P. R. Krishnaiah and L. N. Kanal, eds., *Handbook of Statistics, Vol. 2*
©North-Holland Publishing Company (1982) 621–649

Optical Character Recognition—Theory and Practice*

George Nagy

1. Introduction

This article presents an overview of optical character recognition (OCR) for statisticians interested in extending their endeavours from the traditional realm of pattern classification to the many other alluring aspects of OCR.

In Sections 2–5 the most important dimensions of data entry are described from the point of view of a project manager considering the acquisition of an OCR system; major applications are categorized according to the type of data to be converted to computer-readable form; optical scanners are briefly described; and the preprocessing necessary before the actual character classification can take place is discussed.

Section 6 outlines the classical decision-theoretic formulation of the character classification problem. Various statistical approximations to the optimal classifier, including dimensionality reduction, feature extraction, and feature selection are discussed with references to the appropriate statistical techniques. Parallel classification methods are contrasted with sequential methods, and special software and hardware considerations relevant to the various methods are mentioned.

Section 7 expands the scope of the previous discussion from isolated characters to the use of the contextual information provided by sequences of characters. References are given to available collections of statistical data on letter and word frequencies.

The paramount importance of accurate estimation of the error and reject rates is discussed in Section 8. A fundamental relation between the error rate and the reject rate in optimal systems is described, and the advantages and disadvantages of various experimental designs are discussed. Major sources of error in OCR are pinpointed and operational error rates for diverse applications are cited.

The bibliography contains references to authors specifically cited in the text as well as to selected background reading.

*Article submitted in August 1978.

2. OCR problem characterization

Among the elements to be considered in the conversion of manual data-entry operations to OCR are the permissible error rate; data volume, load distribution and displacement costs; document characteristics; and character style and quality.

2.1. *Error rate*

The constituents of the error rate are the *undetected substitution rate* and the *reject rate*. The cost of substitution errors is difficult to assess. When queried, customers respond that they cannot tolerate any undetected errors. A realistic baseline figure may be provided by key data entry rates: on typed plain text the substitution rate (before verification) runs to about 0.1%. It is considerably higher on 'meaningless' data such as lists of part-numbers, and even higher on hand-printed forms (where verification may not reduce the error rate appreciably).

On low-volume systems rejected characters may be displayed to the operator on a screen for immediate correction. On high-volume systems, however, the resulting loss of throughput would be intolerable, and documents with rejected characters are merely stacked in a separate bin for subsequent manual entry. Reject rates on commercial systems are therefore frequently quoted on a per-line or per-document basis, and the cost of rejects must be calculated from the cost of manual entry of the entire line or document. The computer attached to most current OCR systems takes care of merging the corrected material with the rest of the data to provide an essentially error-free file.

Commercial systems are normally adjusted to run at an equivalent per-character reject-to-substitution rate of 10 : 1 or 100 : 1. The substitution rates are two or three orders of magnitude below those reported in the academic literature for similar material.

2.2. *Data volume*

The early OCR systems, like early computers, ran to hundreds of thousands and even millions of dollars, and were economical only if they were able to displace dozens or hundreds of keypunch operators. Current systems are economical for data volumes as low as those corresponding to four or five manual stations. A rule of thumb for manual data entry, regardless of the type of device employed (keypunch, key-to-disk, key-to-tape, display terminal), is two keystrokes per second. At the low end of the OCR spectrum, hand-held scanners (quoted currently at below $1000 per system!) are used for shelf-inventories, library check out, retail sales tickets, and other applications requiring only a restricted character set.

2.3. *Document format*

Since the document transport accounts for a major portion of the cost of the larger OCR systems, the data density on the document is an important considera-

tion. Turn-around documents with stylized characters, such as credit-card slips, have only two or three machine-readable lines per document, and the document transport must be able to carry several dozen documents per second under the read head. Fortunately such documents are small and have rigid constraints on size and paper quality. The read-field format is usually a programmable function, and may be changed for each batch of documents, sometimes by means of a machine-readable format field.

Page readers move only one or two documents per second, and each typed page may contain up to 2000 characters. The vertical motion of the page is often combined with the line-finding function. Perhaps the most demanding transport requirements are those imposed by mail-sorting applications: size, thickness, paper quality, and address location on the envelope are virtually uncontrollable.

Document transports are often combined with imprinters and sorters. The imprinter marks the document (using either an OCR font or a bar code) for subsequent ease of automatic routing in a manner similar to the imprinting of checks with MICR (magnetic ink character recognition) codes by the first bank handling the check. The sorter divides the documents according to the subsequent processing necessary for each type of document (including reject entry).

2.4. Print quality

The design of the transducer, character acquisition, and classification systems depends mainly on the character quality (see I.S.O. Standard R/1831: Printing Specification for OCR). Some aspects of quality are independent of the spacing and shape of the characters: these include the reflective properties of the paper and of the material to be read, the presence of extraneous material on the document, and the characteristics of the printing mechanism itself such as edge definition and density variations within the symbols. Other aspects include the disposition of the symbols on the page (line and character spacing and alignment), variations in character size, the ratio of character size to average stroke width, the variability of stroke-width within the symbols, the number of classes, and the degree of differentiation among the most similar character pairs.

The parameters discussed in this section tend to be relatively similar within a given class of applications. Commercial systems are therefore developed for each class, and only minor adjustments are necessary to tune them to a specific task. The next section describes six major classes of data-entry applications in decreasing order of suitability for automation using currently available techniques.

3. Applications

3.1. Stylized fonts

Stylized typefaces are designed specifically for ease of machine recognition. Consequently they are used mainly on turn-around documents where the organization responsible for converting the data to computer-readable form has full

Fig. 1. OCR-A font.

control over document preparation. Typical applications are credit card slips, invoices, and insurance application forms prepared by agents. Special accurately machined typeheads for certain stylized fonts are available for ordinary typewriters which must, however, be carefully aligned to satisfy the OCR-reader manufacturers' specifications with regard to character skew, uniformity of impression, and spacing. Standards have also been promulgated with respect to acceptable ink and paper reflectance, margins, and paper weight.

Among the typefaces most popular in the United States is the 64-symbol OCR-A font (Standard USA 5X3.17) which is available in three standard sizes (Fig. 1). In the United States, OCR standards are set by the A.N.S.I.X3A1 Committee. In Europe the leading contender is the 113-symbol OCR-B font (Standard ECMA-11) which is aesthetically more pleasing, and which also includes lower-case letters (Fig. 2). Some OCR devices are capable of reading only subsets of the full character set, such as the numerals and five special symbols.

3.2. Typescript

With a well-aligned typewriter, carbon-film ribbon, and carefully specified format conventions, single-font typewritten material can be readily recognized by machine. Sans-serif typefaces generally yield a lower error rate than roman styles, because serifs tend to bridge adjacent characters. When specifying the typeface, it

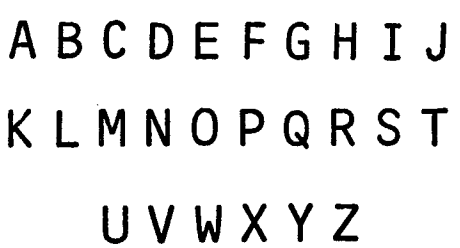
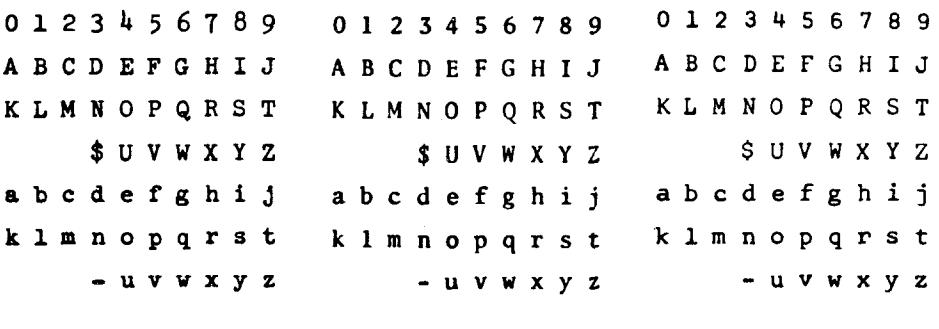
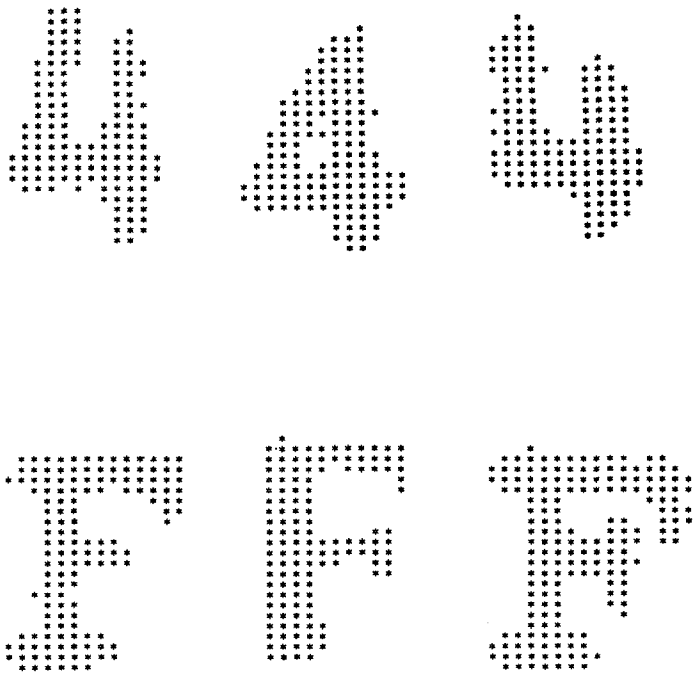


Fig. 2. OCR-B font (upper case only).

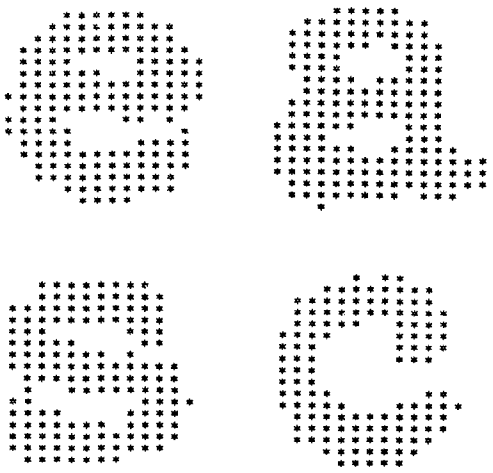


(a)



(b)

Fig. 3. Typewritten characters. (a) Elite, Adjudant, and Scribe fonts. (b) Digitized versions of '4' and 'F' from the three font styles shown above. (c) Examples of difficult to distinguish lower case letters.



(c)

Fig. 3 (continued).

is important to name the manufacturer as well: Underwood's Courier may not be identical in design to Olympia's. Styles suitable for OCR will have a distinction between lower-case l and the numeral one, and between upper-case O and zero.

Multi-font typewritten material commonly yields error rates in excess of 0.1% because the shape of individual letters varies so much among the 2000 or so typestyles available on U.S. typewriters (Fig. 3). Recognition logic has been, however, designed for up to a dozen intermixed fonts. An example of a well-established successful application is the OCR system operated by the State of Michigan for reading typed State Medicaid billing forms submitted by upwards of 25,000 individual providers of health services.

Only two different horizontal spacings are in common use. Elite or twelve-pitch fonts have 12 characters to the inch; pica or ten-pitch fonts have 10 characters to the inch. The words 'elite' and 'pica', incidentally, have a slightly different meaning in the context of typeset text. Typewriters with proportional spacing increase the difficulty of character segmentation, and normally yield error rates many times higher than fixed-pitch material. The most common vertical spacing is six single-spaced lines per inch, but material typed for OCR is almost always double-spaced. A scanning resolution of 0.004″ is sufficient for most typewritten material.

Important applications include data entry, mail sorting, and manuscript preparation using word-processing systems.

3.3. Typeset text

The major application for reading typeset text is automatic (computerized) information retrieval. Examples are searching the three million U.S. patents

issued to date, retrieving relevant federal and state laws, and converting library catalog cards to computer readable form. Most of these applications may, however, eventually disappear when computerized typesetting becomes so prevalent that all newly published material is simultaneously made available in computer readable form as a byproduct of typesetting.

The greatest problems with automatic reading of typeset material are the immense number of styles (3000 typefaces are in common use in the United States, each in several sizes and variants such as italic and boldface), and the difficulty of segmenting variable-width characters. The number of classes in each font is also larger than the 88 normally available on typewriters: combinations of characters (called 'ligatures') such as fi, ffi, and fl usually appear on a single slug, and there are many additional special symbols.

A resolution of about 0.003″ is necessary for segmenting and recognizing ordinary bookface characters (Fig. 4). The higher resolution is necessary because the characters are smaller than in typescript, because of the variability in stroke width within a single character (for instance, between the round and straight parts of the letter 'e'), and because of the tight character spacing.

3.4. Handprinted characters

The recognition rate obtained on handprinted characters depends on the number of writers, on the number of character classes, and on the training of the

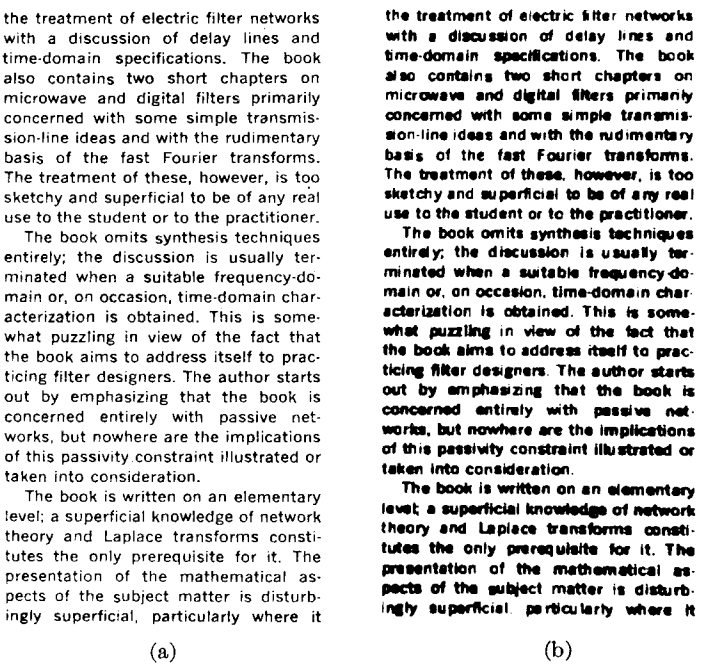


Fig. 4. Typeset text. (a) Original. (b) Photoplotter output after digitization.

writers (see Handprint Standard X3.45). Applications include filling out short forms such as driver's license applications, magazine renewals, and sales' slips. At one time automatic interpretation of coding forms was considered important, but this application has lost ground due to the spread of interactive programming. Individual experimenters can usually learn to print consistently enough to have their machine recognize their own characters with virtually zero error. Although many of the handprinted character recognition devices described in the literature are trainable or adaptive in nature, in some successful applications most of the adaptation takes place in the human. An example is the operational and successful Japanese ZIP-code reader, where the characters are carefully printed in boxes printed on the envelope.

3.5. Cursive writing

The recognition of cursive writing is not yet approaching commercial maturity and serves primarily as an experimental vehicle for studying context-dependent classification methods. It has many analogies with the recognition of continuous speech, but because it lacks the wide applicability of the latter, the recognition of cursive writing has had far less effort devoted to it. Recently, however, several large-scale experiments have been published on signature verification with on-line capture of the stylus motion. Typical results show a rejection rate of 3% and a false acceptance rate of 0.2% among a population of 300 writers including amateur forgers.

3.6. Special alphabets

Among special alphabets of interest in optical character recognition are cyrillic characters, Chinese and Japanese characters, and map symbols. The classification of printed and typed cyrillic characters differs little from that of latin characters; the most difficult pair is usually Ш and Щ. The recognition of Chinese ideographs is complicated by the immense number of classes (5000 symbols in ordinary newspapers) and the complexity of individual characters. A simplifying factor, however, is the fact that only half-a-dozen different typefaces are in common use. Japanese OCR requirements include Kanji characters (essentially the Chinese ideographs), Katakana characters (55 different symbols), as well as the latin alphabet. Because typewriters are still much less prevalent in the Far East than in the West, the classification of handprinted characters takes on added importance.

4. Transducers

On examination under a magnifying glass, most printed material intended for optical character recognition appears well defined and relatively simple to classify. Digitization by means of an optical scanner, however, frequently obliterates the significant distinctions between patterns corresponding to the different classes, and introduces distortions which greatly complicate recognition. The technology

for accurate and faithful conversion of grey-scale material to digital form does of course exist, but the necessary apparatus—flat-bed and rotating drum micro-densitometers—is far too expensive for most academic research programs and far too slow for commercial application. The challenge in optical character recognition is, in fact, to classify characters as accurately as possible with the lowest possible spatial quantization and the minimum number of grey levels in the transducer. Most OCR scanners therefore operate at a spatial resolution barely sufficient to distinguish between idealized representations of the classes and convert all density information to binary—black and white—form.

The remainder of this section outlines the principal characteristics of optical transducers used for character recognition either in the laboratory or in the field.

4.1. Device types

Optical scanners may be divided into *flying spot* devices, where successive portions of the document are illuminated in turn and all of the reflected or transmitted light is collected to determine whether the illuminated spot was black or white, and *flying aperture* devices, where the entire document is illuminated but light is collected only from a single spot. It is also possible to combine the two methods and both illuminate and observe only a single spot at a time; this expensive arrangement results in greatly improved signal-to-noise ratio and therefore more accurate grey-scale quantization.

Scanners may also be categorized according to whether they operate with light *reflected* from a document or light *transmitted* through a transparent image of the document. Since the optical design of transparency scanners is somewhat simpler they are sometimes used in laboratory research, but photographing every document is not generally practicable in a production environment.

The devices themselves may be crudely classified according to the mechanism used to address successive portions of the document: *mechanical*, *television camera*, *cathode-ray tube*, and *solid state* scanners are the most common. Some scanners utilize hybrid combinations of these basic types.

Most commercial OCR systems use mechanical motion of the document in the vertical direction to scan successive lines of print. The earlier machines also used mechanical motion to scan across each line by means of a rotating or oscillating mirror or a prism assembly. The information from a single vertical column in a character, or even from an entire character, was collected by means of a photocell array. The development of fiber optics added another option to the design of mechanical systems. Well-designed mechanical scanners have excellent geometric properties and virtually unlimited resolution, but are restricted to rigid scan patterns. Because of the necessity for immunity to vibration and the need for extremely rapid motion for high throughput, they are also very expensive.

Television cameras in a flying-aperture reflection-scanner configuration are a favorite device for laboratory pattern recognition research, but their geometric resolution is barely sufficient for page-reading applications. Most laboratory cameras have poor linearity and a resolution of only about 400×400 elements, in

contrast to the 2000×2000 array required for a printed page. Attempts to combine mechanical movement with commercially available television cameras have not proved successful.

Both research and commercial systems have made use of cathode-ray tube flying spot scanners, where each spot on the document is illuminated in turn by the glow of phosphor from the corresponding spot on the screen of the tube and all of the reflected or transmitted light is collected by means of a photomultiplier tube array. Although the scan pattern of such systems is extremely versatile, since the deflection plates can be placed under direct computer control, careful—and expensive—optical design is required to provide acceptable geometric and densitometric fidelity. The geometric resolution is just barely sufficient for page scanners, but the positional linearity can be considerably increased by incorporating feedback from a built-in reseau pattern.

Most current systems use solid-state scanners. Flying spot devices use linear or two-dimensional light-emitting diode (LED) arrays. Chips with 4096×1 and 256×256 arrays are already available commercially. Flying-aperture devices use arrays of photodiodes or phototransistors in similar configurations. Self-scanned arrays, where the addressing mechanism necessary to activate successive devices is built into the chip itself, simplify the electronic design. The optical design is also relatively simple since current microfabrication techniques ensure geometric fidelity and produce devices small enough to be used at 1 : 1 magnification or less. The amplitude response of all the elements within a chip is typically within a few percent of the mean.

4.2. Geometric characteristics

The major geometric characteristics of optical scanners are:

(1) *Resolution*. The exact measurement of the two-dimensional optical transfer function is complex, but for OCR purposes one may equate the effective spot size to the spatial distance between the 10% and 90% amplitude-response points, as a black–white knife-edge is advanced across the scanning spot. Standard test charts with stripe patterns are also available to measure the modulation as a function of spatial frequency.

(2) *Spot shape*. A circular Gaussian intensity distribution is normally preferred, but a spot elongated in the vertical direction reduces segmentation problems.

(3) *Linearity*. This characteristic, which may be readily measured with a grid pattern, is important in tracking margins and lines of print. Solid-state devices are inherently linear, but cathode-ray tubes require pin-cushion correction to compensate for the increased path-length to the corners of the scan field. Skew is sometimes introduced by the document transport.

(4) *Repeatability*. It is important to be able to return exactly to a previous spot on the document, for example to rescan a character or to repeat an experiment. Short-term positional repeatability is generally more important—and better—than long-term repeatability.

4.3. Photometric characteristics

(1) *Resolution.* The grey-scale resolution may also be measured in different ways. To some investigators it means the number of grey levels that may be reliably discriminated at a specific point on the document; to others it means the number of discriminable grey levels regardless of location. The latter interpretation is considerably more stringent, since a flat illumination or light collection field is difficult to achieve without stored correction factors. Quantum noise also affects the grey-scale resolution, but the signal-to-noise ratio may be generally increased at the expense of speed by integrating the signal.

(2) *Linearity.* Grey-scale linearity is meaningless for binary quantization, but if several grey-levels are measured, then an accurately linear or logarithmic amplitude response function may be desirable. Standard grey-wedges for this measurement are available from optical suppliers.

(3) *Dynamic range.* For document scanners a 20:1 range of measurable reflectance values is acceptable. Transparency scanners, however, may provide adequate response over 3.0 optical density units (1000:1).

(4) *Repeatability.* It is important that the measured grey level be invariant with time as well as position on the document.

(5) *Spectral match.* The spectral characteristics of the source of illumination and of the detector should be closely matched to those of the ink and of the paper. The peak response of many OCR systems is in the invisible near-infrared region of the spectrum.

4.4. Control characteristics

Most OCR devices operate in a rigid raster scan mode where the scan pattern is independent of the material encountered. Line followers, however, track the black lines on the page and have been used principally for the recognition of hand-printed characters. Completely programmable scanners are invaluable for experimentation, but have become a strong commercial contender only with the advent of microprocessors. Programmable scanners allow rescanning rejected characters, increase throughput through lower scan resolution in blank areas of the page, and reduce storage requirements for line-finding and character isolation.

5. Character acquisition

The conversion of the pattern of black-and-white areas constituting an entire document into the set of smaller patterns which serve as the input to the character classification algorithm has not received much attention in the literature. In academic investigations the challenging problems associated with this area of OCR are usually circumvented either through the utilization of specially prepared and formatted documents or through manual methods of isolating the characters. Needless to say, neither of these approaches is practicable for operational OCR machines.

The *preprocessing* necessary for classification must satisfy two basic requirements. The first requirement is to locate each character on the page in an order that 'makes sense' in the coded file which eventually results from classification. The second requirement is to present each isolated character to the recognition algorithm in a suitable form.

5.1. Format control and line finding

The difficulty of locating each character in succession depends, of course, on the application at hand. With stylized characters the format and line spacing are usually rigidly controlled. Material which need not be considered is often printed in ink with spectral characteristics which are invisible to the optical scanner, and special symbols guide the scanner to each field of data. Adequate spacing is provided to eliminate any horizontal or vertical interference between characters.

With typeset material, however, format problems may be almost insurmountable without some form of human intervention. Consider, for example, a two-column magazine article interrupted by a two-column wide illustration. Even if the system recognizes correctly the presence of the illustration after processing the top left column, should processing continue in the left-hand column below the illustration or at the top line of the right hand column? Other challenging problems occur in scanning formulas, tables, and alphanumeric material on technical drawings and maps.

None of these tasks can be performed with currently available commercial systems. Satisfactory line-finding algorithms are, however, available. When the printed lines are straight and widely spaced, almost any algorithm will work.

Closely spaced lines may be located by projecting the grey-level distribution on the vertical axis. For typeset and typewritten serif material, there are usually four peaks to every line: two major ones at the top and bottom of the lower case characters, and two minor ones at the ends of the ascenders and descenders. Because the peaks tend to spread out if the material is not perfectly aligned with the axes of the digitizer, usually only a column of material a few inches wide is used for line location. Modifications are then necessary to locate short lines. Fourier transform methods have also been proposed for line and character location, but the periodicities of printed matter are seldom regular enough for this purpose. Adaptive line following algorithms are capable of compensating for baseline drift in the lines of characters. Individual character misalignments are usually addressed after character isolation.

5.2. Character isolation

Imperfect separation between adjacent characters accounts for a large number of misclassifications, although in experimental studies segmentation errors are often omitted from the reported classification error rate. With stylized characters and with typewritten material produced by well-adjusted typewriters the expected segmentation boundaries can be accurately interpolated by correlating an entire line with the many easily detectable boundaries that occur between narrow

characters. The problem is, however, more complicated if the geometrical linearity and repeatability of the transducer itself is low relative to the character dimensions.

If the scanner geometry is undependable (CRT scanners) or if the characters are not uniformly spaced (typeset material), then there is no alternative to character-by-character segmentation. This, in turn, requires a scanning aperture two or three times smaller than that required for recognition to find narrow zig-zag white paths between adjacent characters. Ad hoc algorithms depending, for instance, on 'serif suppression', are used to separate touching characters. The expected fraction of touching character pairs—as measured directly on the document using ten-fold magnification—depends on the type-style, type-size, printing mechanism, and on the ink-absorbing characteristics of the paper, but typescript or printed material (serif fonts) with up to 10% touching characters is not uncommon. The worst offender in this respect is probably the addressograph machine: with a well-inked ribbon one may seldom see an unbridged pair of characters within the same word.

Some character pairs simply cannot be segmented unless the individual components are first recognized. Current classification algorithms, however, generally require isolated character input. The development of optimal 'on-the-fly' recognition, combining segmentation with classification, is one of the major challenges facing OCR.

5.3. Normalization, registration, and centering

Once the characters are isolated, it is necessary to transform them into the form expected by the recognition algorithm. Generally, the more sophisticated the algorithm, the less it is affected by difference in the size, orientation, and position of the characters. Simple template matching algorithms, such as those used in many commercial OCR systems, require that variations of this type be eliminated before classification.

Since the size of the array processed by the recognition system is usually of fixed size (say 20×30), it is advantageous to have each pattern fill as much of the array as possible. Size normalization algorithms either expand or contract the pattern until it just fits into the array, or adjust the pattern so as to fix the horizontal and vertical second moment (standard deviation). Size normalization is useful for multifont type-script, for printed matter, and for handprinted characters.

Most character recognition algorithms are neither rotation nor skew invariant. Typewritten characters may, however, have sizeable rotation variance due to typewriter misadjustments or to document misalignment, and handprinted characters tend to exhibit very significant skew. Rotation and skew correction may be based on principal-components analysis or on enclosure of the pattern in one of a variety of geometrical figures.

Registration or centering algorithms translate each pattern to a standard position according to the location of its centroid. Computation of the horizontal

and vertical medians (the imaginary lines which divide the pattern into an equal number of black bits in each quadrant) is more economical than computation of the centroid, but both methods suffer from vulnerability to misalignments of the printing mechanism, which causes one side, or top or bottom, of the character to be darker than the other. An equally unsatisfactory alternative is lower-left-corner (or equivalent) registration after stray-bit elimination.

Misregistrations of two or three pels (i.e., picture elements) in the vertical direction and one or two pels horizontally are not unusual with the normally used scanning resolution for typewritten characters. If uncorrected, such misregistration necessitates that template matching be attempted for all possible shifts of the template with respect to the pattern within a small window (say, 7×5) in order to guarantee inclusion of the ideal position.

6. Character classification

In a classic paper, Chow derived the *minimum risk* character classification function in terms of the loss $W(a_i, d_j)$ incurred when decision d_j is made and the true class is a_i, and of the conditional probability functions $P(v|a_k)$ of observing the signal v when the class of the pattern under consideration is a_k. The derivation makes no assumptions regarding the form of the underlying distribution of v. The possibility of rejecting a character (i.e., not assigning it to any class) is considered by including a 'reject' decision d_0. Specific examples were given for the case where the observations consist of Gaussian statistically independent noise added to the ideal signal representing each class.

In the same paper, the decision criteria corresponding to the *minimum error* classification are also derived. It is shown that the optimum decision rule is not randomized in either case.

If the cost of misclassification is uniform regardless of the particular error committed, then it can be shown that the optimal decision consists of selecting the class a_k for which the a posteriori probability $P(a_i|v)$ is the largest. If v is a vector with discrete-valued components, as is usually the case in optical character recognition, then Bayes' formula allows the computation of the a posteriori class probabilities for a given observation $\bar{v}$ in terms of the conditional probability of that observation given the class $P(\bar{v}|a_k)$, the a priori probability of the class $P(a_k)$, and the overall probability of the observation $P(\bar{v})$:

$$P(a_k|\bar{v}) = P(\bar{v}|a_k)P(a_k)/P(\bar{v}).$$

Since we are seeking to maximize the a posteriori probabilities, we can eliminate $P(\bar{v})$, which is common to each term, from consideration. For the following discussion we can also forget about the a priori probabilities $P(a_k)$. If they are not all equal we can always take them into account as a weighting factor at the end. Hence the important term is $P(\bar{v}|a_k)$, the probability of observing $\bar{v}$ when the true class of the character is a_k.

Let us now consider the computational problems of estimating $P(\bar{v}|a_k)$ for all possible values of $\bar{v}$ and a_k in an OCR environment. For the sake of concreteness, let us assume that each observation $\bar{v}$ corresponds to the digitized grey values of the scan field. If the number of observable grey levels is S, the number of elements in the scan field is N, and the number of character classes is M, then the total number of terms required is $M \times S^N$. In a practical example we may have $S=16$ (the number of differentiable reflectance values), $N=600$ (for a 20×30 array representing the digitized character), and $M=64$ (upper and lower case characters, numerals, and special symbols.) The number of probability estimates required is thus $64\times16^{600}=2^{2406}\cong10^{700}$. The goal of much of the work in character recognition during the last two decades has been, explicitly or implicitly, to find sufficiently good approximations, using a much smaller number of terms, to the required probability density function.

Among the approaches tried are:

(1) Assuming statistical independence between elements of the attribute vector.

(2) Assuming statistical independence between subsets of elements of the attribute vector (feature extraction).

(3) Computing only the most important terms of the conditional probability density function (sequential classification).

In the next several paragraphs we will examine how these simplifications allow us to reduce the number of estimations required, and the amount of computation necessary to classify an unknown character. In order to obtain numerical comparisons of the relative number of computations, we shall stay with the above example.

6.1. Binary observations

Assuming that the values of each component of the observation vector v are restricted to two values does not in itself materially reduce the calculations (it decreases the number of estimates required from $M \times S^N$ only to $M \times 2^N$), but it simplifies the subsequent discussion. We shall therefore henceforth assume that the values of the reflectances are thresholded to *black* (1) and *white* (0). The value of the threshold used to differentiate black from white may in itself depend on the values of the neighboring elements or on other factors.

6.2. Statistical independence

If we assume that for a given class of characters the observations are statistically independent from one another, then we may express the a posteriori probability in product form:

$P(v|a_k)=\prod_{j=1}^{N}P(v_j|a_k)$, where v_j is the jth component of $\bar{v}$. Instead of estimating 2^N values for each class, we need estimates only for N values (since $P(v_j=1|a_k)=1-P(v_j=0|a_k)$). Furthermore, the multiplication operations necessary to compute the a posteriori probabilities can be replaced by additions by taking logarithms of both sides of Bayes' equation, as shown by Minsky. Since we

need determine only the largest of the a posteriori probabilities, and the logarithm function is monotonic, taking logarithms preserves the optimal choice.

This derivation leads to the *weighted mask* approach, where the score for each class is calculated by summing the black picture elements in the character under consideration with each black point weighted by a coefficient corresponding to its position in the digitized array. The contribution of the white points can be taken into account by the addition of a constant term. The character with the highest score is selected as the most likely choice. If none of the scores are high enough, or the top two scores are too close, then the character is rejected.

It may be noted that the weighted mask approach implemented through resistor networks or optical masks was used in experimental OCR systems long before the theoretical development was published. A special case, called *template matching* or *prototype correlation*, consists of restricting the values of the coefficients themselves to binary or ternary values; when plotted, the coefficients of the black points resemble the characters themselves. A further reduction in computation may be obtained by discarding the coefficients which are least useful in discriminating between the classes, leading to *peephole templates*.

6.3. *Restricted statistical independence*

An approach less restrictive than requiring complete statistical independence consists of assuming that certain groups of observations v_j are statistically independent of each other, but that statistical dependencies exist within each group. Each such group of variables may then be replaced by a new variable. If, for instance, the a posteriori probability is expanded as follows:

$$P(v|a_k) = P(v_i|a_k) \cdot P(v_j|v_i, a_k) \cdot P(v_m|v_i, v_j, a_k) \cdots ,$$

then one may assume that higher order dependences are insignificant, represent the statistical relations in the form of a *dependence tree*, and restrict the computation to the most important terms, as shown by Chow.

This point of view also leads to a theoretical foundation for the ad hoc feature extraction methods used in commercial systems. Straightline segments, curves, corners, loops, serifs, line crossings, etc., correspond to groups of variables which commonly occur together in some character classes and not in others, and are therefore class-conditionally statistically dependent on one another. Even features based on integral transform methods, such as the Fourier transform, can be understood in terms of statistical dependences.

Feature selection and *dimensionality reduction* methods may also be considered in the above context. Given a pool of features, each representing a group of elementary observations v_j, it is necessary to determine which set of groups most economically represents the statistical dependences necessary for accurate estimation of the a posteriori probability. The number of possible combinations of features tend to be astronomical: if we wish to select 100 features from an initial set of 1000 features, there are about 10^{100} possible combinations. Most feature

selection methods therefore use ad hoc techniques, adding or eliminating one feature at a time on the basis of some type of information-theoretic measure, such as entropy.

6.4. Sequential classification

In sequential classification only a subset of the observations v_j is used to arrive at a decision for the character identity; unused elements need not even be collected. The classification is based on a decision tree (Fig. 5) which governs the sequence of observations (picture elements) to be examined. The tree is fixed for a given application, but the path traced through the tree depends on the character under consideration.

The first element v_j to be examined, called the *root* of the tree, is the same for any new character, since no information is available yet as to which element would provide the most information. The second element to be examined, however, depends on whether the first element was black or white. The third element to be examined depends, in turn, on whether the second element was black or white. Each node in the tree, corresponding to a given observation v_j, thus has two offsprings, each corresponding to two other observations. No observation v_j occurs more than once in a path. The *leafs* of the tree are labelled with the character identities or as 'reject' decisions. Normally there are several leafs for each character class (and also for the reject decision), corresponding to the several character configurations which may lead to each classification.

Binary decision trees for typewritten characters typically have from 1000 to 10000 nodes. The path length through the tree, from root to leaf, may vary from

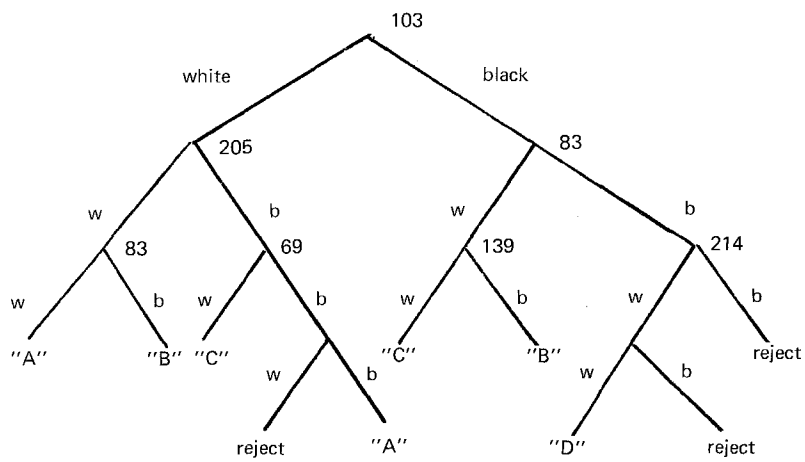


Fig. 5. Decision tree for character recognition. The numbers at the nodes represent the picture elements. The left branch is taken if the element corresponding to the node is white in the character under consideration, the right branch if it is black. The leaves yield either a definite identification of the character class or a 'reject' decision.

10 to 30 nodes. Character classes which are difficult to classify because of their resemblance to other classes require longer paths through the tree since more elements must be examined before a reliable decision can be reached. Several different methods for designing decision trees are available; they differ considerably with regard to the underlying model (deterministic or probabilistic) of classification, design complexity, required sample size, and classification accuracy.

Sequential classification based on decision trees may be applied, of course, also to features other than black or white picture elements, but the design and selection of features for sequential classification is even more complicated than for the parallel methods discussed above. Other approaches to sequential classification are based on Wald's sequential probability ratio test (SPRT), which provides the optimum stopping rule to satisfy a required minimum probability of correct classification.

6.5. *Comparison of classification methods*

Table 1 shows a comparison of several classification methods with respect to the number of operations required for classification of a single character. The table does not include any figures on recognition performance, which depends very markedly on the extent to which the design assumptions correspond to the data used for evaluation (see below). The estimates given for the example mentioned earlier, with $S = 2$, $M = 64$ and $N = 600$ are the author's, and should at best be considered tentative.

6.6. *Implementation of classification algorithms*

High-speed page readers require instantaneous classification rates of upward of 1000 characters per second for an average throughput of one document per second (a double-spaced typewritten page contains about 1500 characters). If the classification procedure is executed sequentially for each of 64 categories of characters, then only 15 microseconds at most are available to carry out the

Table 1
Comparison of statistical classification algorithms. The comparison of the computational requirements of the various classification methods is based on 64 character classes and 600 (20×30) binary picture elements per character

Method	Computational requirement
Exhaustive search	$2^{600} = 10^{180}$ comparisons (600 bits each)
Complete pairwise dependence	$\frac{1}{2} \times 64 \times 600^2 = 10^7$ logical operations and additions
Complete second-order Markov dependence	$64 \times 1200 = 80000$ logical operations and additions
Class-conditional independence (weighted masks)	$64 \times 600 = 40000$ logical operations and additions
Mask-matching (binary masks)	$64 \times 600 = 40000$ logical operations and counts
Peephole templates (30 points each)	$64 \times 30 = 1920$ logical operations and counts
Complete decision tree	600 one-bit comparisons
Decision tree—4000 nodes	$^2\log 4000 = 12$ one-bit comparisons

necessary calculations for each category. Consequently most high-speed commercial OCR machines use special hardware for recognition.

Prototype correlation and weighted masks are most often implemented by means of optical comparisons, resistor summing networks, or high-speed parallel digital logic circuitry. Position invariance is achieved by shifting the character to three, five, fifteen, or twenty-five successive positions in a one- or two-dimensional shift register. Character segmentation is performed either explicitly by comparing successive vertical scans, or implicitly by looking for peaks in the output of the recognition circuitry. On the newer machines a small general-purpose computer acts as the control unit for the entire system, performs validity checks, generates and monitors test inputs, and formats the output as required for a particular application.

Lower-speed devices, including those used in conjunction with hand-held wands, use a combination of hardwired digital logic and programmable microprocessors. Features are usually extracted in hardware, but the final classification or reject decision, which requires processing a much smaller amount of data than does the front-end, may be relegated to a microprocessor with a read-only memory. A separate user-programmable microprocessor is sometimes used for checking and formatting the output. This microprocessor may then be programmed directly through the wand itself by means of codes in a typeface which can be recognized by the unit.

7. Context

Characters that are difficult to classify by their shape alone can sometimes be recognized correctly by using information about other characters in the same document. For instance, in decoding a barely legible handwritten postcard, one must frequently resort to locating different occurrences of the same misshapen symbol. Contextual information may be used in a number of different ways to aid recognition; Toussaint lists six major categories. In this section, however, we will discuss only the use of the information available from the non-random sequencing of characters in natural-language text, including such relatively 'unnatural' sequences as postal addresses, prices, and even social security numbers.

For a sequence of observed patterns $V = \bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n$, we may again use Bayes' rule to obtain

$$P(A|V) = P(V|A)P(A)/P(V)$$

where A is a sequence of character identities A_i, $i = 1,2,\ldots,n$ and A_i takes on values a_k, $k = 1,2,\ldots,M$ (M is the number of classes).

To estimate $P(V|A)$ for all possible character sequences A of arbitrary length n clearly requires an astronomical number of calculations. Approximations for estimating the sequence are considered either *n-gram* oriented (Subsection 7.1) or *word* (*dictionary-lookup*) oriented (Subsection 7.2). Sources of the necessary

statistical information about letter frequencies are discussed in Subsection 7.3, while word frequencies are discussed in Subsection 7.4.

7.1. N-gram methods

A simplifying assumption for the calculation of the a posteriori probabilities $P(V|X)$, exploited independently by Abend and by Raviv, is to consider text as an mth order Markov source. It can be shown that under this assumption the a posteriori probabilities for the ith character in the sequence can be recursively computed in terms of the conditional probabilities of the pattern (feature) vectors of the $(i-1)$st, $(i-2)$th,..., and $(i-m+1)$st characters. A further simplification is obtained by letting the decision for the ith character depend only on the *decision* (rather than the conditional probabilities of the feature vectors) for the previous characters. Experiments show that on typed English text the error rate based on feature vectors alone can be decreased considerably by considering only the class attributed to the two characters immediately preceding the character under consideration. The statistical data necessary to accomplish this is a table of letter trigram probabilities.

7.2. Dictionary look-up

One of the earliest dictionary look-up methods is that reported by Bledsoe and Browning in 1959. Here the compound a posteriori probability of every word in the dictionary of the same length as the one under consideration is computed by multiplying together the probabilities (based on the feature vectors) of its constituent characters. The word with the highest a posteriori probability is chosen.

When the Markov assumption for letter occurrences can be justified, the *Viterbi algorithm* provides an efficient method of computing the a posteriori probability of each word in the dictionary. The computation can be further accelerated without a noticeable increase in the error rate by considering only the most likely candidates (based on the feature vectors) at each character position in the word.

The dictionary look-up algorithm can make use of the confusion probabilities (obtained empirically) between each pair of character identities instead of the calculated a posteriori character probabilities. Yet another variant is the combination of sequential feature extraction (Subsection 6.4) and the Markov assumption on the character sequences. It has been shown that there are circumstances under which an additional measurement on a previous character is preferable to an additional measurement on the character about to be classified!

7.3. Letter frequencies

The distribution of letter frequencies is, of course, of interest in deciphering cryptograms, and singlet and doublet letter frequencies are tabulated in most texts on the subject. Table 2 shows the doublet frequencies from a corpus of a 600000 characters legal text. Here again the frequencies of the less common pairs

Table 2
Bigram frequencies based on 600000 characters of legal text (×10)

		A	B	C	D	E	F	G	H	I	J	K	L
1	0.693	0.044	0.001	0.015	0.178	0.375	0.099	0.044	0.038	0.003	0.003	0.008	0.062
2 A	0.207	0.000	0.004	0.030	0.011	0.032	0.011	0.010	0.057	0.019	0.001	0.000	0.038
3 B	0.079	0.011	0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.005	0.000	0.000	0.000
4 C	0.124	0.028	0.000	0.004	0.000	0.044	0.000	0.000	0.000	0.047	0.000	0.000	0.000
5 D	0.057	0.017	0.000	0.000	0.001	0.085	0.000	0.000	0.000	0.021	0.000	0.000	0.015
6 E	0.057	0.000	0.033	0.053	0.063	0.025	0.016	0.021	0.216	0.022	0.003	0.008	0.045
7 F	0.064	0.006	0.000	0.000	0.000	0.017	0.017	0.000	0.000	0.014	0.000	0.000	0.002
8 G	0.016	0.011	0.000	0.000	0.006	0.012	0.000	0.001	0.000	0.014	0.000	0.000	0.001
9 H	0.047	0.001	0.000	0.032	0.000	0.003	0.000	0.013	0.000	0.000	0.000	0.000	0.000
10 I	0.130	0.025	0.006	0.026	0.037	0.010	0.026	0.007	0.046	0.000	0.000	0.002	0.027
11 J	0.016	0.001	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12 K	0.004	0.004	0.000	0.004	0.000	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.001
13 L	0.026	0.067	0.010	0.013	0.001	0.031	0.003	0.002	0.001	0.025	0.000	0.001	0.042
14 M	0.044	0.011	0.001	0.000	0.002	0.016	0.000	0.003	0.000	0.017	0.000	0.000	0.000
15 N	0.047	0.118	0.000	0.000	0.000	0.091	0.000	0.003	0.001	0.168	0.000	0.002	0.000
16 O	0.153	0.000	0.018	0.084	0.008	0.003	0.033	0.004	0.025	0.076	0.003	0.000	0.014
17 P	0.075	0.029	0.000	0.000	0.000	0.010	0.000	0.000	0.000	0.007	0.000	0.000	0.000
18 Q	0.004	0.000	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19 R	0.057	0.063	0.005	0.006	0.003	0.140	0.009	0.008	0.003	0.017	0.000	0.000	0.000
20 S	0.116	0.060	0.003	0.001	0.005	0.075	0.000	0.004	0.001	0.081	0.000	0.001	0.007
21 T	0.307	0.099	0.001	0.043	0.000	0.022	0.004	0.001	0.008	0.078	0.000	0.000	0.005
22 U	0.024	0.010	0.011	0.010	0.006	0.000	0.006	0.005	0.003	0.000	0.013	0.000	0.006
23 V	0.023	0.008	0.000	0.000	0.001	0.018	0.000	0.000	0.000	0.010	0.000	0.000	0.002
24 W	0.075	0.007	0.000	0.000	0.001	0.007	0.000	0.000	0.001	0.000	0.000	0.000	0.001
25 X	0.000	0.005	0.000	0.000	0.000	0.015	0.000	0.000	0.000	0.001	0.000	0.000	0.000
26 Y	0.005	0.011	0.019	0.002	0.001	0.005	0.001	0.000	0.002	0.000	0.000	0.000	0.020
27 Z	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000

depend greatly on the domain of discourse; in the example given, the frequency of the pair 'ju' (from jury, judge, judicial, jurisprudence, etc.,) is much higher than in other types of material. The most common letter in English is 'e', accounting for 13% of all letters. The most common pair is 'he'.

In studying higher-order letter frequencies, upper and lower case, blanks, and punctuation must also be considered. The letter 'q', for instance, is more common at the beginning of words than in the middle or at the end—hence 'q–' has a lower frequency than '–q'. The distribution of n-gram frequencies according to the position within the word has been studied by Shinghal and Toussaint. A study of the singlet frequencies of Chinese ideographs (which may be regarded as complete words, but represent single characters as far as OCR is concerned) is available in Chen. Japanese Katakana character frequencies are tabulated in Bird.

In deriving n-gram probabilities from a sample of text it is necessary to make special provision for estimating the frequency of n-grams which do not occur in the text at all. More generally, it is desirable to use a better estimator than the sample frequency itself. By postulating an a priori distribution for each n-gram frequency, an improved estimator may be derived either by minimizing the risk

Table 2 (continued)
Bigram frequencies based on 600 000 characters of legal text ($\times 10$)

	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	0.018	0.176	0.074	0.008	0.000	0.097	0.213	0.180	0.002	0.011	0.013	0.005	0.096
2 A	0.031	0.020	0.008	0.020	0.000	0.038	0.013	0.049	0.006	0.007	0.022	0.003	0.000
3 B	0.002	0.000	0.003	0.000	0.000	0.001	0.000	0.000	0.008	0.000	0.000	0.000	0.000
4 C	0.001	0.031	0.010	0.000	0.000	0.009	0.005	0.001	0.014	0.000	0.000	0.005	0.000
5 D	0.000	0.090	0.006	0.000	0.000	0.025	0.001	0.000	0.008	0.000	0.000	0.000	0.000
6 E	0.039	0.037	0.003	0.043	0.000	0.124	0.074	0.085	0.012	0.037	0.020	0.003	0.003
7 F	0.000	0.003	0.095	0.000	0.000	0.002	0.000	0.000	0.002	0.000	0.001	0.000	0.000
8 G	0.000	0.052	0.003	0.000	0.000	0.004	0.000	0.000	0.005	0.000	0.000	0.000	0.000
9 H	0.000	0.001	0.005	0.003	0.000	0.001	0.016	0.258	0.000	0.000	0.022	0.001	0.000
10 I	0.024	0.017	0.004	0.004	0.000	0.047	0.033	0.096	0.010	0.024	0.019	0.002	0.001
11 J	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12 K	0.000	0.002	0.001	0.000	0.000	0.006	0.001	0.000	0.000	0.000	0.000	0.000	0.000
13 L	0.000	0.004	0.017	0.017	0.000	0.003	0.003	0.003	0.017	0.000	0.001	0.000	0.001
14 M	0.011	0.001	0.028	0.001	0.000	0.011	0.002	0.003	0.005	0.000	0.003	0.000	0.001
15 N	0.000	0.005	0.143	0.000	0.000	0.008	0.000	0.002	0.023	0.000	0.004	0.000	0.000
16 O	0.015	0.032	0.005	0.027	0.000	0.043	0.016	0.067	0.001	0.003	0.006	0.000	0.002
17 P	0.012	0.000	0.013	0.028	0.000	0.007	0.008	0.000	0.013	0.000	0.000	0.002	0.000
18 Q	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19 R	0.000	0.001	0.094	0.039	0.000	0.008	0.000	0.028	0.043	0.000	0.003	0.000	0.000
20 S	0.004	0.035	0.013	0.001	0.000	0.023	0.029	0.025	0.032	0.000	0.003	0.000	0.005
21 T	0.000	0.087	0.032	0.007	0.000	0.043	0.072	0.011	0.024	0.000	0.000	0.002	0.000
22 U	0.004	0.004	0.051	0.006	0.010	0.011	0.032	0.013	0.000	0.000	0.000	0.000	0.000
23 V	0.000	0.003	0.011	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24 W	0.000	0.002	0.018	0.000	0.000	0.001	0.001	0.004	0.000	0.000	0.000	0.000	0.000
25 X	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26 Y	0.000	0.009	0.001	0.000	0.000	0.014	0.001	0.019	0.000	0.000	0.000	0.000	0.000
27 Z	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

using a loss function or by the maximum likelihood consideration. The estimator obtained with a square-loss function turns out to be identical to the a posteriori probability obtained from the maximum likelihood formulation. In particular, with a uniform a priori density function for each n-gram, the formula is

$$p = (m+1)/(n+2)$$

where m is the number of cooccurrences of the ith n-gram in the sample and n is the total number of n-grams. The actual a priori distributions may be more realistically approximated by means of beta distributions with two parameters, yielding a posteriori distributions of the type $(m+a)/(n+b)$.

The estimates of n-gram frequencies obtained by various authors have been compared by Suen in an article that includes also an excellent bibliography on the statistical parameters of textual material.

7.4. *Word distributions and frequencies*

Word distributions depend, of course, both on the language and on the domain of discourse under consideration. Vocabularies in children's books are small, while the average word-length in legal documents is much longer than in newspapers. In technical material the 500 most common words account for 76% of all the words. The distribution of word lengths in a corpus containing ten different types of text is tabulated in a report by Toussaint and Shinghal.

8. Error/reject rates

8.1. *Prediction*

Based on his 1957 work (see above), in 1970 Chow derived some very interesting relations between the substitution error rate and the reject rate of a character recognition system. The principal results are:

(1) The optimum rule for rejection is, regardless of the underlying distributions, to reject the pattern if the maximum of the a posteriori probabilities is less than some threshold.

(2) The optimal reject and error rates are both functions of a parameter t, which is adjusted according to the desired error/reject tradeoff (which, in turn, is based on the cost of errors relative to that of rejections).

(3) The reject rule divides the decision region into 'accept' and 'reject' regions; the error and reject rates are the integrals over the two regions of the probability $P(v)$ of the observations.

(4) Both the error and reject rates are monotonic in t.

(5) t is an upper bound on the error rate.

(6) The slope $\mathrm{d}E/\mathrm{d}R$ of the error–reject curve increases from $-1+1/n$ (n is the number of classes) to 0 as R increases from 0 to 1.

(7) The error/reject rate is always concave upwards ($\mathrm{d}^2E/\mathrm{d}R^2 \geqslant 0$).

(8) The optimum error rate may be computed from the reject rate according to the equation $E(t) = \int_{t=0}^{t} t\,\mathrm{d}R(t)$, regardless of the form of the underlying distribution.

The importance of (8) lies in the fact that in large-scale character-recognition experiments it is generally impossible to obtain a sufficient number of *labelled* samples to estimate small error rates directly. Eq. (8) allows the estimation of the error rate using *unlabelled* samples.

Chow gives examples of the optimum error–reject curve for several forms of the probability density function of the observations.

8.2. *Experimental estimation of the error rate*

Because printed characters do not lend themselves to simple and accurate syntactic or probabilistic description, theoretical predictions of the misclassifica-

tion rate, based on parametric characterizations of the underlying multidimensional probability distributions, are primarily of academic interest. Consequently, whether a given classification scheme meets the specified performance criteria with respect to substitution error and reject rate must be determined by experiment.

In the ideal situation, where an infinite number of character samples is available (and infinite computational resources), there is no theoretical difficulty: simply partition the sample into two (infinite) sets, design the classifier using the method of choice on one set (the *design set*), and estimate the error rate by observing the fraction of misclassified or rejected samples in the other set (the *test set*).

If, however, the number of samples or the available computational resources are finite, then restricting the size of the design set generally leads to an inferior design. On the other hand, using all of the available samples for design leads to a biased estimate of the error rate, because the classifier is tailored to the particular samples used in its design. The problem is, therefore, how to partition the samples in order to obtain both a good design and an acceptable estimate of the error rate. The following discussion is based on an article by Toussaint which contains over one hundred references to original work on error-rate estimation.

Method 1. In early experiments in character recognition, and occasionally in more recent work, the design set and the test set were identical. This leads, as mentioned, to too low an estimate of the error rate. The extent of bias is related to the relative number of 'free parameters' in the classifier and to the size of the sample set.

Method 2. Here the sample is divided into a design set and a test set, with the ratio of the number of samples in the design set to the total number of samples (N) equal to R. $R \times N$ must, of course, be an integer. R is often chosen equal to $\frac{1}{2}$.

Method 3. The sample is partitioned into K sets of size $(1-R)\times N$. Each set from the first partition is then randomly paired with a set from the second partition, and K separate design-and-test experiments are performed. The estimate of the error rate is the average error rate on the K test sets. The variance of this estimate is lower than that of Method 2. Commonly used values of R are $R=\frac{1}{2}$, and $R=(N-1)/N$. The latter is known as the U-method; it provides the largest possible number of independent design-samples for a given total sample size. The maximum possible value of K in the U-method is N.

The lowest expected error rate which can be obtained on an infinitely large, independent test sample using a *finite design sample* is, of course, lower than what could be obtained with an *infinite design sample*, but with a finite sample we can estimate the former. It can be shown that Method 1 yields, on the average, an optimistic estimate, while Methods 2 and 3 yield pessimistic estimates. Given N samples, the U-method yields the lowest underestimate of the expected performance using N characters for design.

Another line of work in character recognition explores the improvement in recognition performance in adaptive recognizers where each new sample contrib-

utes information to tune the classifier. The main branches of this endeavour are *supervised classification* where each new sample is identified, and *unsupervised classification* ('learning without a teacher') where the true identities of the new characters remain unknown. None of this work seems to have found much application to practical OCR systems.

Commercial OCR manufacturers normally test their devices on samples of several hundred thousands or millions of characters in order to make performance estimates at realistic levels of substitution error and reject rate. Academic researchers, on the other hand, usually test their algorithms on a few hundred or thousand samples only, using general purpose computers. The IEEE Pattern Recognition Data Base contains several dozen test sets, originally collected by the IEEE Computer Society's Technical Committee on Pattern Recognition, which are available to experimenters for the cost of distribution. These test databases include script, hand-printed characters, single and multifont typewritten characters, and also some non-character image data. Several articles have been written comparing different studies using identical design and test data. Another source of fairly large files of alphanumeric test data is the Research Division of the U.S. Postal Service.

8.3. Major sources of classification errors

In a complete OCR system, the vast majority of the errors can usually be attributed to the preprocessor or front-end. This includes, as we have seen, the optical transducer itself and the character-acquisition stages. *Photometric non-linearities* in the optical scanner and lack of contrast in the material lead to troublesome and inconsistent black–white inversions. *Geometric non-linearities* in the scanner and misadjusted printing mechanisms (such as bent type bars) result in distortions in the shape of the characters and in misaligned and skewed patterns. These problems can be readily detected by visual inspection of the digitized pattern arrays, hence most systems include provisions for monitoring the pattern 'video'.

Even with high quality printing, *segmentation errors* often result in mutilated or incomplete characters presented to the classifier. This is due to the fact that while a four-mil scanning resolution is considered acceptable for the classification of most standard-size fonts, the amount of blank space between adjacent characters is frequently much less than the scanner spot size, and the digitized arrays corresponding to adjacent characters are therefore 'bridged'. Decreasing the spot-size not only increases the amount of data to be processed but also, with many serif fonts, a significant fraction of adjacent character-pairs is not separated by *any* detectable white space. Character classification methods that do not depend on accurate segmentation show a definite advantage on such material.

Even if the characters are correctly segmented, the presence of extraneous blots may affect registration. Blots which are cleanly separated from the character are not difficult to remove, but contiguous blots pose a severe problem. Since in practical devices recognition is usually attempted in only a few shifted positions, a

two or three pixel misregistration may lead to additional rejects.

In the classification stage, errors on clean, well-segmented, and registered characters are committed either because the structure and variability of the material do not correspond to the model used to design the classifier (inappropriate choice of classification method), or because the data used to adjust the parameters of the classifier (the training set) is not sufficiently representative of the data used for testing or operational application. Except for the classification of hand-printed characters, however, any effort invested in improving the quality of the output of the digitizing and preprocessing stages tends to be more cost-effective than fine-tuning the classifier itself.

8.4. Recognition rates in diverse applications

Because print quality is difficult to characterize accurately, few commercial manufacturers are willing to be pinned down to specific performance rates. Furthermore, once a system is installed the user has almost no way to find out its operational substitution error rate: it is simply too costly to record the correct identities of several hundreds of thousands of characters and to match them with the string of identities assigned by the OCR system. The ranges of substitution error and reject rates cited in this section should therefore be considered with extreme caution (Table 3).

For *handprinted characters* the most important single factor affecting recognition performance is the motivation of the writers. The performance rate on an untrained but motivated population is of the order of 99% correct recognition for numerals only and 95%–98% for alphanumeric applications. The error rate is two or three times lower with a captive population, such as clerks in retail stores, where it is possible to provide feedback to correct misrecognized stylistic idiosyncrasies of the writers.

The performance rate on *specially-designed stylized* characters may reach 1/200000 substitution error rate and 1/20000 reject rate. Almost comparable performance may be obtained with certain typewriter fonts, such as *modified Courier*, when typed on specially adjusted typewriters using carbon film ribbon and high quality paper. On *ordinary typescript*, using standard typefaces, error and reject rates may be one to two order of magnitude higher, depending on the typeface and the print quality. Restricting the alphabet to one case or to numerals only results in an improvement roughly proportional to the decrease in the number of classification categories.

Table 3
'Typical' error and reject rates

	Error rate	Reject rate
Stylized characters	0.000005	0.00005
Typescript (OCR quality)	0.00001	0.0001
Ordinary typescript	0.0005	0.005
Handprint (alphanumeric)	0.005	0.05
Bookface	0.001	0.01

There are twenty-four large capacity *mail sorters* installed in the U.S. Post Offices in large cities throughout the country. The older machines are effective only in sorting out-going mail, where the contextual relations between the city, state, and zip-code on the last line of the address allows correct determination of the destination even with several misrecognized characters. On high quality printed matter (most of the mail in the United States, as opposed to Japan, has a printed or typewritten address) 70% to 90% of the mail pieces are routed correctly, and the rest is rejected for subsequent manual processing. In order to use the machines effectively, obviously unreadable material is not submitted to the machines. Large mailers participate in the U.S. Postal Service's 'red tag' program by affixing special markers to large batches of presorted mail which is known to have the appropriate format for automatic mail sorting.

The recognition performance on *variable-pitch typeset material* is much lower than on typewritten and special OCR fonts—up to 1% of the characters may not be recognized correctly. Only a few manufacturers market machines for typeset material. Some of these machines must be 'trained' on each new font. Mixed-font OCR has been successfully applied to reading-aids for the blind. Here even if a relatively high fraction of the characters is misrecognized, the human intelligence can make sense from the output of the device. In one commercially available system the reader is coupled to an automatic *speech synthesizer* which voices the output of the machine at an adjustable rate between 100 and 300 words per minute. When the rules governing its pronunciation fail, this machine can spell the material one letter at a time.

Few major benefits can be expected from further improvement of current commercial classification accuracy on clean stylized typescript and on stylized fonts. Current developments are focussed, therefore, on enabling OCR devices to handle increasingly complicated formats (such as technical magazine articles) in an increasing variety of styles including hand-printed alphanumeric information, ordinary typescript, and bookface fonts, all at a price allowing small decentralized applications. When these problems are successfully solved, we may expect general purpose OCR input devices to be routinely attached to even the smallest computer systems, complementing the standard keyboard.

Acknowledgment

The author wishes to acknowledge the influence on the points of view expressed in this article of a number of former colleagues, particularly R. Bakis, R. G. Casey, C. K. Chow, C. N. Liu, and Glenn Shelton, Jr. He is also indebted to R. M. Ray III for some specific suggestions.

Bibliography

[1] Abend, K. (1968). Compound decision procedures for unknown distributions and for dependent states of nature. In: L. N. Kanal, ed., *Pattern Recognition*. Thompson, Washington/L. N. K. Corporations College Park, MD.

[2] Ascher, R. N. (et al.), (1971). An interactive system for reading unformatted printed text. *IEEE Trans. Comput.* **20**, 1527–1543.
[3] Bird, R. B. (1967). Scientific Japanese: Kanji distribution list for elementary physics. Rept. No. 33. Chemical Engineering Department, The University of Wisconsin.
[4] Bledsoe, W. W. and Browning, I. (1959). Pattern recognition and reading by machines. *Proc. E.J.C.C.*, 225–233.
[5] British Computer Society. (1967). *Character Recognition*. BCS, London.
[6] Casey, R. G. and Nagy, G. (1968). An autonomous reading machine. *IEEE Trans. Comput.* **17**, 492–503.
[7] Casey, R. G. and Nagy, G. (1966). Recognition of printed Chinese characters. *IEEE Trans. Comput.* **15**, 91–101.
[8] Chen, H. C. (1939). *Modern Chinese Vocabulary*. Soc. for the Advancement of Chinese Education. Commercial Press, Shanghai.
[9] Chow, C. K. (1957). An optimum character recognition system using decision functions. *IRE EC* **6**, 247–257.
[10] Chow, C. K. and Liu, C. N. (1966). An approach to structure adaptation in pattern recognition. *IEEE SSC* **2**, 73–80.
[11] Chow, C. K. and Liu, C. N. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inform. Theory* **14**, 462–467.
[12] Chow, C. K. (1970). On optimum recognition error and reject tradeoff. *IEEE Trans. Inform. Theory* **16**, 41–46.
[13] Deutsch, S. (1957). A note on some statistics concerning typewritten or printed material. *IRE IT* **3**, 147–148.
[14] Doyle, W. (1960). Recognition of sloppy, hand-printed characters. *Proc. W.J.C.C.*, 133–142.
[15] Fischer, G. L. (1960). *Optical Character Recognition*. Spartan Books, Washington.
[16] Genchi, H., Mori, K. I., Watanabe S. and Katsuragi, S. (1968). Recognition of handwritten numeral characters for automatic letter sorting. *Proc. IEEE* **56**, 1292–1301.
[17] Greanias, E. C., Meagher, P. F., Norman, R. J., and Essinger, P. (1963). The recognition of handwritten numerals by contour analysis. *IBM J. Res. Develop.* **7**, 14–22.
[18] Greenough, M. L. and McCabe, R. M. (1975). Preparation of reference data sets for character recognition research. Tech. Rept. to U.S. Postal Service. Office of Postal Tech. Res., Pattern Recog. and Comm. Branch, Nat. Bur. of Standards, NBS1R75-746. Washington.
[19] Harmon, L. D. (1972). Automatic recognition of print and script. *Proc. IEEE* **60**, 1165–1176.
[20] Hennis, R. B. (1968). The IBM 1975 optical page reader. *IBM J. Res. Develop.* **12**, 345–371.
[21] Herbst, N. M., and Liu, C. N. (1977). Automatic signature verification. *IBM J. Res. Develop.* **21**, 245–253.
[22] Hoffman, R. L. and McCullough, J. W. (1971). Segmentation methods for recognition of machine-printed characters. *IBM J. Res. Develop.* **15**, 101–184.
[23] Hussain, A. B. S., and Donaldson, R. W. (1974). Suboptimal sequential decision schemes with on-line feature ordering. *IEEE Trans. Comput.* **23**, 582–590.
[24] Kamentsky, L. A., and Liu, C. N. (1963). Computer automated design of multi-font print recognition logic. *IBM J. Res. Develop.* **7**, 2–13.
[25] Kanal, L. N. (1980). Decision tree design—the current practices and problems. *Pattern Recognition in Practice*. North-Holland, Amsterdam.
[26] Kovalevsky, V. A. (1968). *Character Readers and Pattern Recognition*. Spartan Books, Washington.
[27] Liu, C. N. and Shelton, G. L., Jr. (1966). An experimental investigation of a mixed-font print recognition system. *IEEE Trans. Comput.* **15**, 916–925.
[28] Minsky, M. (1961). Steps towards artificial intelligence. *Proc. IRE.*
[29] Nadler, M. (1963). An analog-digital character recognition system. *IEEE Trans. Comput.* **12**.
[30] Neuhoff, D. L. (1975). The Viterbi algorithm as an aid in text recognition. *IEEE Trans. Inform. Theory* **21**, 222–226.
[31] OCR Users Association (1977). *OCR Users Association News*. Hackensack, NJ.
[32] Ledley, G. (1970). Special issue on character recognition. *Pattern Recognition* **2**. Pergamon Press, New York.

[33] Raviv, J. (1967). Decision making in Markov chains applied to the problem of pattern recognition. *IEEE Trans. Inform. Theory* **13**, 536–551.

[34] Riseman, E. M. and Ehrich, R. W. (1971). Contextual word recognition using binary digrams. *IEEE Trans. Comput.* **20**, 397–403.

[35] Riseman, E. M. and Hanson, A. R. (1974). A contextual postprocessing system for error correction using binary N-grams. *IEEE Trans. Comput.* **23**, 480–493.

[36] Schantz, H. F. (1979). A.N.S.I. OCR Standards Activities. *OCR Today* **3**. OCR Users Association, Hackensack, NJ.

[37] Shannon, C. (1951). Prediction and entropy of printed English. *BSTJ* **30**, 50–64.

[38] Shillman, R. (1974). A bibliography in character recognition: techniques for describing characters. *Visible Language* **7**, 151–166.

[39] Shinghal, R., Rosenberg, D., and Toussaint, G. T. (1977). A simplified heuristic version of recursive Bayes algorithm for using context in text recognition. *IEEE Trans. Systems Man Cybernet.* **8**, 412–414.

[40] Shinghal, R., Rosenberg, D., and Toussaint, G. T. (1977). A simplified heuristic version of Raviv's algorithm for using context in text recognition. *Proc. 5th Internat. Joint Conference Artificial Intelligence*, 179–180.

[41] Stevens, M. E. (1961). Automatic character recognition-state-of-the-art report. Nat. Bureau Standards, Tech. Note 112. Washington.

[42] Suen, C. Y. (1979). Recent advances in computer vision and computer aids for the visually-handicapped. *Computers and Ophthalmology. IEEE* Cat. No. 79CH1517-2C.

[43] Suen, C. Y. (1979). N-gram statistics for natural language understanding and text processing. *IEEE PAMI* **1**, 164–172.

[44] Suen, C. Y. (1979). A study on man-machine interaction problems in character recognition. *IEEE Trans. Systems Man Cybernet.* **9**, 732–737.

[45] Thomas, F. J. and Horwitz, L. P. (1964). Character recognition bibliography and classification. IBM Research Report RC-1088.

[46] Toussaint, G. T. (1974). Bibliography on estimation of misclassification. *IEEE Trans. Inform. Theory* **20**, 472–479.

[47] Toussaint, G. T. and Shinghal, R. (1978). Tables of probabilities of occurrence of characters, character pairs, and character triplets in English text. McGill University, School of Computing Sciences. Tech. Rept. No. SOCS 78-6, Montreal.

[48] Toussaint, G. T. (1978). The use of context in pattern recognition. *Pattern Recognition* **10**, 189–204.

[49] Walter, T. (1971). Type design classification. *Visible Language* **5**, 59–66.

[50] Wright, G. G. N. (1952). *The Writing of Arabic Numerals*. University of London Press, London.