Adaptive and Interactive Approaches to Document Analysis

George Nagy[1] and Sriharsha Veeramachaneni[2]

[1] RPI ECSE DocLab, Troy, NY 12180 USA, nagy@ecse.rpi.edu
[2] IRST Trento, Italy, sriharsha@itc.it

Summary. This chapter explores three aspects of learning in document analysis: (1) field classification, (2) interactive recognition, and (3) portable and networked applications. Context in document classification conventionally refers to language context, i.e., deterministic or statistical constraints on the sequence of letters in syllables or words, and on the sequence of words in phrases or sentences. We show how to exploit other types of statistical dependence, specifically the dependence between the shape features of several patterns due to the common source of the patterns within a field or a document. This type of dependence leads to field classification, where the features of some patterns may reveal useful information about the features of other patterns from the same source but not necessarily from the same class. We explore the relationship between field classification and the older concepts of unsupervised learning and adaptation. Human interaction is often more effective interspersed with algorithmic processes than only before or after the automated parts of the process. We develop a taxonomy for interaction during training and testing, and show how either human-initiated and machine-initiated interaction can lead to human and machine learning. In a section on new technologies, we discuss how new cameras and displays, web-wide access, interoperability, and essentially unlimited storage provide fertile new approaches to document analysis.

1 Introduction

The classical models for character recognition and document image analysis must be extended to accommodate the classification of multiple common-source patterns. We show how field classification exploits statistical dependence due to the common source of a field of patterns and also leads to a simple and operational definition of classifier adaptation. We explore diverse contextual constraints beyond those imposed by language models. Instead of ignoring the ever-present human-computer interaction, we propose more effective ways of exploiting it. We also examine the impact of recent technological developments on OCR and DIA and raise some research questions.

G. Nagy and S. Veeramachaneni: *Adaptive and Interactive Approaches to Document Analysis*, Studies in Computational Intelligence (SCI) **90**, 221–257 (2008)
www.springerlink.com

In this introductory section we list some of the limitations of conventional models for classification and propose extensions to field classification, adaptation and unsupervised learning. In the second section, *Field Classification*, we examine the contextual constraints that favor field classification and present some situations for which field classification algorithms have already been developed. We also attempt to clarify some distinctions between *trainable, supervised, semi-supervised, unsupervised, adaptive and self-organizing* algorithms for *training, teaching*, and *learning*.

The third section, *Interaction in training and testing*, considers paradigms where some interaction helps either or both the operator or the algorithmic parts of the system. Our premise is that interaction will remain necessary, for the foreseeable future, in most operational recognition systems. The fourth section, *Technology and Applications*, explores how advances in technology foster new applications in OCR and DIA. While the rest of this survey is mainly a retrospective and attempts to rationalize existing results, here we attempt to look ahead. In the *Conclusions* we list some trends and open research problems.

1.1 The Classical Paradigm for Pattern Recognition

Until the last decade, the customary framework for statistical pattern recognition in Optical Character Recognition (OCR), Hand-printed Character Recognition (HCR), and Document Image Analysis (DIA) was based on three key constraints:

1. *Representative training set.* The data was divided into two mutually exclusive sets of patterns for training and testing, each consisting of samples from a fixed number of classes with given or estimated prior probabilities. It was generally assumed that the patterns in both sets were independent samples produced by selecting a class label according to the prior class probabilities, then generating an observation (feature, attribute) vector from the corresponding class-conditional probability distributions. The labels of the test set were used only to determine classification accuracy. (Some researchers partitioned the training set further to provide a validation set for tuning parameters.)
2. *Singlet classification.* The patterns were classified one at a time. Each pattern was assigned a label on its own merits, independently of every other pattern. In OCR and HCR, each pattern was a single glyph (i.e., a letter, numeral, or ideograph). In DIA, it could be an entire word, a drawing or a photograph, or even an entire document. The only important exception to this constraint was the application of *linguistic context* in character recognition. Other entities, like forms, tables and documents, were also usually processed as though they occurred in isolation.
3. *No interaction.* Only algorithmic processes were considered of interest. It was understood that in real applications the labels in both the training

> set and the test set would have to be provided by key entry, that human help was necessary to produce segmented character or word patterns for experimentation, and that intervention would be necessary at the operational level to deal with unclassified and misclassified patterns. However, these interactive components of the system were considered extraneous to the pattern recognition system, and in research settings and research publications little attention was devoted to optimizing them.

This architecture is typically represented by a data flow diagram similar to that shown in Fig. 1. No special provisions are made to indicate either the relevant data sets or the class labels.

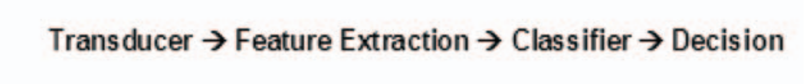

Fig. 1. Generic first generation pattern recognition system

Over the last decade or two, many systems were proposed that did not fit neatly into the above paradigm. Some of the new approaches were the result of theoretical advances, while others arose from the realization that some important applications grossly violated the stated constraints. Many simply exploited technological advances: faster CPUs, larger amounts of storage, miniaturization, portability and connectivity, and better displays.

Our objective in this chapter is to construct a more general framework for pattern classification that encompasses recent research and may even leave room for new ideas. The notions at the core of the new paradigm are *learning and adaptation*, *style*s, *multi-pattern classification*, and *human-machine interaction*. We will give examples of methods and applications that fit the new paradigm, and discuss the technological advances that made them possible. We will also show how some widely used techniques, like clustering, expectation maximization, and active learning, fit naturally into the proposed framework.

We propose to define the new paradigm at a level of detail sufficient for probabilistic simulation of alternative classification algorithms. In this kind of simulation, the labels and patterns are generated by pseudo-random-number generators such as are readily found in most programming language libraries and in Matlab or Excel. Languages and software packages designed for simulation, like Simula, Modsim, Ross, Simscript, and Matlab toolboxes, offer a variety of built-in univariate probability distributions. It is, however, more difficult to generate multivariate distributions (e.g., Multinomial, Dirichlet, or Uniform), other than Gaussian, with the desired degree of statistical dependence completely specified by an arbitrary covariance matrix.

Our architecture for simulation does not address a critical component of all pattern recognition systems, *feature extraction.* Feature extraction is the step that transforms the output of the transducer (scanner, camera, tablet, microphone) into an abstract high-dimensional vector space where classification boundaries are defined. The error rate achievable by an ideal classifier depends on the chosen set of features. We are not, however, aware of any general technique for designing a good feature set, and even methods for selecting a subset of good features from a larger set leave much to be desired. We will therefore blithely assume that for each application some expert has already provided software for generating feature vectors. The simulations will start with a probabilistic feature space where the simulation parameters can simply be set to “good” features or “bad” features.

Another important aspect of OCR and DIA that we cannot simulate (but will discuss) is *segmentation.* Much effort has been devoted to separating text from illustrations, locating paragraphs and lines of print, and to word and character segmentation. Although the relevant algorithms fall in the realm of image processing rather than pattern recognition, segmentation and classification are often combined. As for feature extraction, there are few statistical models and tools for segmentation.

In the next subsection we define the components of a more comprehensive classifier architecture.

1.2 Definitions for an Expanded Paradigm

The definitions here pertain primarily to the role of various data sets in a classification system, with particular regard to simulated data. Merely envisaging it lends precision to definitions.

Training set. The training set consists of a set of labeled pattern (feature) vectors. For the purpose of analysis, one can assume that the feature vectors have either continuous or discrete valued components, but simulators can generate only discrete valued features. The number of components in the feature vectors, called the *dimensionality* of the problem, is fixed. There are four types of labels: *class labels, source labels, style labels,* and *instance labels*, as described below. The training set must have at least class labels, but the presence or absence of source and style labels leads to different types of classifiers. Patterns with the same source label share the same style, while patterns with different source labels may or may not be of the same style.

Test set. The test set consists of a sequence of feature vectors with source and class labels. The class labels must be used only for error counts. Each test pattern has a source label. The *source length* is the number of test patterns from the same source. The source distributions may be the same as in the training set, or different. Even if they are the same, the correspondence between the source labels of the test set and the source labels of the training set is assumed to be unknown. The test set has no style labels.

Field. For purpose of classification, the patterns of the test set are divided into fields. A field consists of a fixed number of patterns (called *field length*) from the same source. The choice of field length depends on the available computing resources, while the source length (the number of patterns from the same source in the test set) defines the scope of statistical dependence or context. Even in the absence of linguistic context, a field length of only two (i.e., *pair classification*) may lead to a significant increase in accuracy over singlet classification.

Source, style, and class labels. The assignment of these labels is the most time-consuming part of preparing a real dataset for experimentation. Under the assumption that each document is generated by the same source, only one *source label* per document need be entered. *Style labels* in the training set may be assigned by inspection, by font recognition for printed characters, by clustering or expectation maximization for handprint, or not assigned at all. Initial *class labels* are usually assigned by some classifier, and then the errors are found by proofreading and corrected manually. Sometimes data for experiments on printed characters is automatically generated by a script that generates so many samples of each font, in which case source, style and class labels can be assigned automatically.

Instance labels. Although not necessary for describing a classification scheme, it is good experimental practice to attach a unique label (*accession code, serial number*, *identifier*) to every pattern. This allows tracking changes in class label assignments when classifier parameters are changed, and whether errors committed by different classifiers are correlated. It may also serve as a *time stamp* for scenarios where the order of the patterns within the field matters, as in the case of linguistic context.

Example: Some NIST data sets have samples of isolated digits (10 classes). Each pattern is represented by a 24x30 binary array, therefore the dimensionality of the feature vector is 720 [1]. There are 600 writers, and the serial number of the writer is attached to each digit. These writer labels are our source labels. Writer consistency in the shape of the numerals is one aspect of style. Several writers may have the same style. To ensure that patterns of the same writer do not occur on both training and test set, the data is partitioned *by writer.* There are about 100 digits from each writer, so the source length is approximately 100. The NIST data set does not include style labels. As we will see, the presence of styles may improve classification accuracy even without the presence of explicit style labels.

Simulated data. The source label, which identifies patterns guaranteed to have the same style, is generated first. Then a style label is selected with fixed prior probability over the styles. Next, a sequence of class labels is generated according to *class priors.* The source length may be fixed or subject to a probability distribution that governs the number of patterns per source (for example, the number of digits in the courtesy field of a bank check, or the number of letters, digits and punctuation in a business letter). Finally, the feature vectors are generated from *class-and-style conditional feature distributions.* The patterns from each source are restricted to a single style; in other words, isogenous or common-source patterns of the same class are independently drawn samples from the same distribution. Before classification, the test set is partitioned into same-source fields. To simulate some applications, we may allow all of these probability distributions to change gradually. This makes a difference only if the classifier has a *bounded horizon*, i.e., if the field is shorter than the test set.

The important distinction between the new framework and old framework is the presence of multiple feature distributions that are constant within a source but may change from source to source. In order to exploit within-source consistency, *field classification* rather than singlet classification is necessary. A further distinction is that the distribution of the patterns may *change with time*. The nature of the classifiers appropriate for different scenarios within the above overall framework is elaborated in the next section.

2 Field Classification

We are now ready to consider situations where it is advantageous to classify entire groups of objects instead of classifying each object in isolation. This is generally the case in DIA and OCR, where a message (substantiated as a document) consists of an ordered collection of visual objects (*glyphs*). We show that many common constraints on acceptable sequences of symbols, and on the visual appearance of the glyphs used to represent them, can be expressed in terms of statistical dependence between patterns. Because the estimation and exploitation of the underlying joint probability distributions requires examination of more than one pattern at a time, we discuss *field classification*. We relegate the relevant mathematical formalisms to the cited references, but we present some tools that facilitate the study of the inter-pattern feature dependences, and state the assumptions under which optimal or approximate field classification algorithms have already been developed. We conclude the section with a discussion of adaptive classification and unsupervised learning.

2.1 Context

Information relevant to classifying an object (digit, letter, word, illustration or document) is often extraneous to the object itself. It may either reside in other objects that are also to be classified, or it can be considered part of the environment in which the classifier operates. In the first case, recognition accuracy can be improved by taking into account the characteristics of an entire group of objects to classify each one, i.e., by field classification. In the second case, the recognition can be improved by providing means to specialize or tune the classifier for either singlet patterns or fields to its environment. The additional information is generally called *context*, regardless of whether it can be derived from the available samples [2, 3].

In character and speech recognition, the word “context” is often reserved for linguistic context. It has, however, a much wider scope in Artificial Intelligence, as exemplified by the topics discussed at the biennial ACM Context conferences, which draw on several centuries of studies in epistemology [4, 5, 6, 7].

We will examine situations other than linguistic context where field classification is useful, but neglect broader considerations that need to be taken into account in preprocessing and feature extraction rather than in the classifier itself. In other words, we will concentrate on the kinds of context where the patterns to be classified provide information about each other.

Since the use of linguistic context is well established in both character and speech recognition, we will first look at *language models*. Then we will examine some relations between the *shapes* of the patterns. We will distinguish between *order-independent* and *order-dependent* relations, and also between forms of statistical dependence that arise *between labels*, *between shape features*, and *between labels and shape features* –of all the patterns within a field.

Language Models

Language models are approximate descriptions of natural language at the morphological, lexical, syntactic, semantic, or pragmatic levels. While many of the earlier models were rule-based, the advent of large computer-readable corpora for estimating parameters has given rise to statistical models.

Morphological models typically consist of polygram frequencies [8]. These frequencies vary from language to language and are always highly skewed [9]. In English text, for example, the probability of “e” is 0.1241, while that of “z” is only 0.0007. The skew increases with polygram length: P[th]=0.04, while P[qh]=0. It is clear that an ambiguity between an `e` and a `c` after a `d` should be resolved in favor of `e`, but is `e` or `c` more likely after `u`? Elaborate methods have been proposed to estimate the probabilities of rare letter or phoneme sequences [10].

Lexical models are based on word frequencies and word transition frequencies. The simplest systems are based on dictionaries (strictly speaking, *lexicons*) that report only the existence or non-existence of a sequence of letters as a valid word of a particular language, without its frequency of occurrence. (*Agglutinative* languages with many case endings and verb forms, like Italian, typically require lexicons at least three times larger than English.) Commercial OCR systems routinely use not only large general lexicons but also specialized lexicons of biological, chemical or legal terminology, and lists of abbreviations, acronyms, and proper names. Most often dictionary-lookup is carried out only as a post-processing step, which is generally suboptimal. Some examples of over-reliance on lexicons are given in [11], which also describes many other sources of OCR errors.

The best statistical *syntactic models* surpass the power of rule-based systems [12, 13, 14]. Estimates of transition frequencies between syntactic categories (noun, verb, adjective, adverb ...) can be obtained from large annotated corpora. Syntactic models are of limited use in English because of the multiple categories carried by many words (e.g., *Yellow* soap / I wonder where the *yellow* went, or To *fit* a dress / A *fit* athlete / A good *fit*). Applying semantic and pragmatic models is even harder [15].

Formally, all linguistic context in character recognition can be expressed as statistical dependence between the *labels* of patterns. The random variables whose joint probabilities must be estimated are letter, word, or part-of-speech labels. Linguistic context is always *order-dependent*, and therefore often modeled with transition frequencies in Markov Chains, Hidden Markov Models, and Markov Random Fields [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. Linguistic variables are usually assumed to be independent of character shape, even though titles and headings in large or bold type have a different language structure than plain text. Optimal field classification of printed matter is often approximated by a post-processor that simply attempts to integrate confidence measures based on shape and language.

Style

We term *style* any difference between the statistical characteristics of a group of patterns generated by a single source and the characteristics of a group of patterns generated by several sources [29, 30, 31]. A single-source group usually exhibits some shape consistency. For instance, we may be able to distinguish numerals written by Alice from numerals written by Bob. Alice's numerals seem similar to each other, and Bob's numerals are also similar to each other, but Alice's numerals are different from Bob's. The same notion can be applied also to text printed in different fonts. Forensic analysts can tell whether two sets of letters or numerals were written with the same pen, or printed on the same printer.

More formally, style context is defined as the presence of statistical dependence arising between patterns (represented as random vectors) because they are from the same source. Unlike language context, it is independent of the order of the patterns in the field. It takes two distinct forms, which we call intra-class style and inter-class style [32].

Intra-class style is the shape consistency of a single class from each source. It reveals how consistent a writer is in writing a glyph. Does Alice always cross her 7s, while Bob never does? It is, of course, even more marked in print, where words, paragraphs, and entire documents are often composed in a single typeface. Experts can recognize dozens of typefaces by inspection. More subtle than typeface consistency is the intra-class style within documents printed by the same printer or scanned by the same scanner. In OCR, where each glyph (a letter, numeral or ideograph) is usually represented by a feature vector, we say that a data set exhibits intra-class style if the feature-vectors of patterns of the *same* source and class, considered as random variables, are (class-and-style-conditionally) statistically dependent.

Inter-class style determines how much the shape of a given class reveals about the appearance of *other* classes from the same source. The way Alice writes `1` helps predict the way she will write `7`. If the `n` has no serifs, neither will `h`, `m`, or `r`. We say that the data set exhibits inter-class style if the feature-vectors of patterns of *different classes*, considered as random variables, are (class-and-style-conditionally) statistically dependent. Fig. 2. illustrates two pairs that cannot both be recognized correctly as 17 by a singlet classifier, and an instance where the label assigned by a singlet classifier is corrected by the field classifier.

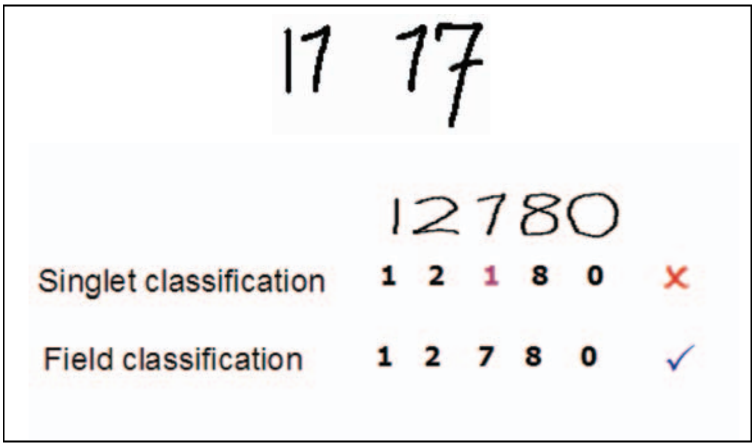

Fig. 2. Benefit of pair classification when there is inter-class style

We show in Fig. 3 a useful representation for visually comparing singlet and field classification boundaries. (This representation allows showing only a single feature for each pattern, hence only a 2-D *field feature.*) We plot some field features, which have bimodal Gaussian mixture distributions, for each field class (AA, AB, BA, BB) of a two-class problem (A,B) with a single feature x. We also show the 2-D decision boundaries of the singlet classifier and of the field classifier. The optimal field classification boundaries and the singlet boundaries are different, so we would expect some gain with a field classifier. Simulation of a field classifier shows a reduction in the error rate of single patterns from 15% to about 10%.

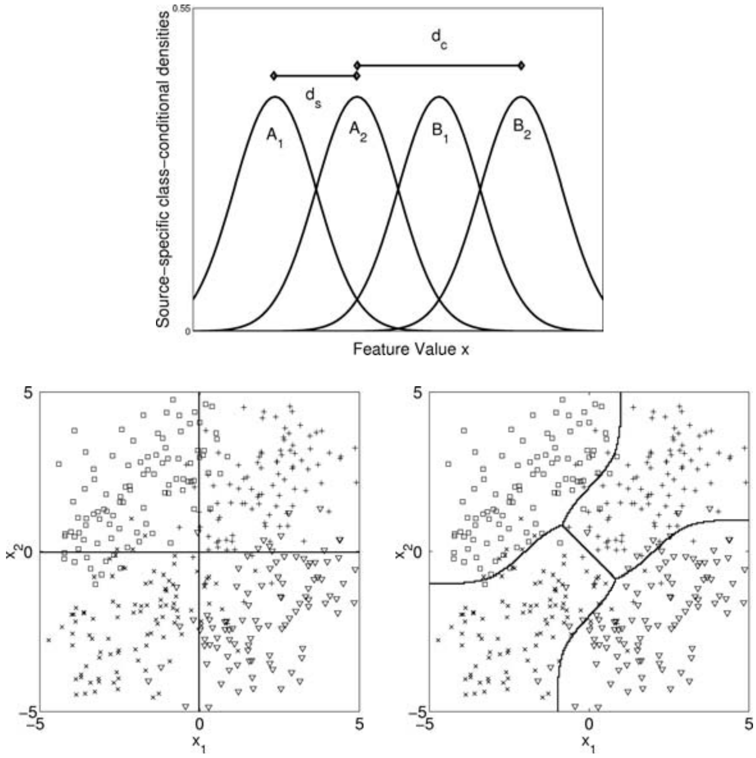

Fig. 3. Gaussian class-conditional distributions of a single feature x for classes A and B and styles 1 and 2. Below are the representations of singlet (left) and pair (right) feature (spaces x_1, x_2) and decision boundaries for a field of two patterns. The AA region is bottom left, AB is bottom right, BA top left, and BB top right

Order-Dependent Inter-Pattern Dependence

Inter-pattern class-feature dependence is fairly rare. It occurs when features of a pattern depend on the *class*, rather than on the rendering (features), of an adjacent pattern. For example, the vertical location of an apostrophe may depend on whether the previous letter had an ascender, but not on its font (Fig. 4). That is, the features of the apostrophe are independent of the preceding letter, *given its label.*

Feature dependence between adjacent patterns is common, as illustrated in Fig. 5 (from [3]). In cursive writing, the location of the last stroke of a pattern determines the nature of the ligature that joins it to the next pattern. It is different from style, because the ligature-sensitive features of the two patterns depend on the shape and *order* of the patterns. A similar phenomenon in speech is called co-articulation.

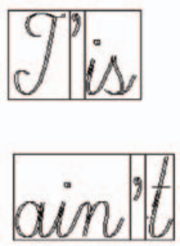

Fig. 4. Example of inter-pattern class-feature dependence

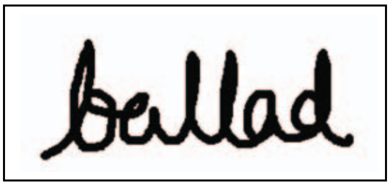

Fig. 5. Examples of order-dependent inter-pattern feature dependence: note the difference between the ligatures preceding the a's

2.2 Field Classifiers

Dependence between patterns suggests that a field classifier should assign a *field label*, consisting of a sequence of class labels, based on the feature vectors of *all the patterns* in a test field. The number of possible field labels rises exponentially with field length, thereby effectively limiting the maximum operational field length.

We discuss below several types of field classifiers that have been proposed under various assumptions. These field classifiers are generally based on the formulation of singlet classifiers: for instance, they may be Bayes classifiers or MAP classifiers, and either parametric or non-parametric classifiers. In addition to standard statistical classifiers, neural networks and support vector machines can also be exploited for field classification. To classify each pattern, all field classifiers combine information derived from the entire training set with information from the whole test field.

Field-Trained Classifiers

An obvious idea is to concatenate the features of singlet patterns to form field feature vectors, and train the classifier on every possible field class. All of the well-developed theory of singlet classification then applies. This method, however, requires training samples of every field class, and is therefore generally impractical with field lengths greater than two.

In text, not all combinations of letters occur. Word classifiers can therefore be trained on words, rather than on every possible sequence of characters. One version of this approach divides the letters of the alphabet into fewer and more easily classified categories based on character shape codes (ascenders and descenders) [33, 34]. Because character-level segmentation is error prone, most word classifiers are not based on concatenating singlet character features, but on features extracted from the entire word. This approach is particularly suitable for limited-vocabulary applications like postal addresses or legal amounts on bank checks [35, 36], and for correction of OCR errors [37]. Another example of holistic word classification is based on statistics extracted from each cell of a grid superimposed on the word [38]. For degraded documents with larger vocabularies, word level indexing (as opposed to keyword spotting) was proposed with a three-stage comparison based on word aspect ratios, vector features extracted with a grid superimposed on each word, and within-word connectivity. Experimental evidence for high precision and recall in retrieval was adduced from a multilingual collection of OCR-resistant documents spanning four centuries [39].

Font Classification

For printed matter, *font classifiers* and *font-specific character classifiers* can be trained on data sets of specific type faces or on broad groups (serif/sans-serif, italic, bold). The font classifier is then applied first to a test field, and its decision is used to select the appropriate character classifier [40, 41, 42, 43]. The same idea can be applied to writer identification [44]. Many words of text may be necessary to reliably identify the font. Furthermore, the resulting classifier is generally suboptimal, because the features in character classification are neglected in font classification, and those used in the font classifier are neglected in character classification. Style classifiers, discussed below, use the entire set of features.

Discrete-Style Classifier and Style-First Classifier

If the underlying feature distributions are Gaussian, and the training set has style labels that allow estimating the parameters of the class-and-style conditional feature distributions, then the joint posterior mixture-distributions of the field classes can be computed for fields of arbitrary length. The resulting optimal classifier is known as the *Discrete Style Classifier* [31]. The lengthy computation (exponential with field length) can be approximated by keeping track of frequently co-occurring (same-source) shapes [45] or, more consistently, by a *Style First Classifier* that computes the posterior probability based on the most likely style [46]. Non-parametric nearest-neighbor field classifiers are described in [47] and support vector machine field classifiers in [48].

Style-Conscious Quadratic Discriminant Field Classifier

In some applications, like handprint recognition with a multitude of writers, it is sensible to assume a continuous distribution of Gaussian styles instead of some predetermined fixed number. The posterior distribution for any field length can then be determined from only the cross-covariance matrix of *pairs* of same-source pattern feature vectors (*op. cit.* [46]). The resulting Style-Conscious Quadratic Discriminant Field classifier is optimal under the stated assumptions. The experiments described in the cited references indicate that all of these field classifiers achieve lower character error rate than the singlet classifiers on which they are based.

2.3 Adaptive Classifiers

The word *adaptive* (which surfaced in conjunction with stochastic approximation, potential functions, adelines, madelines, and perceptrons), is overloaded and has been used in many different ways since its appearance – first in automatic control then in pattern recognition – more than forty years ago [49]. Adaptation and learning were linked to stochastic approximation (Robbins-Monroe and Kiefer-Wolfovitz processes) by Aizerman, Tsypkin and Fu among others [50, 51, 52]. Nevertheless we need a word for a concept that fits with our definitions of training and test sets and of field classification, and that shares the connotation associated with adaptation. In our context, adaptation can be defined clearly and simply without introducing any additional notions. Our definition offers the advantage that it applies equally to structural adaptation, parameter adaptation, and to complex classification formulas that could be equivalent to either.

We define an *adaptive classifier* as a *field classifier with a field that encompasses the entire test set.*

Such a classifier can clearly use all of the information that is available in the patterns to be classified. Not only does the classification of the last pattern in the test set profit from information garnered from the first pattern, but the classification of the first pattern also benefits from the last pattern. In principle, the field-classification boundaries of an adaptive classifier can be determined entirely from the training set. This does not imply that the distribution of shorter subsequences of patterns in the field is stationary, but it does require all of the test patterns to be available at the same time.

For long test sets, the computation of the posterior field probabilities must be approximated. *Dynamic field classifiers* adjust their classification parameters after classifying a finite subset of the test field, thereby approximating an optimal adaptive classifier. The approximation may be necessary either because there are insufficient computational resources to classify the entire test set optimally, or because some of the test patterns must be classified before all of them are available.

Dynamic classifiers present the danger of wandering off course, perhaps because of a completely mislabeled subset of the test field, and never recover. It may therefore be prudent to test them periodically on some typical validation field and, if the error rate is too high, reset the parameters to those obtained from the original trusted training set. Adaptation in commercial OCR systems seldom exceeds page length (c. 2000 characters).

The style-constrained classifiers described below are not dynamic, because their decision depends only on the ensemble of patterns of the current field, which is generally only a subset of the test set. If a field reappears later, after many other fields from different sources and styles were classified, the result will be the same. (In contrast, a classifier that is dynamic according to our definition could well classify a subsequent but identical field differently.) Nevertheless, it may be appropriate to claim that these classifiers adapt to the style of each test field.

2.4 Supervised, Semi-supervised, and Unsupervised Learning

Algorithmic grouping or *clustering* of unlabeled patterns according to their distance to each other in feature space is often called *unsupervised* classification or learning [53]. Persistent attempts since the sixties to endow *self-organization*, (*self-*)*adaptivity*, *learning without a teacher*, *training without a trainer, self-produced pattern discrimination, self-correction*, and *unsupervised, semi-supervised* or *non-supervised classification* with a stable meaning have proved futile [54, 55, 56]. As mentioned above, we reserve the word *adaptive* for a more specific concept.

We take the position that pattern recognition in OCR and DIA cannot be entirely unsupervised, because documents, words, letters and numerals already have some prior meaning to human readers. At some point, this meaning must be communicated to the classifier so as to regain the correspondence between the arbitrary labels assigned by the machine and the labels of the user community (for instance ASCII character labels, or Reuters document categories). We attempt next to discover what is the "hidden" information used by various "unsupervised" pattern recognition methods.

The least information necessary to turn a mixture decomposition method into a classifier is admirably elucidated in [57, 58]. The unlabeled patterns are presented as a Gaussian mixture distribution with unknown mixing parameters. It is shown that with an increasing number of samples, the parameters of the constituent distributions can be estimated to arbitrary precision. However, in order to determine with better than chance accuracy which constituent corresponds to which class, we need at least *one* labeled pattern. Increasing the proportion of labeled to unlabeled samples brings such a classifier closer to the vanilla-flavored (supervised) classifier.

The idea of first partitioning unlabeled samples, and then assigning labels to each partition, was thoroughly explored in the sixties from the perspective of both signal-processing [59, 60, 61] and potential functions [62, 63, 64]. In 1966 Dorofeyuk presented several clustering algorithms, and then assigned labels to each cluster according to the known majority label in each cluster. He tested his algorithms on five classes of hand-printed digits. He called the procedure *teaching without a teacher*, because the labels were not used in the clustering process [65].

Examples of easy- and difficult-to-cluster pattern configurations are simple to visualize in two dimensions [66, 67]. The widely-used K-means clustering algorithm was popularized as a general method for “exploratory” multivariate analysis of unlabeled data [68, 69]. A variation that addressed some of the shortcomings of the elementary algorithm by splitting and merging classes was called Isodata [70]. In the communications community, iterative minimization of the sum-of-squared-error criterion became known as Vector Quantization [71, 72]. Among the first attempts at evaluating regiorously the effectiveness of clustering methods were Dubes and Jain [73]. Variations of the method with respect to initialization, cost function, splitting and merging clusters, and distance metrics, have been amply described [74, 75]. Current research focuses on combining multiple cluster configurations obtained by different algorithms, i.e., *clustering ensembles* [76].

Clustering with the K-means algorithm using *labeled seeds* (initial cluster centroids) circumvents the need for assigning labels after the clustering process. One of the simplest adaptive classifiers (called *decision-directed approximation* [77]) is a minimum-distance-to-class-centroid classifier that iteratively recomputes the class centroids according to the class labels assigned on the previous step (Fig 6). It is clear that the final class centroids depend, as do cluster centroids in K-means, only on the initial seeds (here the class means of the training set), and on the patterns in the test field. Several successful examples of decision-directed classification in OCR have been reported [78, 79, 80, 81, 82, 83]. Other applications include Morse Code transcription [84], adaptive equalization [85, 86], and thematic mapping in remote sensing [87, 88].

Crisp clustering algorithms assign each pattern to a single cluster. Fuzzy clustering algorithms assign membership functions. Agglomerative and divisive algorithms group or partition patterns according to a pre-calculated matrix of pairwise similarities (which need not have metric properties). Statistical methods based on the very general principle of Expectation Maximization [89, 90] attempt to decompose mixture distributions into their constituents. In each of these approaches, the problem is simplified if the number of classes is known. The number of classes may be replaced or augmented by other pertinent information, like constraints on the number of patterns per cluster, or on the cluster diameters.

We underline that any of the above methods for grouping patterns according to some predetermined measure of similarity may serve as the foundation for “unsupervised” classification. All of them exploit style consistency, albeit only intra-class style. Like other style classifiers, these methods reduce the need for labeled samples. The methods for addressing small-training-sample problems [2], [9] and ‘wrong’ (or non-representative) training-sample problems are essentially equivalent. However, classifiers based on only intra-class styles are obviously suboptimal when the data exhibits inter-class style as well.

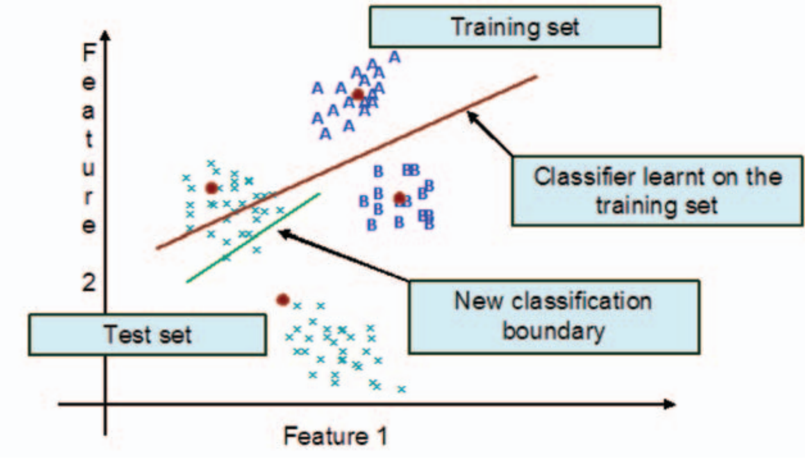
Training set
Classifier learnt on the training set
New classification boundary
Test set
Feature 2
Feature 1
A
B

Fig. 6. Adjustment of the classification boundary in a decision-directed classifier. The new boundary is at equal distance from the class centroids of the patterns as classified by the original classifier learn on the training set

Patterns are often clustered for multi-stage classification of large- alphabets, like Chinese [92, 93]. As a demonstration of the power of language context, all the characters in a document can be clustered, and labels assigned to the cluster labels by solving a substitution cipher, without using any prior class-related shape information [94, 95, 96, 97, 98].

Clustering is not necessarily followed by assigning an object label to each cluster. Clustering the connected components (most of which correspond to individual characters) in an isogenous text image is the basis for efficient text-image compression, such as DjVu and JBIG2 [99, 100, 101]. Clustering of approximate representations of document *words* was used to reduce the number of comparisons of a query versus document words in the multilingual word indexing scheme mentioned above [102]. The method was called *font-adaptive* because the words in each source were clustered separately.

3 Interaction in Training and Classification

Research aimed at fully automating the processing of document images has received sustained attention over the past 40 years. Nevertheless, any of the dozens of surveys to date (one of our favorites is [103]) will reveal that progress in automatic recognition and interpretation has been slower than predicted. Further improvement on cursive handwriting and degraded print may be even slower because the remaining challenges are harder. As in speech recognition, bridging the “semantic gap” between machine and human knowledge appears problematic. The context in all the varieties discussed above brought by humans to any classification task is much greater than what can be codified automatically from even the largest collections of training samples available to our community. Endowing fully automated systems with broad knowledge remains an elusive goal. Fortunately, in many applications it is not necessary to fully automate the task of document analysis. This may be the case when the focus is on a relatively few high-value documents (perhaps just one). The computer can play the role of an assistant to help the user acquire information that would otherwise remain inaccessible. While such documents could be collected and returned to a central repository for scanning and batch processing in the traditional manner, it may be advantageous to exploit the information immediately and *in situ*.

There are pronounced differences between human and machine cognitive abilities. A divide-and-conquer strategy for visual recognition can partition difficult domains into components that are relatively easier for both human and machine (Table I). Humans excel in gestalt tasks, like object-background separation. They apply to recognition a rich set of contextual constraints and superior noise-filtering abilities. They can also easily read degraded text (e.g., CAPTCHA's [104]) on which the best optical character recognition systems produce only gibberish. On the other hand, the study of psychophysics reveals that humans have limited memory and poor absolute judgment [105].

Computers can perform many tasks faster and more accurately. They can store thousands of images and the associations between them, and never forget a name or a label. They can compute geometrical properties like higher-order moments whereas a human is challenged to determine even the centroid of a complex figure. Spatial frequency and other kernel transforms can be easily computed to differentiate similar textures. Computers can count thousands of connected components and sort them according to various criteria (size, aspect ratio, convexity). They can quickly measure lengths and areas, and flawlessly evaluate multivariate conditional probabilities, decision functions, logic rules, and grammars. Nevertheless, computer vision systems have difficulty in recognizing "obvious" differences and they do not generalize well from limited training sets

We are *not* advocating here exploratory data analysis in feature space [106, 107], but operator interaction with displayed document images or parts thereof. Although we cannot clearly separate human interaction during training and testing (because when a human helps the system during classification time, it can be viewed as training) we attempt to categorize interaction as: (1) Human-initiated or Machine-initiated; (2) Durable or Ephemeral. *Durable Interaction* immediately alters some system parameters and therefore affects how the system deals with new data. *Ephemeral Interaction* merely labels new patterns or modifies the results of classification.

3.1 Examples of Human-Initiated Interaction

The most common example of human-initiated interaction is *labeling* training patterns. Another example is *word completion* on touch-screen devices (word

HUMAN	**MACHINE**
Dichotomies	
	Multi category classification
Figure-ground separation	
Part-whole relationships	
Salience	
	Non-linear high dimensional classification boundaries
Extrapolation from limited training samples	
Broad context	
	Precise mesaurement of individual features
	Enumeration
	Store and recall *many* labeled reference patterns
	Accurate estimation of statistical parameters
	Application of Markovian properties
	Estimation of decision functions from training samples
	Evaluation of complex sets of rules
Gauging *relative* size and intensity	
Detection of *significant* differences between objects	
	Computation of geometric moments
	Orthogonal spatial transforms (e.g. wavelets)
	Connected components analysis
	Sorting and searching
	Rank-ordering items according to a criterion
	Additive *white* noise, salt and pepper noise
Colored noise; Texture	
Non-linear feature dependence	
	Determination of local etrema in high-dimensions
Global optima in low dimensions	

Table 1. Comparison of relative strengths of human and machine in diverse aspects of visual pattern recognition

completion is seldom used with regular keyboards because it tends to distract the operator). Such interaction has also proved its value in the *vectorization* of engineering drawings and maps. We briefly describe these three applications.

Labeling Training Patterns in OCR

Nowadays all manual labeling of documents, or parts of documents, is carried out with computer display of digitized material, and can therefore be considered interactive. Indeed, considerable ingenuity has been applied to provide interfaces that speed up the process and reduce mislabeling. Commercial OCR firms strive to improve successive releases of their recognition systems by accumulating millions of labeled characters. If everything is keyed, it is human-initiated interaction. If they first OCR the training documents and only correct the errors, then the interaction is machine-initiated. It is *ephemeral* because the *current* classifier does not benefit from the newly labeled patterns.

Most OCR systems also provide at least limited facilities for additional training in the field for new shapes and new classes. If necessary, the operator can separate document segments set in different typefaces or written by different individuals. Entering only part of a document may help a recognition system designed with this in mind to fine-tune the classification algorithms. The underlying assumption is that if the remainder of the document(s) is from the same source, then the adjusted parameters will yield more accurate recognition. Training is not limited to characters: for example, a table-location algorithm can be trained via multi-parameter optimization [108].

Mobile Text Entry

It is clear that one bottleneck in mobile interactive document analysis is text entry. Without scanning or a regular keyboard, the alternatives are (1) virtual keyboard on a touch sensitive screen, (2) finger-operated keypad on arm or thigh (perhaps incorporated in the operator's clothing), and (3) automatic speech recognition. We believe that the stylus is the most appropriate solution, because in addition to text entry it can also mediate the graphical communication essential in other phases of document image analysis.

The virtual keyboard was invented in the seventies to avoid having the operator shift constantly between pointing device and keyboard while digitizing maps and line-drawings. It consisted of a picture of a keyboard that could be shifted to the area of the drawing being vectorized. Current virtual keyboards usually appear in a fixed partition of the touch-sensitive screen of a handheld device. Edwards' survey of input interfaces in mobile devices covers most of the relevant issues [109]. Data input is usually a local operation, so it makes little difference whether the device is networked or not.

Important considerations for stylus data entry are speed, operator comfort, and ramp-up time. The first two factors are influenced by the amount of space allocated to the keyboard, to the recognized or keyed text, and to control functions. The third factor depends heavily on the keyboard layout. The QWERTY layout, developed to prevent binding of type bars in mechanical typewriters, is suboptimal even for typing, and even more so for one-handed stylus entry.

Ancona gives a good overview of alternative keyboards and word-completion algorithms [110]. An upper bound on the speed of individual character entry is imposed by Fitts' Law, which is a nonlinear relationship between pointing time and the distance and size of the target. The relevant distance is that between the screen areas ("keys") corresponding to consecutive letters. The letter transition frequency is given by a language model. It is possible to reduce the average distance by having multiple keys for common symbols, but this decreases the size of the keys. Several researchers have optimized keyboard designs according to various language models [111, 112, 113, 114]. The computed speeds hover about 40 words per minute, but actual text entry is much slower.

The speed increase obtainable by word completion depends on the language model. Ancona (*op. cit.* [110]) demonstrates a keyboard with separate keys for the ten most common words (with a cumulative word frequency of 28%). After each tap on the screen, the ten most likely words appear in the selection area of the screen. If the correct word is included, it can be selected with one additional tap. If not, another letter is tapped, which brings up ten new words. With a vocabulary of 13,000 words, the expected number of taps per word was 3.3. The performance of word-completion systems depends on how well the stored lexicon is matched to the user input. Multiple lexicons – for different languages and applications – can be either stored on board, or downloaded via a wireless connection.

Vectorization

Entering line art (maps and engineering drawings) manually is even more laborious and expensive than keying text. Manual vectorization was first conducted from hardcopy on a digitizing table. The operator traced the lines with cross-hairs under a magnifying glass with a MARK button. After the advent of large-size roller-feed scanners and bitmapped displays all service bureaus and in-house operations adopted on-screen vectorization. Vectorized lines could now be displayed with a different color, deviations between the manually entered line segments and the original bitmap became clearly visible, and the operator could zoom in on dense portions of the drawing.

If most of the labels on a drawing or map cannot be recognized by OCR because of poor document quality or unusual character shapes, it is still possible to rapidly mark their location and orientation, rotate them to horizontal, and move them to a single area of the screen [115]. This accelerates manual label entry (a single E-size drawing may contain over 3,000 alphanumeric symbols). Most such data-entry systems are part of GIS or CAD software designed for standard workstations, where all graphical operator interaction is mediated by the mouse. As demonstrated by Engelbart and colleagues at SRI long ago, direct-action devices, like a touch-sensitive stylus, would allow faster and more accurate interaction [116]. However, the aspects of interest here are the machine-initiated algorithms developed for semi-automated data entry.

To enter colored maps, different color layers are first separated according to RGB values. Vectorizing algorithms are manually initialized to a line segment or curve, and automatically follow that line at least to the next intersection point. Some systems also attempt to automatically recognize map and drawing symbols (e.g., for schools or resistors). If it fails, the operator overrides it. The character recognition software recognizes cleanly lettered labels (elevations, part numbers, resistor values), but leaves labels confused by overlaid line art or poor lettering to the operator.

These interactive systems (like CAVIAR, below) exhibit clear speed advantages over completely manual data entry, and are robust enough (unlike automated systems) for operational application. Although some of these systems are laboriously trainable, one key difference compared to CAVIAR is that no commercial system that we are aware of incorporates active algorithms (i.e., durable interaction) that take advantage of routine operator input.

3.2 Machine-Initiated Interaction

All trainable systems incorporate, by definition, durable interaction. Most such systems, however, are human-initiated: training is a preliminary, separate phase from the recognition, without regard to what can be correctly or incorrectly recognized without additional training. We believe that eventually all interaction in DIA should be *machine-initiated* and *durable.* In other words, the operator should not even have to look at data that the system had no trouble in classifying, and every interaction should be utilized by the system to improve subsequent classification. We therefore present some of our work outside of DIA on machine-initiated, durable interaction. Then we propose several phases in DIA where we see potential applications of similar types of interaction.

Machine-Initiated, Durable Interaction in CAVIAR Systems

CAVIAR (Computer Assisted Visual Interactive Recognition) is an interactive system for recognizing faces and flowers, both problems of a level of difficulty (i.e., current automated accuracy) comparable to document recognition [117, 118, 119, 120, 121]. Experiments on sizable databases of faces and flowers indicate that interactive recognition is more than twice as fast as the unaided human, and yields an error rate ten times lower than state-of-the-art automated classifiers. The benefit margin of interactive recognition increases with improved automated classification. Parsimonious human interaction throughout the interpretation process is much better than operator intervention only at the beginning and the end, e.g., framing the objects to be recognized or dealing with rejects. Furthermore, this interactive architecture has been shown to scale up: it can start with only a single sample of each class, and it improves as recognized samples are added automatically to the reference database (decision-directed adaptation).

The notions embodied in CAVIAR differ in fundamental ways from past efforts at mobile, interactive recognition. Whether such an approach can be equally effective in the domain of documents as it is for flowers and faces is unproven, and adapting CAVIAR to document analysis requires further research. There are, however, other projects that share similar goals and assumptions. The Army Research Laboratory's Forward Area Language Converter (FALCon) system provides mobile optical character recognition (OCR) and translation capabilities [122, 123], but, so far as we know, it has a traditional user interface. Research on camera-based document acquisition is growing [124, 125]. However, this work, like FALCon, treats the later processing stages as though they will be fully automated.

Camera-based systems for locating and recognizing text in traffic signs and providing translation services for visitors to foreign lands are somewhat similar [126, 127], but their interaction paradigm is less integrated into classification than CAVIAR's. Reading systems for the vision-impaired likewise focus on page-at-a-time processing, but offer an auditory user interface [128, 129]. A somewhat similar notion is recent work on developing tools to support forensic document analysis [130]. Forensic systems are, however, intended for off-line use by domain experts (as opposed to opportunistic document readers whose primary jobs lie elsewhere), and have no need for mobility.

Potential for Machine-Initiated Durable Interaction in DIA

We mention some DIA tasks where CAVIAR-like systems may prove advantageous. We focus on scenarios where automated algorithms work accurately only on exceptionally clean documents, but where a little interaction can quickly produce acceptable results on ordinary material.

Binarization. Most OCR algorithms are designed for binarized images, because all scripts avoid discrimination based on shades of gray or color. Therefore documents must be converted to binary images after digitization to 8-bit gray scale or RGB. Global binarization algorithms work only if the foreground and background reflectance are uniform throughout the document, which may not be the case if part of a folded documents suffers prolonged exposure to sunlight, or if there are dark areas around the edges of a photocopy. Local binarization algorithms set the threshold according to the distribution of reflectance in a window translated through the page. The threshold estimates of the relative density and configuration of the foreground (ink) and background invariably depend on explicit or implicit assumptions that hold only for a narrow class of documents. An operator can easily tell when binarization fails. Setting the appropriate window size for local algorithmic thresholding requires far less work than setting the threshold manually everywhere, and it is more robust than fully automated local thresholding.

Page segmentation. Column, paragraph and line segmentation are other instances where interaction may be effective. The first step is usually estimating global document skew. While accurate skew estimation and correction algorithms have been developed for printed matter, they do not work well on handwriting because the orientation of individual lines varies, the margins are not straight, there may be only a few words on a page, or there may be several columns of words or phrases at different angles. Humans can, however, judge skew remarkably well, and convey this information to the computer by a few well chosen stylus taps or by rotating a superimposed grid. After the computer-proposed skew correction and line finding is corrected, the occasional merged pair of lines – due to overlapping ascenders and descenders – can be likewise rapidly separated.

Word segmentation. This is relatively easy for printed text, except for extremely tightly-set, micro-justified print. In handwriting, however, large spaces may appear within words. Towards the end of a line, words are often squeezed together. In Arabic and other scripts, some inter-letter spaces are mandatory. Underlined groups of words can further complicate the task. Again, humans can usually spot missed word boundaries even in unfamiliar languages and scripts. If the writing lines are already properly segmented, then a simple interface can be designed to correct linked and broken words.

Character recognition. An operator can provide global assistance to the character recognition system. He or she may be able to recognize the language or script of a document, indicate the average slant, and (in Western scripts) the prevalent case. The operator may decide which of the available lexicons would provide the best language model, and the chosen lexicons can be automatically updated with entries from the processed documents that have been deemed correct. Humans can also tell where perfect accuracy is important, as in telephone numbers, email addresses, and proper nouns and, if recognition fails, enter them manually or select them from the top recognition candidates. Finally, if the typeface is entirely outside the machine's repertory, it can cluster the character images, so that the operator need to label only a representative member of each cluster [131].

Active Learning: Machine-Initiated Durable Interaction During Training

During the training of pattern classifiers it is often feasible to provide labels for the training patterns incrementally. The most 'informative' patterns can be chosen iteratively, and their labels queried. This learning paradigm, wherein the learner is allowed to choose the information to be acquired, is called *active learning* and has been shown to significantly reduce the labeling cost, while preserving the accuracy of the trained classifier [132, 133]. Although we are not aware of any formal application of active learning in DIA (as opposed to document categorization), the training samples are often augmented when classification errors arise. This practice is seldom documented.

4 Technology and Applications

In this section we briefly discuss recent technological advances that alter the landscape of OCR and DIA and open up new applications.

4.1 Cameras and Displays

Solid state sensors are more sensitive to light than film. Current digital consumer-grade cameras, PDA cameras, and even cell-phone cameras with tiny lenses have comparable spatial sampling rate, geometric fidelity, and higher photometric range than desktop scanners of just a few years ago. Top of the line camera-phones already provide 5 mega pixels in color, which is sufficient for most A4 pages. The effects of non-uniform illumination can be mitigating by taking calibration pictures. We can therefore expect that most document acquisition will soon take place with 2-D sensor arrays rather than linear sensor sweep [134, 135].

High quality portable document acquisition systems (first for law enforcement and military applications) will require personal OCR, DIA, and document interpretation support systems [136]. Current defense interests are mainly in foreign-language documents and non-Latin scripts. Since the person acquiring the document is likely to have some expertise or at least interest in its contents, and images are not acquired in large quantities, increased interaction seems appropriate, at least in preprocessing. Interaction will be enhanced by direct action which allows pointing faster and more accurately than with a mouse, but hampered by the miniature screen. A letter-size document is certainly not readable on any camera-back display. Zooming and scrolling on both directions is impractical. Perhaps the new textile based displays will provide a satisfactory interface. Another alternative for interaction is notebook-sized touch-sensitive displays like the Tablet-PC.

Another topic of rising interest is reading text in videos, including road signs from car-mounted cameras [137, 138, 139, 140]. Such text is often in color and exhibits more geometric and photometric distortion than text scanned from paper. Furthermore, there is less context of every kind.

4.2 Web-Wide Data Accessibility

Rapidly increasing storage and communication capacity has led to a qualitative change in the nature of document image collections available for experimentation. The information retrieval community is alread making good use of web-based document collections to evaluate diverse approaches in the contests of the *Text Retrieval Conferences* (TREC) *Genomics Track*, the *Knowledge Discovery and Data Mining Cup*, and the *Creative Assessment of Information Extraction in Biology.* Image test databases typically contain at least two orders of magnitude fewer documents than test collections for information retrieval, extraction, categorization, and screening (because about a million bytes must be processed per page image, versus a few thousand bytes for an encoded page).

Most DIA research has been based on *ad hoc* collections of documents assembled by the researchers themselves because they are rich in aspects relevant to their particular research task. Although the collection, annotation and documentation of such test databases is not a trivial task, we seldom see much reuse by different groups of researchers, except possibly in Chinese and Japanese character recognition. This is likely to change as more and more research data sets are posted on the web. Large enough benchmarks would allow each test to be run on new, but statistically representative, samples. This would help avoid tuning algorithms to the test set, which is an almost inevitable consequence of open test collections of limited size [141].

Most applications must contend with highly repetitive material (for example, some firms do nothing but convert telephone directories to computable readable form). Nevertheless, many researchers strive for diversity within the constraints imposed by their task. Although this approach tests the range of applicability of the algorithms, it would also be desirable to experiment with adaptation on large, relatively homogeneous sets of document images that can now be readily found on the web.

Collection tools for DIA research require some database of digital libraries with downloadable page images, and a search engine capable of searching the database (or the whole web). The first step is the location of one or more collections with images of the desired type. (It is not always easy to tell, just by looking at a display, which pages are in image format, and which pages are in coded format). Whether *partial* processing of documents, such as type categorization, script or language recognition, contrast enhancement, skew detection and removal, segmentation at various levels (e.g. paragraph, line, word), table spotting, etc., is valuable by itself may be open to question, but there are certainly a great many researchers and publications engaged in pursuing such relatively narrow goal because end-to-end document processing requires a large team with varied resources. It would therefore be valuable to develop tools for the extraction of document collections with specific characteristics including degree of homogeneity or heterogeneity from digital libraries, and appropriate specifications of standard formats for intermediate results.

A small-scale study was reported on the *Making of America* collection (part of Cornell University's Digital Library), which at the time comprised 267 monographs (books) and 22 journals (equaling 955 serial volumes) for a total of 907,750 pages, making it three orders of magnitude larger than the datasets traditionally used in document analysis research (e.g., the UW1 CD-ROM).

Two tasks were evaluated: optical character recognition and table detection. In the case of the former, the textual transcriptions provided by the digital library (primarily for retrieval purposes) were used as the ground-truth, while for the latter, a visual inspection of the pages purported to contain tables was conducted, enabling precision (but not recall) measurements [142].

4.3 Digital Libraries

The *Million Book Project* at Carnegie Mellon is already well over the half-way mark. The Google consortium plans to digitize over 10.5 million unique books. The non-profit *Open Content Alliance*, initiated by the Internet Archive and Yahoo, proposes to provide broad access to non-proprietary world culture on paper. The CMU project produces only page images, but Google is experimenting with commercial OCR systems in dozens of languages with a view to provide searchable text. The *European Library* offers access to both digital and non-digital resources of the 45 national libraries of Europe. Most current digitization projects produce only page images: browsing digital libraries accessible through university libraries suggests that only a small fraction of their content has been transcribed. Crane addresses the issues of scale and sampling of quasi-infinite collections [143].

These “cultural” collections, which convert old books to computer-readable form, represent only part of the growth of digital libraries. Equally important are specialized collections for research, assembled from journals, conference proceedings and reports that are already in computer readable form. Some well known examples are: *ArXiv* for Physics, *DML* for Mathematics, *CiteSeer* for Computer Science, and *Medline* for medicine, but there are growing collections in every field of study.

In addition to web-wide access to cultural and technical collections, there are many novel services. Among the most popular are music servers, genealogical searches, and software that allows organizing and sharing personal photographs. Newspapers, radio and television stations offer access to their archives, and specialized search engines have been developed for sound effects [144]. Some of these require audio interaction, while many game sites and some web-based educational laboratories need a haptic modality. Nevertheless, we do not believe that multimedia will diminish the importance of digital sources of conventional printed information, and of related technologies. Current developments at the intersection digital library development and DIA include research opportunities in digitizing, coding, annotating, disseminating and preserving library documents [145].

4.4 Interoperability

A simple ASCII or Unicode file may be sufficient output for experiments on isolated digits and characters. But how should we code the output of an equation recognition system? Most researchers use either a proprietary format or TeX [146, 147, 148]. Both lack a good transition to analytical and numerical equation processing tools like Mathematica and Maple. A similar quandary arises with table recognition. Again, we would like a format that allows a smooth transition to a database query language. We favor *Wang Notation*, which provides a layout-independent representation of the relations between hierarchical category headers and content cells [149, 150]. For archival circuit diagrams and engineering drawings, the natural choice seems to be one of the widespread CAD formats (like Spice, Synopsis, and AutoCad).

Most business documents now carry XML tags, which facilitate their interpretation by whatever community agrees to the underlying convention. XML tags allows automated processing of equivalent fields, regardless of what they are called on the document. For instance, one tag may specify to whom payment should be routed, regardless of whether the name field is called vendor, provider, supplier, or seller. Digital libraries have evolved elaborate conventions for tagging metadata, beginning with the Library of Congress MARC format, and migrating from SGML to XML and the Dublin and SDLIP Cores [151]. Perhaps it is finally time for our research community to agree at least on XML schemes for "interoperability" [152]. This will also eventually help to relieve us from the tedious task of reading technical articles, which will be delegated to indefatigable autonomous agents.

XML tags have no actual meaning or semantics. The notion of meaning appears to require some kind of shared understanding of a topic. Since many current attempts to formalize meaning are focused on *ontologies*, ontological engineering may play a part in the extraction of concepts from documents [153].

4.5 Document Storage

We keep defining new units to keep up with the size and speed of storage devices. We can store book-length files and high-resolution pictures on devices that hang on our key chain. Whereas document image compression used to be a popular field of research that led to impressive increases in the compression ratio of mainly-text images, there is no longer any need to compress documents at the "retail" level. Large archives are still compressed, but communication links are so fast that they are often decompressed *before* retrieval.

Merely digitizing or coding something does not guarantee permanent access. For instance, many records from WWII were kept on punched cards. Not only did the punch cards disintegrate, but the card readers have disappeared. Magnetic tape and disk and optical media have a relatively short life. Furthermore, the software required to read the coded data may be incompatible with computers of another generation. It is not uncommon for engineering drawings prepared on earlier Computer Aided Design systems to be rescanned and revectorized, simply because the CAD software can no longer run on any available computer. Diskettes, tape cassettes, and ZIP drives are already obsolete. Until recently, many organizations opted for archiving documents on microfilm or microfiche instead of digital media. However, at current storage costs, it is plausible to keep everything *on line*. When the server is replaced, everything is copied, so there is no need to worry about removable media forgotten in some cabinet. Is the solution to keep every document spinning for ever?

5 Conclusions

This survey can be regarded only as samples of research selected from a large population according to prior probabilities that correspond only to the authors' own research interests. It is far from exhaustive or representative, and different topics are covered at different depths. In spite of the plethora of citations, it suffers from small-sample effects. Some of our samples may be distorted or mislabeled. As a training set for predicting research, it is highly biased. Nevertheless, we take the opportunity to list our impressions, based on imperfect training and grossly suboptimal recognition, of where the field is heading, and of what research problems should be addressed to reach the next stage.

5.1 Trends

Increasing processing power and storage capability by over a factor of 1000 during the last decade allows much greater use of context of all types. This leads naturally to exploiting common-source language and style constraints, i.e., field classification in OCR and joint processing of multiple tables, forms, figures and other document components in DIA. Further decreases in error rates are unlikely without adaptive/dynamic field classification. Although none of the underlying ideas are new, they can now be tested without access to supercomputers, and perhaps even incorporated into commercial OCR engines.

The availability of useful OCR and DIA software and inexpensive scanners on desktop computer systems means that all necessary interaction, including labeling training material, adjusting parameters, proofreading and corrections, is likely to be carried out by folks who would rather be doing something else. We do not believe that interaction can be eliminated in the foreseeable future, i.e., that most tasks of interest can be fully automated. This increases the premium on transparent and effective interaction. The demands on users can be alleviated by systems capable of taking full advantage of machine-initiated, durable interaction.

OCR and DIA capability is migrating to consumer-grade cameras, pocket computers and even to camera-phones. Although good interfaces that replace page-displays and keyboards are still lacking, many users will find these devices convenient enough to accept a touch-sensitive screen-and-audio interface for casual document capture.

5.2 Open Problems

Although both style and language constraints have been extensively investigated, we found little or no research on combining style, language constraints and order-dependent shape context. How can these diverse constraints be combined optimally?

In interactive dynamic recognition systems, both operator interventions and classification results can permanently change some of the classification parameters. Therefore the overall accuracy of such systems depends both on the quality of the interaction and on the order of arrival of patterns to be classified. How should such systems be evaluated?

Interactive visual classification benefits from the availability of a visual model for mediating communications between the operator and the machine. How can such visual models be constructed for new visual recognition tasks?

In all OCR and DIA systems with which we are familiar, feature sets are constructed by trial and error. How can a complete OCR and DIA system for a new language, script, and page layout, be generated automatically, starting only with a collection of labeled pixel maps?

Acknowledgement. George Nagy acknowledges support from the National Science Foundation under Award #0414854.

References

1. Grother, P.: Handprinted Forms and Character Database, NIST Special Database 19, technical report, March. 1995.
2. McCarthy, J.: Notes on formalizing contexts. In Kehler, T., and Rosenschein, S., eds., *Proceedings of the Fifth National Conference on Artificial Intelligence*, pp. 555–560. Los Altos, CA, Morgan Kaufmann, 1986.
3. Veeramachaneni, S., Sarkar, P., Nagy, G.: Modeling Context as Statistical Dependence, in *Procs. Modeling and Using Context: 5th International and Interdisciplinary Conference* CONTEXT 2005, Paris, France, July 5-8, (2005). Lecture Notes in Computer Science, Volume 3554, pp. 515–528, Jul 2005.
4. Bouquet, P., Serafini L.: Comparing formal theories of context in AI. *Artificial Intelligence*, (2004). 155: pp. 1–67.
5. Modeling and Using Context: Second International and Interdisciplinary Conference, CONTEXT'99, Trento, Italy, September pp. 9–11, (1999), Proceedings Lecture Notes in Computer Science Vol. 1688 Bouquet, P.; Serafini, L.; Brezillon, P.; Benerecetti, M.; Castellani, F. (Eds.)
6. Modeling and Using Context: 4th International and Interdisciplinary Conference, CONTEXT 2003, Stanford, CA, USA, June 23-25, 2003, Proceedings Series: Lecture Notes in Computer Science, Vol. 2680 Blackburn, P.; Ghidini, C.; Turner, R.M.; Giunchiglia, F. (Eds.) (2003).
7. Modeling and Using Context: 5th International and Interdisciplinary Conference, CONTEXT 2005, Paris, France, July 5-8, 2005, Proceedings Series: Lecture Notes in Computer Science, Vol. 3554 Dey, A.; Kokinov, B.; Leake, D.; Turner, R. (Eds.) (2005).

8. Yannakoudakis, E., Angelidakis, G.: An insight into the entropy and redundancy of the English dictionary, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6), 960–970, 1988.
9. Suen, C.Y.: N-gram statistics for natural language understanding and text processing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2), 164–172, 1979
10. Katz, S. M. : Estimation of probabilities from sparse data for the language model component of a speech recognizer, *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, March 1987.
11. Rice, S., Nagy, G., Nartker T.: Optical Character Recognition: An Illustrated Guide to the Frontier, Kluwer Academic Publishers, Boston/Dordrecht/London, 1999.
12. Hull, J.J., Srihari S.N.: Experiments in Text Recognition with Binary N-Gram and Viterbi Algorithms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 4(5), 520–530, Sept. 1982.
13. Hull, J.J.: A hidden Markov model for language syntax in text recognition. In *Proceedings of the Eleventh Conference on Pattern Recognition*, volume 2, 124–127, 1992.
14. Hull, J.J.: Incorporating language syntax in visual text recognition with a statistical model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(12):1251–1256, 1996.
15. Nagy, G.: Teaching a Computer to Read, *Proc. 11th Int'l Conf. Pattern Recognition,* vol. 2, pp. 225–229, 1992.
16. Raviv, J.: Decision Making in Markov Chains Applied to the Problem of Pattern Recognition, *IEEE Trans. Information Theory,* VOL. IT-13, no. 4, 536–551, 1967.
17. Toussaint, G. T.: The use of context in pattern recognition,*Pattern Recognition,* Vol. 10, 189–204, 1978.
18. Shinghal, R., Toussaint, G.T., Experiments in text recognition with the modified Viterbi algorithm, *IEEE Trans. Pattern Analysis and Machine Intelligence* 1(2), 184–193, 1979.
19. Shinghal, R., Toussaint, G.T.: The sensitivity of the modified Viterbi algorithm to the source statistics, *IEEE Trans. Pattern Analysis and Machine Intelligence* 2(2), 1181–1184, 1980.
20. Sinha, R.M.K., Prasada, B.: Visual Text Recognition through Contextual Processing, *Pattern Recognition*, 20(5), 463–479, 1988.
21. Sinha, R.M.K., Prasada, B., Houle, G. F., Sabourin, M.: Hybrid Contextural Text Recognition with String Matching, *IEEE Trans. Pattern Anal. Mach. Intell.* 15(9), 915–925 (1993).
22. Rabiner, L.R. : A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, 77(2), 257–286, 1989.
23. Gilloux, M., Leroux, M. Bertille J.M.: Strategies for Handwritten Words Recognition Using Hidden Markov Models, *Proc. Second Int'l Conf. Document Analysis and Recognition*, 299–304, 1993.
24. Kuo, S.S., Agazzi, O.E.: Visual keyword recognition using hidden Markov models, *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 329–334, 1993.
25. Nathan, K.A., Bellegarda, J.R., Nahamoo, D., Bellegarda, E.J.: On-Line Handwriting Recognition Using Continuous Parameter Hidden Markov Models, *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing,* vol. 5, 121–124, 1993.

26. MacKay, D.J.C., Peto, L.: A hierarchical Dirichlet language model. *Natural Language Engineering,* 1(3):1–19, 1994.
27. Bazzi, I., Schwartz, R, Makhoul, J.: An Omnifont Open-Vocabulary OCR System for English and Arabic, *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21(6) 495–504, June 1999.
28. Feng, S., Manmatha, R., McCallum, A.: Exploring the Use of Conditional Random Field Models and HMMs for Historical Handwritten Document Recognition, *Proc. 2nd IEEE International Conference on Document Image Analysis for Libraries,* DIAL 2006, Lyon, France, April 2006.
29. Sarkar, P., Nagy, G.: Classification of Style-Constrained Pattern-Fields, *Proc. 15th Int'l Conf. Pattern Recognition*, 859–862, 2000.
30. Sarkar, P., Nagy, G.: Style Consistency in Isogenous Patterns, *Proc. Sixth Int'l Conf. Document Analysis and Recognition*, pp. 1169–1174, 2001.
31. Sarkar, P., Nagy, G.: Style Consistent Classification of Isogenous Patterns, *IEEE Trans. Pattern Analysis and Machine Intelligence,* 27(1), Jan. 2005.
32. Veeramachaneni, S., Nagy, G.: Analytical Results on Style-constrained Bayesian Classification of Pattern Fields, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29(7) 1280–1285, July 2007.
33. Spitz, A.L.: An OCR based on character shape codes and lexical information, *Proceedings of the Third International Conference on Document Analysis and Recognition* (Volume 2) Volume 2, Page: 723, 1995.
34. Spitz, A. L., Maghbouleh, A.: Text Categorization using Character Shape Codes, *SPIE Symp on Electronic Image Science and Technology,* San Jose, pp. 174–181, 2000.
35. Ho, T.K., J.J. Hull, S.N. Srihari: A Computational Model for Recognition of Multifont Word Images, *Machine Vision and Applications 5,* 157–168, 1992.
36. Ho, T.K., J.J. Hull, S N. Srihari: A Word Shape Analysis Approach to Lexicon Based Word Recognition, *Pattern Recognition Letters 13*, 821-826, 1992.
37. Hong, T., Hull, J.J.: Visual Inter-Word Relations and Their Use in OCR Postprocessing, Proc. Third Int'l Conf. Document Analysis and Recognition, vol. 1, pp. 442–445, 1995.
38. Cesarini, F., Gori, M., Marinai, S., Soda, G.: INFORMys: A flexible invoice-like for reader system, *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 20(7), 730–745, July 1998
39. Marinai, S., Marino, E., Soda, G.: Font Adaptive Word Indexing of Modern Printed Documents, *IEEE Trans. Pattern Recognition and Machine Intelligence* 28(8), 1187–1199, August 2006.
40. Shi, H., Pavlidis, T.: Font Recognition and Contextual Processing for More Accurate Text Recognition, *Proc. Fourth Int'l Conf. Document Analysis and Recognition*, vol. 1, pp. 39–44, 1997.
41. Zramdini, A.W., Ingold, R.: Optical Font Recognition from Projection Profiles, *Electronic Publishing* 6(3): 249–260 (1993).
42. Zramdini, A.W., Ingold, R.: Optical Font Recognition Using Typographical Features, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(8), 877–882, Aug. 1998.
43. Bapst, F., Ingold, R.: Using Typography in Document Image Analysis. In *Proc. Raster Imaging and Digital Typography* (RIDT'98), Saint-Malo (France), pp. 240–251, 1998.

44. Srihari, S.N., Bandi, K., Beal, M.: A Statistical Model for Writer Verification, *Proc. Int. Conf. on Document Analysis and Recognition* (ICDAR-05) Seoul, Korea, August 2005.
45. Kawatani, T.: Character Recognition Performance Improvement Using Personal Handwriting Characteristics, *Proc. Third Int'l Conf. Document Analysis and Recognition,* vol. 1, pp. 98–103, 1995.
46. Veeramachaneni, S., Nagy, G.: Style Context with Second-Order Statistics, *IEEE Trans. Pattern Analysis and Machine Intelligence,* 27(1), Jan. 2005.
47. Andra, S.: Non-parametric approaches to style-consistent classification, Rensselaer Polytechnic Institute PhD dissertation, December 2006.
48. Andra, S., Nagy, G.: Combining Dichotomizers for MAP Field Classification, *Proceedings of International Conference on Pattern Recognition-XVIII*, Hong Kong, September 2006.
49. Widrow, B, Hoff, M.E.: Adaptive switching circuits, *1960 IRE WESCON Conv. Record,* Part 4, 96-104, 1960.
50. Aizerman, M.A., Braverman, E.M., Rozonoer, L.I.: The Robbins-Monroe process and the method of potential functions, *Automation and Remote Control* 26, 1882–1885, November 1965.
51. Tsypkin, Y. Z.: Adaptation, training, and self-organization in automatic systems, *Automation and Remote Control,* vol. 27, pp. 1652, January 1966.
52. Fu, K.S.: Learning techniques in pattern recognition systems, in Pattern Recognition (L.N. Kanal, ed.) Thompson Book Company, Washington, 1968.
53. Jain, A., Duin, R., Mao, J.: Statistical Pattern Recognition A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
54. Zadeh, L.A.: On the definition of adaptivity, *Proceedings of the IRE* 51, #3. 469–470, 1963.
55. Lendaris, G.G.: On the Definition of Self-Organizing Systems, *Proceedings of the IEEE 52*, 3, March, 1964
56. Nagy, G.: Pattern Recognition IEEE 1966 Workshop, *IEEE Spectrum*, pp. 92–94, February 1967.
57. Castelli, V., Cover, T.: On the exponential value of labeled samples, *Pattern Recognition Letters* 16, 105–111, 1995.
58. Castelli, V., Cover, T.: The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter, *IEEE-Trans. Information Theory* 42(6), 2101–2117, 1996.
59. Scudder, H.J.: Probability of error of some adaptive pattern-recognition machines, *IEEE. Trans. Information Theory* IT-11, 363–371, July 1965.
60. Spragins, J.: Learning without a teacher, *IEEE Trans. Information Theory,* vol. IT-12, 223–229, April 1966.
61. Stanat, D.F.: Unsupervised learning of mixtures of probability functions, in Pattern Recognition (L.N. Kanal, ed.) Thompson Book Company, Washington, 1968.
62. Aizerman, M.A., Braverman, E.M., Rozonoer, L.I.: The probability problem of pattern recognition learning and the method of potential functions, *Automation and Remote Control* 25, 1175–1192, September 1964.
63. Braverman, E.M.: Experiments on machine learning to recognize visual patterns, translated from *Automat. i Telemekh.,* vol. 23, pp. 349–364, March 1962, *Automation and Remote Control,* vol. 23, 315–327, 1962.

64. Braverman, E.M.: The method of potential functions in the problem of training machines to recognize patterns without a trainer, *Automation and Remote Control,* vol. 27, 1748-1771, October 1966.
65. Dorofeyuk, A.A.: Teaching algorithm for a pattern recognition machine without a teacher, based on the method of potential functions, *Automation and Remote Control,* vol. 27, 1728–1737, October 1966.
66. Nagy, G.: State of the Art in Pattern Recognition,*Proceedings of the IEEE* 56, #5, 336–362, May 1968.
67. Jain, A.K.: Cluster Analysis, Chapter 2 in *Handbook of Pattern Recognition and Image Processing* (K-S Fu and T-Y Young, eds), Academic Press, NY 1986.
68. Ball, G.H.: Data analysis in the social sciences: What about the details? *Procs. Fall Joint Computer Conference,* pp. 533–560, Spartan Books, 1965.
69. MacQueen, J.: Some methods for classification and analysis of multivariate observations, *Proc. 5th Berkeley Symp on Statistics and Probability,* pp. 281-297, Berkeley, CA University of California Press, 1967.
70. Ball, G.H., Hall, D.J.: A clustering technique for summarizing multivariate data, Behavioral Science, 12, pp. 153–155, March 1967.
71. Linde, Y., Buzo, A., Gray, R.M. : An algorithm for vector quantization design, *IEEE Trans. Comm.* 28, 84–95, 1980
72. Gersho, A., Gray, R. M. : Vector Quantization and Signal Compression, The International Series in Engineering and Computer Science, 1991.91
73. Dubes, R., Jain, A.K.: Validity studies in clustering methodologies, *Pattern Recognition 11*, 235–254, 1979.
74. Jain, A.K., Dubes, R.: Algorithms for Clustering Data, Prentice Hall 1988.
75. Theodoridis, S., Koutroumbas, T.: Pattern Recognition, Academic Press, 1999.
76. Topchy, A., Jain, A.K., Punch, W.: Clustering Ensembles: Models of Consensus and Weak Partitions, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(12), 1866–1881, Dec 2005.
77. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. New York: John Wiley and Sons, 2001.
78. Nagy, G., Shelton, G.L. : Self-Corrective Character Recognition System, *IEEE Transactions on Information Theory* IT-12, #2, pp. 215–222, April 1966.
79. Baird, H.S., Nagy, G.: A Self-Correcting 100-Font Classifier, Document Recognition, *Proc., IS&;T/SPIE Symp. on Electronic Imaging: Science Technology*, San Jose, CA, February 6-10, 1994, L. Vincent and T. Pavlidis, eds., vol. 2181, pp. 106–115, 1994.
80. Breuel, T., Mathis, C.: Classification Using a Hierarchical Bayesian Approach, *Proc. 16th Int'l Conf. Pattern Recognition*, 40103–40106, Aug. 2002.
81. Sarkar, P., Baird, H.S., Zhang, X.: Training on Severely Degraded Text- Line Images, *Proc. Seventh Int'l Conf. Document Analysis and Recognition*, pp. 38–43, Aug. 2003.
82. Veeramachaneni, S., Nagy, G.: Adaptive Classifiers for Multisource OCR, *Int'l J. Document Analysis and Recognition,* 6(3), 154–166, Aug. (2004).
83. Marosi, I., Tóth, L.: OCR Voting Methods for Recognizing Low Contrast Printed Documents: in *Proc. 2nd IEEE International Conference on Document Image Analysis for Libraries,* DIAL 2006, Lyon, France, April 2006.
84. Gold, B.: Machine recognition of hand-sent Morse code, *IRE Trans. Information Theory*, vol. IT-5, pp. 17–24, March 1959.
85. Lucky, R. W.: Automatic Equalization for Digital Communication. *Bell Systems Technical Journal,* 44:547–588, 1965.

86. Lucky, R. W.: Techniques for adaptive equalization of digital communication systems. *Bell Systems Technical Journal,* 45:255–286, February 1966.
87. Nagy, G., Tolaba, J.: Nonsupervised Crop Classification through Airborne Multispectral Observations, *IBM Journal of Research and Development 16,* #2, pp. 138–153, March 1972.
88. Shahshani, B.M., Landgrebe, D.A.: Asymptotic improvement of supervised learning by utilizing additional unlabeled samples: normal mixture density case, *Proc. SPIE Vol. 1766, p. 143–155, Neural and Stochastic Methods in Image and Signal Processing,* Su-Shing Chen; Ed., 1992.
89. Dempster, A.P., Laird, M.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm, *J Royal Statistical Soc.*, vol. 39, no. 1, pp. 1–38, 1977.
90. Redner, R.A., Walker, H.F.: Mixture densities, maximum likelihood, and the EM algorithm, *SIAM Review* 26, 2, pp. 195–235, 1984.
91. Raudys, S., Jain, A.K.: Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners, *IEEE Trans. on Patt. Anal. and Machine Intell.,* 13(3), 252–264, 1991.
92. Casey, R.G., Nagy, G.: Recognition of Printed Chinese Characters, *IEEE Transactions on Electronic Computers EC-15,* #1, pp. 91–101, February 1966.
93. Liu, C.-L., Jaeger, S., Nakagawa, M.: On line recognition of Chinese characters: the State of the Art, *IEEE Trans. Pattern Analysis and Machine Intelligence* 26, 2, pp. 198–213, (2004).
94. Casey, R.G., Nagy, G.: Autonomous Reading Machine, *IEEE Transactions on Computers* C-17, #5, pp. 492–503, May 1968.
95. Casey, R.G., Nagy, G.: Advances in Pattern Recognition, *Scientific American* 224, #4, pp. 56–71, 1971.
96. Casey, R.G.: Text OCR by Solving a Cryptogram, *Proc. Eighth Int'l Conf. Pattern Recognition,* pp. 349–351, 1986.
97. Nagy, G., Seth, S., Einspahr, K., Meyer, T.: Efficient Algorithms to Decode Substitution Ciphers with Applications to OCR,*Proceedings of International Conference on Pattern Recognition*, vol. 8, 352–355, Paris, October 1986.
98. Ho, T.K., Nagy, G.: OCR with no shape training,*Proceedings of International Conference on Pattern Recognition-XV*, vol. 4, pp. 27–30, Barcelona, September 2000.
99. Ascher, R.N., Nagy, G.: A Means for Achieving a High Degree of Compaction on Scan-Digitized Printed Text, *IEEE Transactions on Computers* C-23, #11, pp. 1174–1179, October 1974.
100. Bottou, L. Haffner, P., Howard, P., Simard, P., Bengio, Y., LeCun, Y.: High Quality Document Image Compression with DjVu, *Journal of Electronic Imaging*, vol. 7, no. 3, pp. 410–425, July 1998.
101. Witten, I., Moffat, A., Bell, T.: Managing Gigabytes, Academic Press 1999.
102. Marinai, S., Marino, E., Soda, G.: Font Adaptive Word Indexing of Modern Printed Documents, *IEEE Trans. Pattern Recognition and Machine Intelligence* 28(8), pp. 1187–1199, August 2006.
103. Nagy, G.: Twenty Years of Document Image Analysis in IEEE PAMI, *IEEE Trans. Pattern Analysis and Machine Recognition* 22(1), 38–62, January 2000.
104. Baird, H.S., Lopresti, D. P., editors: Human Interactive Proofs, *Procs. Second International Workshop,* volume 3517 of LNCS, 2005.
105. Miller, G.: The magical number seven plus or minus two; some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

106. Sammon, J.W.: Interactive pattern analysis and classification, *IEEE Trans. Computers* C-16, 594-616, July 1970.
107. Ho, T.K.: Exploratory Analysis of Point Proximity in Subspaces, *Proceedings of the 16th International Conference on Pattern Recognition,* Quebec City, August 11–15, 2002.
108. Cesarini, F., Marinai, S., Sarti, L., Soda, G.: Trainable table location in document images, *Proceedings of International Conference on Pattern Recognition-XVI*, Vol. 3, Quebec City, 236–240, 2002.
109. Edwards, J.. New interfaces: Making computers more accessible. IEEE *Computer*, pages 12–14, December 1997.
110. Ancona, M., Locati, S., Mancini, M., Romagnoli, A., Quercini, G.: Comfortable textual data entry for PocketPC: the WTX system. In Advances in Graphonomics, *Proceedings of International Graphonomics Symposium 2005*, Salerno, Italy, June 2005.
111. Langendorf, D.J.: Textware solution's Fitaly keyboard v1.0 easing the burden of keyboard input. *WinCELair Review,* February 1998.
112. Masui, T.: An efficient text input method for pen-based computers. In *Proceedings of the ACM Conference on Computer-Human Interaction*, pages 328–335, 1998.
113. James, C.L., Reischel, K.M.: Text input for mobile devices: Comparing model prediction to actual performance. In *Proceedings of the ACM Conference on Computer-Human Interaction,* pages 365–371, 2001.
114. Mackenzie, S., Soukore, W.: Text entry for mobile computing: Models and methods, theory and practice, *Human-Computer Interaction,* 17:147–198, 2002.
115. Nagy, G., Li, L., Samal, A., Seth, S., Xu, Y.: Integrated text and line-art extraction from a topographic map.*International Journal on Document Analysis and Recognition*, 2(4):177–185, June 2000.
116. English, W.K., Engelbart, D.C., Berman, M.L.: Display-selection techniques for text manipulation. *IEEE Transactions on Human Factors in Electronics,* HFE-8(1):5–15, March 1967.
117. Zou, J. Nagy, G.: Interactive visual pattern recognition,*Proceedings of the 17th International Conference on Pattern Recognition*, XVI, IEEE Computer Society Press, Vol. III, pp. 478–481, Aug. 2002.
118. Zou, J. Nagy, G.: Evaluation of model-based interactive ower recognition. In *Proceedings of the 17th International Conference on Pattern Recognition,* volume 2, pages 311–314, (2004).
119. Cha, S.-H., Evans, A., Gattani, A., Nagy, G., Sikorski, J., Tappert, C., Thomas, P., Zou, J.: Computer Assisted Visual Interactive Recognition (CAVIAR) technology. In *Proceedings of the IEEE Electro/Information Technology Conference, May 2005.* PDF
120. Nagy, G.: Interactive, Mobile, Distributed Pattern Recognition, *Proc. of the 13th International Conference on Image Analysis and Processing* ICIAP, Cagliari, Italy, LNCS 3617, 37–49, 2005.
121. Zou, J. Nagy, G.: Human-computer interaction for complex pattern recognition problems, to appear in Data Complexity in Pattern Recognition, Springer Verlag, Editors: Mitra Basu, Tin Kam Ho, Publication Date: Dec. 2006.
122. Holland, M., Schlesiger, C.: High-mobility machine translation for a battlefield environment. In *Proceedings of NATO/RTO Systems Concepts and Integration Symposium*, volume 15, pages 1–3, Monterey, CA, May 1998.

123. Swan, K.: FALCon: Evaluation of OCR and machine translation paradigms, August 1999. http://www.arl.army.mil/seap/reports/kreport.pdf. (accessed on 8/8/06)
124. Fisher, F.: Digital camera for document acquisition. In Symposium on Document Image Understanding Technology, Columbia, MD, 2001. http://www.dtic.mil/matris/sbir/sbir022/a038.pdf. (accessed on 8/8/06)
125. Jacobs, C., Simard, P.. Low resolution camera based OCR. International Journal on Document Analysis and Recognition. To appear. (2004)
126. Fujisawa, H., Sako, H., Okada, Y., Lee, S.: Information capturing camera and developmental issues. In*Proceedings of the Fifth International Conference on Document Analysis and Recognition* (ICDAR'99), pages 205–208, Bangalore, India, September 1999.
127. Yang, J., Chen, X., Zhang, J., Zhang, Y., Waibel, A.: Automatic detection and translation of text from natural scenes. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP '02), May 2002.
128. Hedgpeth, T., Rush, M., Black, J., Panchanathan, S.: The iCare project reader. In *Procs. Sixth International ACM SIGACCESS Conference on Computers and Accessibility*, October (2004).
129. Peters, J.P., Thillou, C., Ferreira, S.: Embedded reading device for blind people: a user-centred design. In *Procs. IEEE Emerging Technologies and Applications for Imagery Pattern Recognition* (AIPR 2004), pages 217–222, (2004).
130. Srihari, S., Shi, Z.: Forensic handwritten document retrieval system. In *Procs. First International Workshop on Document Image Analysis for Libraries* (DIAL'04), pages 188-194, (2004).
131. Xu, Y., Nagy, G.: Prototype Extraction and Adaptive OCR, *IEEE Trans. Pattern Analysis and Machine Intelligence* Vol. 21, 12, pp. 1280–1296, Dec. 1999.
132. MacKay, D.: Information-based objective functions for active data selection. *Neural Computation,* Vol. 4, No. 4, pp. 590–604, 1992.
133. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. *Machine Learning* 28, 133–168, 1997
134. Liang, J., Doerman, D., .Li, H.: Camera-based analysis of text and documents: a survey, *International Journal on Document Analysis and Recognition,* 7(2-3), 84–104, July 2005.
135. Pollard, S., Pilu, M.: Building cameras for capturing documents, *International Journal on Document Analysis and Recognition,* 7,(2–3), 123–137, July 2005.
136. Lopresti, D., Nagy, G.: Mobile Interactive Support System for Time-Critical Document Exploitation, *Procs. Symposium on Document Image Understanding,* College Park, MD November 2005.
137. Gandhi, T., Kasturi, R., Antani, S.: Application of planar motion segmentation for scene text extraction. In *Proc. of the ICPR*, 2000, I: 445–449.
138. Myers, G., Bolles, R., Luong, Q.-T., Herson, J.: Recognition of text in 3-D scenes. In *Proc. of the 4th Symp. on Document Image Understanding Technology*, pp. 23–25, 2001.
139. Wu, W., Chen, X., Yang, J.: Incremental Detection of Text on Road Signs from Video with Application to a Driving Assistant System, *Proceedings of ACM Multimedia 2004* (MM2004), pp. 852–859 (2004).

140. Yamaguchi, T., Maruyama, M., Miyao1, H., Nakano, Y.: Digit recognition in a natural scene with skew and slant normalization, *International Journal on Document Analysis and Recognition*, 7(2-3), 168–177, July 2005.

141. Salzberg, S.L.: On comparing classifiers: Pitfalls to avoid and a recommended approach, *Data Mining and Knowledge Discovery 1*, 317–327, 1997.

142. Lopresti, D., Zhou, J.: Document analysis and the World Wide Web, In *Proceedings of the Second IAPR Workshop on Document Analysis Systems,* pages 651–659, Malvern, PA, Oct. 1996.

143. Crane, G.: What Do You Do with a Million Books? *D-Lib Magazine* 12(3), ISSN 1082–9873, March 2006.

144. Rice, S.V., Bailey, S.M.: A Web Search Engine for Sound Effects, in *Proceedings of the 119th Convention of the Audio Engineering Society,* Paper #6622, New York, (2005) (PDF).

145. Nagy G., Lopresti, D.: Interactive Document Processing and Digital Libraries, *Proc. 2nd IEEE International Conference on Document Image Analysis for Libraries*, Lyon, France, IEEE Press, 2006.

146. Chou, P.A.: Recognition of equations using a two-dimensional stochastic context-free grammar. In W. A. Pearlman, editor, *Visual Communications and Image Processing* IV, vol. 1199 of SPIE Proceedings Series, 852–863, 1989.

147. Twaakyondo, H.M., Okamoto, M.: Structure analysis and recognition of mathematical expressions, *Proc. third Inter. Conf. on Document Analysis and Recognition,* ICDAR'95, Montral, Canada, pp. 430–437, 1995.

148. Zanibbi, R., Blostein, D., Cordy, J.R.: Recognizing Mathematical Expressions Using Tree Transformation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (11), 1455–1467, 2002.

149. Wang, X.: Tabular abstraction, editing, and formatting, PhD dissertation, University of Waterloo, Canada, 1006.

150. Embley, D., Lopresti, D., Nagy, G.: Notes on Contemporary Table Recognition, Document Analysis Systems VII, 7th International Workshop, *Procs. DAS 2006*, Nelson, New Zealand, February 13–15, 2006, Horst Bunke, A. Lawrence Spitz (Eds.) LNCS 3872, pp. 164–175 Springer 2006.

151. Macgregor, G., McCulloch, E.: Collaborative tagging as a knowledge organisation and resource discovery tool, *Library Review,* Volume: 55 Issue: 5 Page: 291–300, 2006

152. Hitz, O., Robadey, L., Ingold, R.: Using XML in Document Recognition, In *Proc. Document Layout Interpretation and its Applications* (DLIA'99), Bangalore (India), 1999.

153. Tijerino, Y.A., Embley, D. W., Lonsdale, D. W., Nagy, G.: Towards Ontology generation from tables, *World Wide Web Journal* 8, 3, Springer, September 2005.