.36	**99.08**	99.41	99.43
8	99.47	99.36	99.46	98.35	98.45	**98.89**	98.92	99.10	98.28

Analysis of the Results of PS Image Experiments

Comparison with results from S images

A comparison of Table 5.1 with Table 6.1 and Table 5.3 with Table 6.2 shows improved character and word accuracy for some typeface and type size combinations in the **PS** versions (numbers in boldface in Tables 6.1 and 6.2). Even where the overall accuracy percentages were not improved by printing, some characters in the **S** images that were systematically misclassified were read correctly from the **PS** images. Examples of these characters are shown in Appendix C.

Examination of the character images suggests a possible explanation for the misclassifications. The **S** image characters have thin strokes that are one or two pixels wide. It seems that these devices treated the conjunction of thin strokes as indications of two touching characters rather than as part of one character.

The **PS** image characters are noticeably thicker, and such errors were not as prevalent. Indeed, the errors in the **PS** images were spread over almost twice as many different character confusions than in the **S** images, but not as many as in the **RW** images.

Image	Different character confusions	Number accounting for first 50% of total errors
S	1287	5
PS	2273	16
RW	7525	308

Graphs of the characters comprising the first 50% of the errors for each set of images follow:

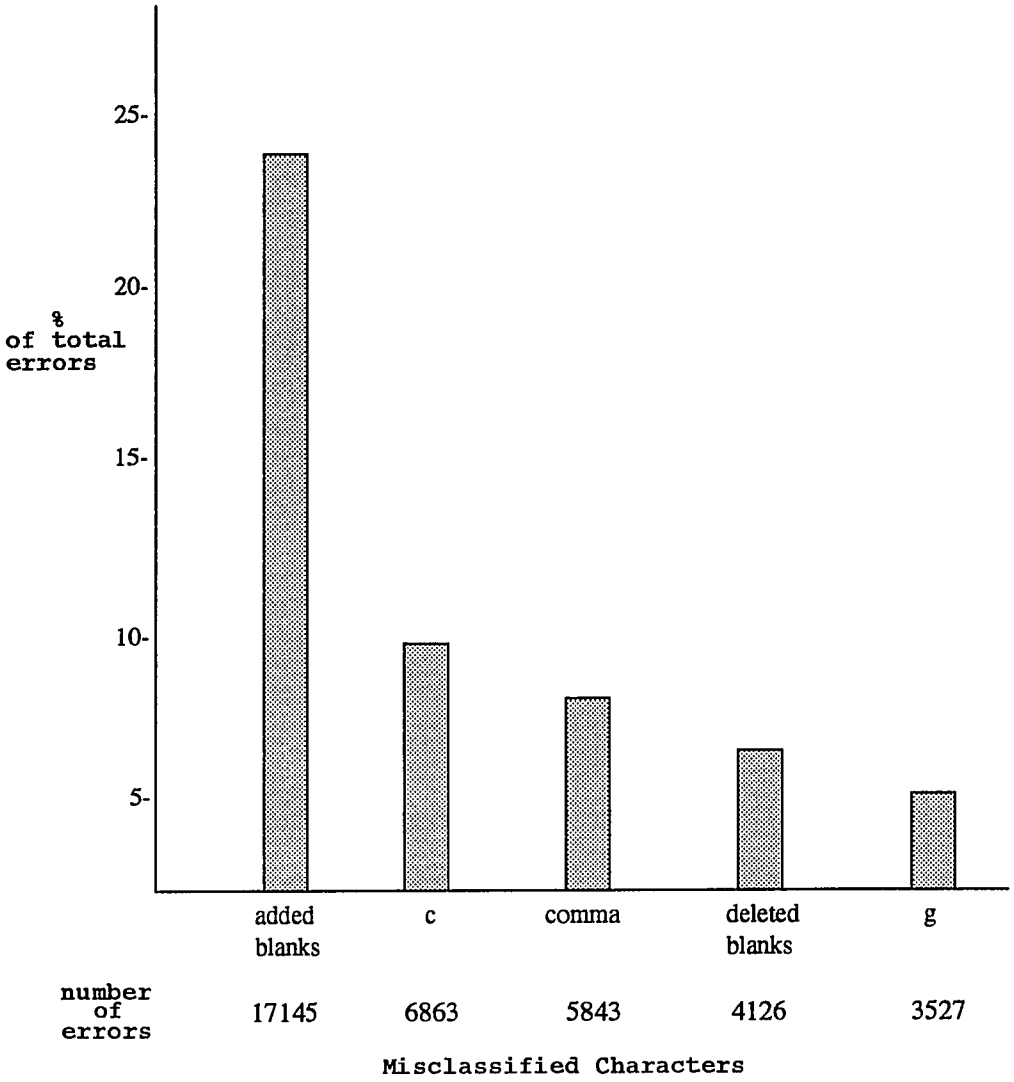


Figure 6.1: First 50% of errors - S images

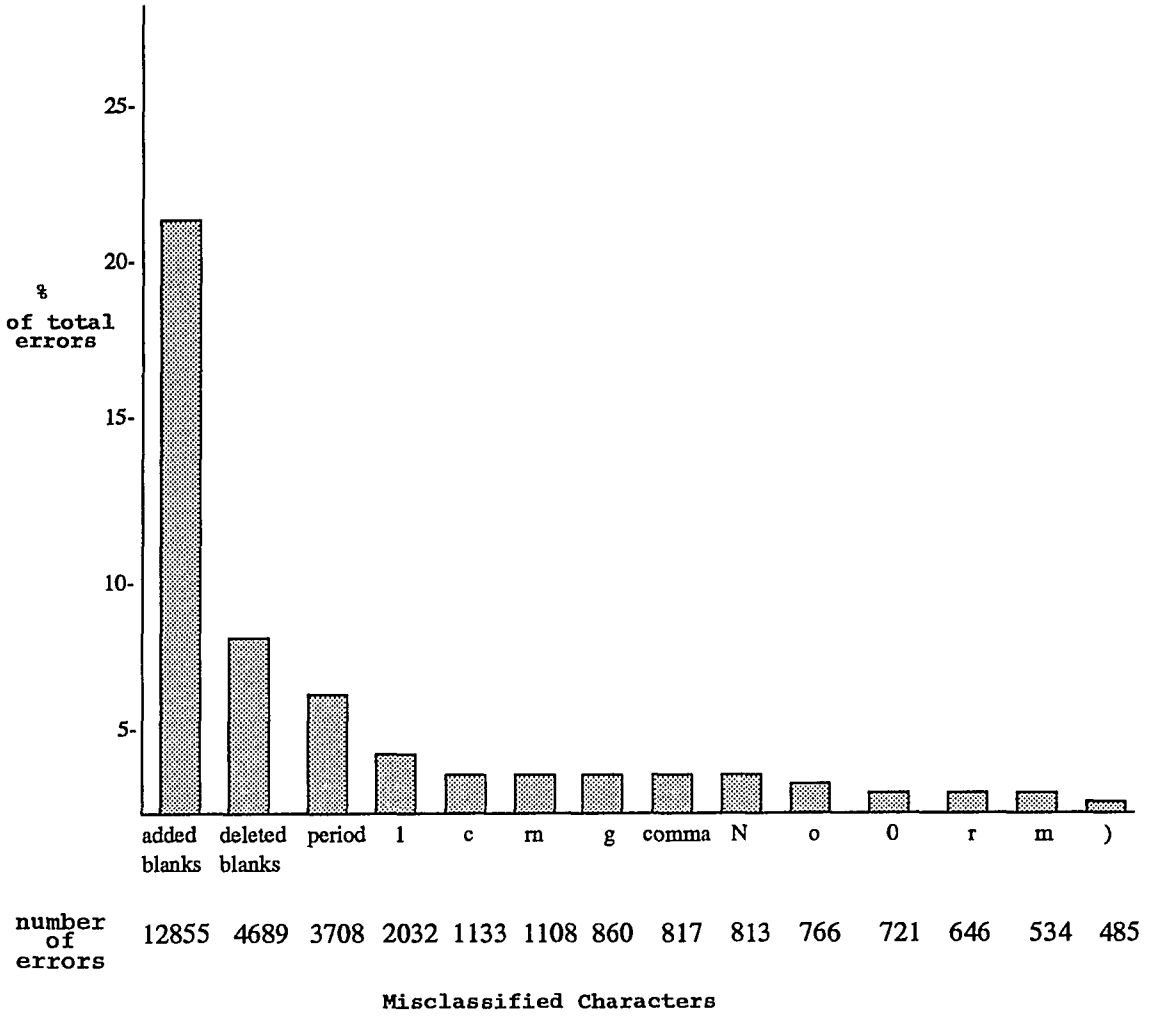


Figure 6.2: First 50% of errors - **PS** images

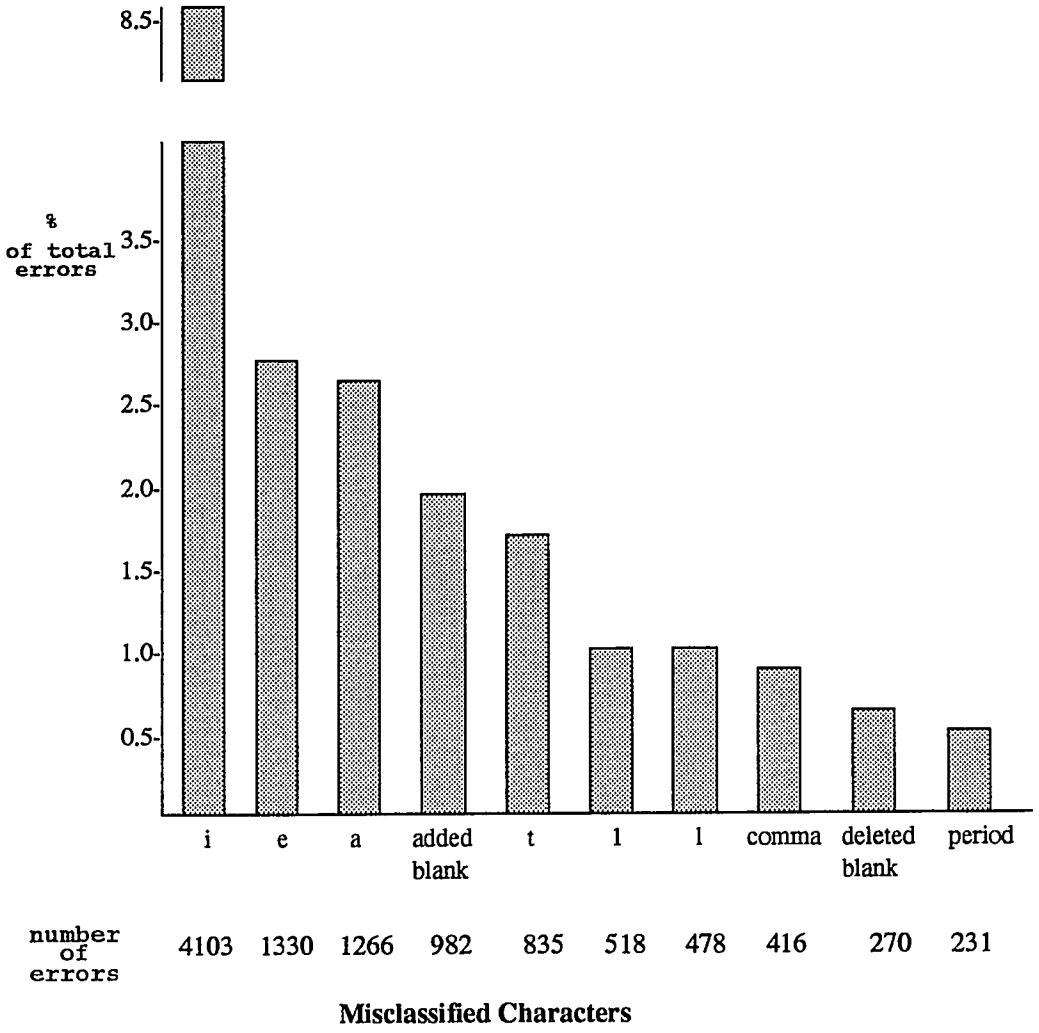


Figure 6.3: First 20% of errors - **RW** images

Note: Only 10 characters (comprising the first 20%) are shown. It would take a graph of 308 characters to show the first 50%.

Blank Errors

As in the **S** images, the blank errors for the **PS** images accounted for the most frequently observed confusion (see table 6.3). The added blanks accounted for 20.9% of the total errors, and the deleted blanks for 7.6%.

Table 6.3: Blank Character Errors, All Fonts and Sizes - **PS** Images

Device	added blanks errors	% of total errors	deleted blanks errors	% of total errors
1	411	12.4	572	17.3
2	1402	21.7	1554	24.1
3	4599	39.2	604	5.2
4	1671	33.4	129	2.6
5	2121	24.4	507	5.8
6	783	9.4	231	2.8
7	542	9.3	557	9.5
8	1326	10.9	535	4.4

By comparison with Table 5.5, the devices generally made fewer blank errors when reading the **PS** images than when reading the **S** images. As table 6.4 shows, they made even fewer when reading the **RW** images.[1]

[1]The added blank and deleted blank numbers in table 6.4 are multiplied by nine to allow direct comparison with the **S** and **PS** tables, both of which represent nine typeface and type size combinations.

Table 6.4: Blank Character Errors - **RW** Images

Device	added blank errors × 9	% of total errors	deleted blank errors × 9	% of total errors
1	396	1.5	360	1.3
2	666	1.3	324	0.1
3	369	0.4	405	0.4
4	504	2.0	180	0.7
5	1755	2.5	414	0.6
6	1845	3.1	90	0.2
7	603	2.2	279	1.0
8	819	1.0	378	0.4

Predictive Ability of PS Images

Since the devices are "black boxes", we can only speculate why they make different types of misclassifications when reading **S**, **PS** and **RW** images. We offer the following observations from the data presented thus far:

- The characters in **S** images are thinner than the devices are trained to read, and they misclassify certain characters, such as identifying a 'c' as an 'o' or an 'N' as '1\T'. Many of these errors are caused by poor character segmentation. In fact, devices seem to be anticipating broken and touching character errors and compensating for them.

- These segmentation errors cause the devices to either separate words by adding blanks, or, less frequently, to combine words by deleting a blank.

- When the **S** images are mechanically thickened and converted to **PS** images, many of these segmentation errors are eliminated. However, other errors, caused by the added noise, are introduced. Since the noise is not consistent, these errors are not as chronic, and the misclassifications are spread over more characters.

- When the devices read **RW** images, which they are trained to read, there are far fewer blank errors caused by segmentation, but more errors caused by the added noise. So errors such as misclassifying an 'i' as an 'l' or an 'I' become common because of the noise effect on the dot over the 'i'.

The net effect seems to be that the closer synthesized images come to emulating **RW** images by the addition of noise, the closer they predict OCR device output. This is supported by table 6.5.

Table 6.5: Correlation Coefficients between **PS** images and **RW** images

Aggregate PS image accuracy	**Character Corr.**	**Word Corr.**
All Fonts, 10pt	0.73	0.83
All Fonts, 12pt	0.89	0.58
All Fonts, 14pt	0.94	0.55
Courier Font, All 3 sizes	0.60	0.38
Helvetica Font, All 3 sizes	0.81	0.82
Times Font, All 3 sizes	0.55	0.56
All fonts and sizes	0.96	0.96

The all fonts and sizes correlations for both character and word accuracy was 0.96. We consider this correlation remarkable. Plots of the data along with best fit lines are shown in Appendix D.

Comparisons by Page Quality Groupings

Table 6.6 shows the **PS** and **RW** image accuracy correlations using page quality groupings. It also shows far higher correlations than the **S** and **RW** comparisons:

One phenomenon that was observed is the reduction in accuracy for both **S** and **PS** images as page quality deteriorates, shown in table 6.7.

Table 6.6: Correlation Coefficients between **PS** images and **RW** images by Page Quality

Best **PS** with Best **RW**	0.85	with All **RW**	0.90
Middle **PS** with Middle **RW**	0.89	with All **RW**	0.87
Worst **PS** with Worst **RW**	0.68	with All **RW**	0.65
All **PS** with Best **RW**	0.92		
All **PS** with Middle **RW**	0.94		
All **PS** with Worst **RW**	0.94		

Table 6.7: Character Accuracies by Page Quality - **S** Images

Device	**Best**	**Middle**	**Worst**
1	99.96	99.92	99.87
2	99.86	99.76	99.66
3	99.68	99.60	99.45
4	99.88	99.79	99.75
5	99.30	99.16	99.06
6	99.87	99.84	99.77
7	99.70	99.46	99.32
8	99.84	99.65	99.56

Except for the **PS** images read by device #6, the accuracy percentages decrease with page quality. Since page quality was assigned according to device accuracy when reading **RW** images, the reason the accuracy should deteriorate when reading the synthesized image versions cannot be noise in the image, but must be something inherent in the characters or words contained in the images.

The word confusion reports were examined to determine stopword[2] and non-stopword percentages. The ratio of non-stopwords to the total number of words increased from the best pages to the worst. The percentage of blank errors also increased. This indicates that device lexicons may be causing a large proportion of the errors. When a misclassification is made, the device lexicon attempts to divide

[2]Stopwords are common words which are not normally used in text retrieval searches, such as 'a', 'of', 'the'.

Table 6.8: Character Accuracies by Page Quality - **PS** Images

Device	Best	Middle	Worst
1	99.91	99.87	99.82
2	99.85	99.75	99.26
3	99.62	99.55	99.43
4	99.89	99.78	99.73
5	99.74	99.64	99.58
6	99.64	99.81	99.56
7	99.86	99.76	99.68
8	99.66	99.50	99.39

the characters into two or more words by adding a blank. This is more likely to happen when large words are in the text, and stopwords are usually larger than non-stopwords. Device accuracy when reading non-stopwords is always lower than word accuracy[Rice93a].

Optimizing the Number of Synthesized Images

The question arises as to whether the number of images chosen to synthesize can somehow be optimized. The confusion data from the zones with 1000 or more characters was extracted and correlations computed in table 6.9. (There were 107 of the 242 zones meeting this requirement. They contained 227,573 characters, or 82% of the sample total.)

For the **S** images, the correlation with the **RW** images for characters and words changed to 0.36 and 0.52 respectively. But the correlation of the **PS** images remained at 0.96 and 0.96. This indicates that a smaller set of **PS** images would suffice to predict device accuracy. (Plots of the data along with best fit lines are at Appendix E.)

Table 6.9: Character and Word accuracies for zones with 1000 or more characters

Device	Character accuracy		Word accuracy	
	S	**PS**	**S**	**PS**
1	99.93	99.88	99.80	99.63
2	99.77	99.77	99.28	99.27
3	99.62	99.56	99.05	98.63
4	99.82	99.83	99.65	99.65
5	99.20	99.68	96.72	98.97
6	99.83	99.65	99.49	99.17
7	99.52	99.80	99.26	99.35
8	99.72	99.56	99.38	99.00

Chapter 7

Conclusions

Synthesizing page images from an ASCII text file is a cost effective way to create data that can be used to test OCR devices.

A high correlation was shown to exist between the character and word accuracies obtained from **PS** and **RW** images. The high correlation was sustained when the **PS** images were compared with subsets of the **RW** images formed by (1) dividing the images by page quality and (2) taking an arbitrary set consisting of 82% of the total characters.

There was little correlation, however, between the **S** or **PS** images misclassifications and those that occurred from the **RW** images. The **S** and **PS** misclassifications were mostly chronic errors, while the **RW** errors were spread out over more characters. The reason is probably that the devices are not trained to read the "thin" characters produced in the synthesizing process, but are optimized to read characters that have some type of "noise" added. When that is missing, touching strokes are sometimes separated or broken characters are anticipated. Device lexicons then compound the errors by adding blanks to force the output of recognizable words.

More research into the use of synthesized images is needed to explain why the accuracies correlate and the misclassifications do not.

Future Work

A mathematical noise model should be developed that will allow researchers to control the type and variety of noise that is introduced into the synthesized images. Of course, such a model should be designed to emulate the noise produced by the printing and scanning processes.

Bibliography

[Adobe90] Adobe Systems Inc. *PostScipt Language Reference Manual, Second Edition* Addison-Wesley, New York, 1990.

[Aldus92] Aldus Developers Desk. *TIFF Revision 6.0* Aldus Corporation, Seattle, Wash., June 1992.

[Baird93] H.S. Baird. Calibration of Document Image Defect Models In *Proceedings of the Second Symposium on Document Analysis and Information Retrieval*, pages 1-16. Las Vegas, Nev., April 1993.

[Baird92] H.S. Baird. Document image defect models. In *Structured Document Image Analysis*, ed. by H.S. Baird, H. Bunke and K. Yamamoto, pages 546-556. Springer-Verlag, New York, N.Y. 1992.

[Bayer92] T. Bayer, J. Hull, G. Nagy. Character Recognition: SSPR'90 Working Group Report. In *Structured Document Image Analysis*, ed. by H.S. Baird, H. Bunke and K. Yamamoto, pages 546-556. Springer-Verlag, New York, N.Y. 1992.

[Bokser92] Mindy Bokser. Omnidocument Technologies *Proceedings of the IEEE*, Vol. 80, No. 7, pages 1066-1078, July 1992.

[Bradford92] R.B. Bradford, T.A. Nartker, and B.A. Cerney. A preliminary report on UNLV/GT1: A database for ground-truth testing in document analysis and character recognition. In *Proceedings of the First Symposium on Document Analysis and Information Retrieval*, pages 300-315, Las Vegas, Nev., March 1992.

[Dickey91] L.A. Dickey. Operational factors in the creation of large full-text databases. In *DOE Infotech Conference*, Oak Ridge, Tenn.,1991.

[Grover93] K.O. Grover, J. Kanai, T.A. Nartker, and S.Rice. The ISRI IMage File Format, Version 1. *ISRI Technical Report 93-09*, Univ. of Nev., Las Vegas, Jul. 1993.

[Jenkins93a] F. Jenkins and J. Kanai A Keyword - Indexed Bibliography of Character Recognition and Document Analysis (Revision 2.0) *ISRI Technical Report 93-07*, Univ. of Nev., Las Vegas, Apr. 1993.

[Jenkins93b] F. Jenkins, J. Kanai and T.A. Nartker Using Ideal Images to Establish a Baseline of OCR Performance *ISRI 1993 Annual Report*, Univ. of Nev., Las Vegas, pages 47-54, Las Vegas, Nev., April 1993.

[Kanai93] J. Kanai, T.A. Nartker, S.V. Rice, G. Nagy. Performance Metrics For Printed Document Understanding Systems. To be presented at ICDAR 93.

[Kanungo92] T. Kanungo, I. Phillips, and R.M. Haralick. Document Degradation Module Design Specifications, Version 1.0. Unpublished paper, Univ. of Wash, Nov. 1992.

[Lamport86] L. Lamport *Latex User's Guide & Reference Manual.* Addison-Wesley, Menlo Park, 1986.

[Micro91] *User's Guide: Microsoft Word for Windows.* Microsoft Corp.,1991.

[Nagy92a] George Nagy. What does a Machine Need to Know to Read a Document? In *Proceedings of the Second Symposium on Document Analysis and Information Retrieval*, pages 1-10, Las Vegas, Nev., April 1993.

[Nagy92b] George Nagy. Teaching A Computer To Read In *11th IAPR Intl. Conf. on Pattern Recognition, Vol II*, pages 225-229,The Hague, Aug-Sep, 1992.

[Rice93] Stephen V. Rice. The OCR experimental environment, version 3. *ISRI 1993 Annual Report*, Univ. of Nev., Las Vegas, pages 83-86, Las Vegas, Nev., April 1993.

[Rice93a] Stephen V. Rice, Junichi Kanai, and Thomas A. Nartker. An evaluation of OCR accuracy. *ISRI 1993 Annual Report*, Univ. of Nev., Las Vegas, pages 9-20, Las Vegas, Nev., April 1993.

[Rice92] Stephen V. Rice, Junichi Kanai, and Thomas A. Nartker. A Report on the Accuracy of OCR Devices. *ISRI Technical Report 92-02*, Univ. of Nev., Las Vegas, March, 1992

[Rubenstein88] Richard Rubenstein. *Digital Typography, An Introduction to Type and Composition for Computer System Design.* Addison-Wesley, New York, 1988.

[Schalkoff92] Robert Schalkoff *Pattern Recognition: Statistical, Structural and Neural Approaches*, John Wiley, New York, 1992.

[Srihari86] S.N. Srihari. Document Image Understanding In *Proc. of the ACM-IEEE Computer Society 1986 Fall Joint Computer Conf.*, Dallas, Nov. 1986.

[Walpole89] Ronald E. Walpole and Raymond H. Myers. *Probability and Statistics for Engineers and Scientists*, 4th ed., Macmillan, New York, 1989.

Appendix A

Chronic Errors in S images

If a device misclassified a character more than 50% of the time, it is listed below. Missed or added blank characters are not included in this list.

Device	typeface	type size	error
4	Courier	10pt	'I' read as 'T'
5	Courier	10pt	'c' read as 'o'
			'g' read as 'q'
			'N' read as 'IV'
		12pt	'N' read as '1\T'
		14pt	'N' read as '1\T' or 'l\T'
	Times	14pt	'W' read as 'Vlf'
7	Helvetica	10pt	',' had no output
			'9' read as '0'
		12pt	',' had no output
	Times	14pt	'H' read as 'LI' or 'Il' or 'Li' or 'II'
			or 'Il' or '11'
8	Times	12pt	'8' output a non-ASCII character

Appendix B

Aggregate Accuracies

Table B.1: Character Accuracies for all 3 Sizes - **S** Images

Device	Courier	Helvetica	Times	Aggregate
1	99.93	99.90	99.91	99.91
2	99.92	99.49	99.87	99.76
3	99.78	99.14	99.82	99.58
4	99.84	99.67	99.86	99.79
5	98.01	99.72	99.79	99.17
6	99.90	99.63	99.95	99.83
7	99.83	98.95	99.72	99.50
8	99.69	99.68	99.67	99.68

Table B.2: Character Accuracies for all 3 Sizes - **PS** Images

Device	Courier	Helvetica	Times	Aggregate
1	99.93	99.88	99.79	99.87
2	99.89	99.58	99.76	99.74
3	99.69	99.13	99.78	99.53
4	99.88	99.69	99.83	99.80
5	99.64	99.61	99.71	99.65
6	99.38	99.77	99.86	99.67
7	99.83	99.67	99.80	99.77
8	99.64	99.37	99.53	99.51

Table B.3: Word Accuracies for all 3 Sizes - **S** Images

Device	Courier	Helvetica	Times	Aggregate
1	99.86	99.74	99.77	99.79
2	99.66	98.78	99.64	99.69
3	99.66	97.70	99.66	99.01
4	99.61	99.50	99.74	99.62
5	91.82	99.22	99.72	96.92
6	99.20	99.05	99.85	99.37
7	99.53	99.20	98.87	99.20
8	99.73	98.69	99.49	98.97

Table B.4: Word Accuracies for all 3 Sizes - **PS** Images

Device	Courier	Helvetica	Times	Aggregate
1	99.90	99.69	99.28	99.62
2	99.65	98.71	99.39	99.25
3	99.44	97.03	99.35	98.61
4	99.79	99.34	99.70	99.61
5	99.21	98.67	98.90	98.93
6	98.54	99.41	99.62	99.19
7	99.46	99.11	99.31	99.29
8	99.43	98.56	98.77	98.92

Appendix C

Examples of Character Images

The following figures show the images of some **S** characters that were misrecognized and the corresponding **PS** versions. These figures are magnified 4 times horizontally and vertically.

The **S** image 'h' in 10pt Courier shown in Figure 2 was confused with 'b' by device #6 6% of the time, but it never missed the **PS** version 'h'.

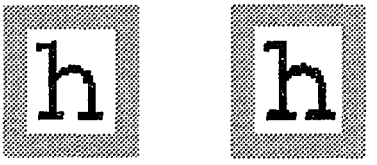


Figure C.1: **S** and **PS** 'h'

Device #7 misclassified the **S** 'H' in Figure 3 as 'LI' or 'Il', but not the **PS** 'H".

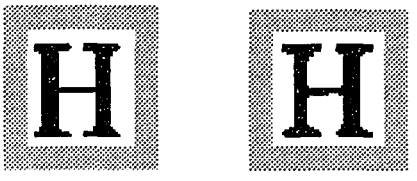


Figure C.2: **S** and **PS** 'H'

Device #7 misread the **S** image '9' as a '0' in 10pt Helvetica 75% of the time, but never made the same mistake on **PS** images.

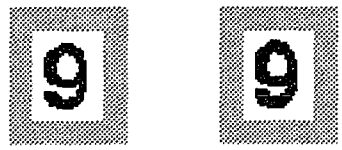


Figure C.3: **S** and **PS** '9'

Device #4 misread 54% of the **S** image 'I' as 'T' in 10pt Courier, but read all the **PS** image 'I' correctly.

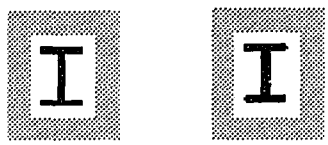


Figure C.4: **S** and **PS** 'I'

Device #5 misclassified all of the **S** image 'c' as 'o' in 10pt Courier, but only missed 1 on the **PS** images.

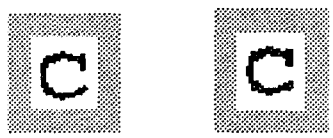
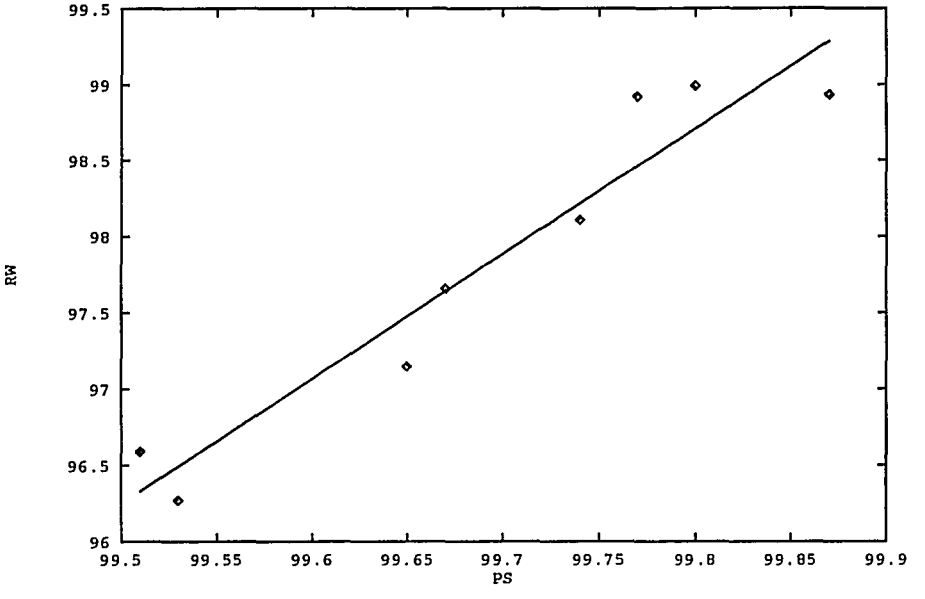


Figure C.5: **S** and **PS** 'c'

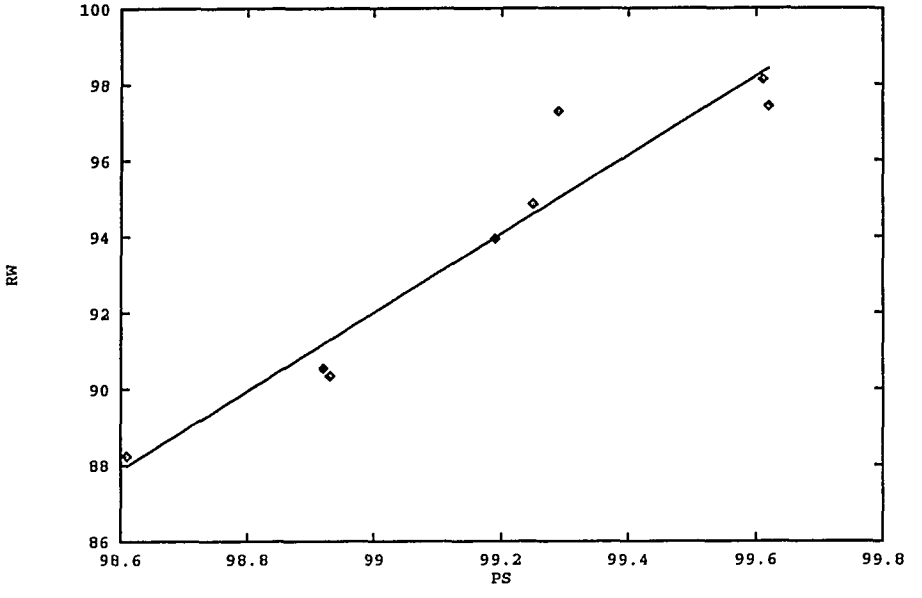
Appendix D

Best Fit Lines for PS Image accuracies

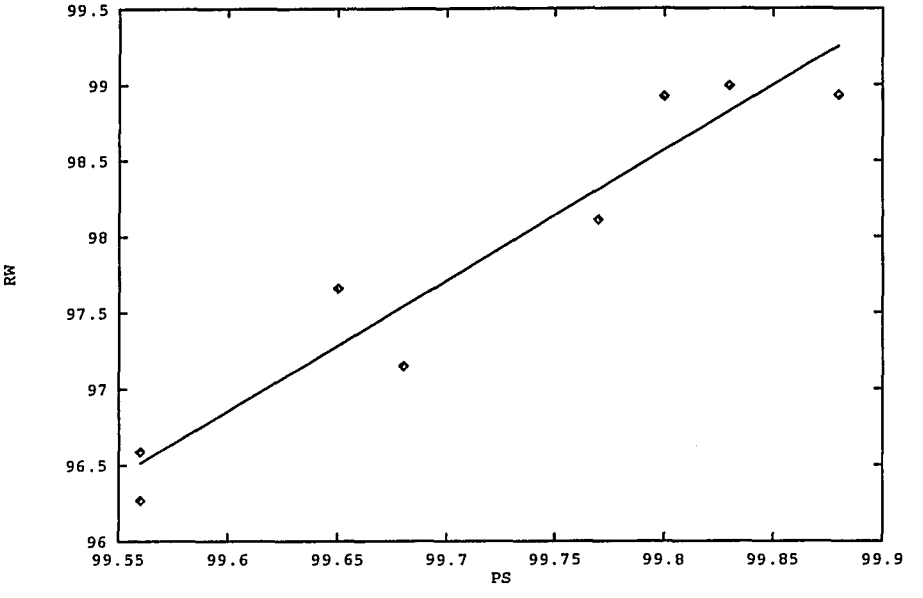
The following graph shows the correlation and best fit line between aggregated character accuracies comparing **PS** images and **RW** images for the eight devices.



The following graph shows the correlation and best fit line between aggregated word accuracies comparing **PS** images and **RW** images for the eight devices.



The following graph shows the correlation and best fit line between aggregated character accuracies for zones with 1000 or more characters comparing **PS** images and **RW** images for the eight devices.



The following graph shows the correlation and best fit line between aggregated word accuracies for zones with 1000 or more words comparing **PS** images and **RW** images for the eight devices.