Document Recognition Without Strong Models

Henry S. Baird

Computer Science & Engineering Department
Lehigh University, Bethlehem, Pennsylvania USA
www.cse.lehigh.edu/˜baird

***Abstract*—Can a high-performance document image recognition system be built without detailed knowledge of the application? Having benefited from the statistical machine-learning revolution of the last twenty years, our architectures rely less on hand-crafted special-case rules and more on models trained on labeled-sample data sets. But urgent questions remain. When we can't collect (and label) enough real training data, does it help to complement them with data synthesized using generative models? Is it ever completely safe to rely on synthetic data? If we can't manage to train (or craft) a single complete, near-perfect, application-specific "strong" model to drive recognition, can we make progress by combining several imperfect or incomplete "weak" models? Can recognition that is carried out jointly over weak models perform optimally while still running fast? Can a recognizer automatically pick a strong model of its input? Must we always pre-train models for every kind ("style") of input expected, or can a recognizer adapt to unknown styles? Can weak models adapt autonomously, growing stronger and so driving accuracy higher, without any human intervention? Can one model "criticize"—and then proceed to correct—other models, even while it is being criticized and corrected in turn by them? After twenty-five years of research on these questions we have partial answers, many in the affirmative: in addition to promising laboratory demonstrations, we can take pride in successful applications. I'll illustrate the evolution of the state of the art with concrete examples, and point out open problems.**

> ***(Based on work by and with T. Pavlidis, T. K. Ho, D. Ittner, K. Thompson, G. Nagy, R. Haralick, T. Hong, T. Kanungo, P. Chou, D. Lopresti, G. Kopec, D. Bloomberg, A. Popat, T. Breuel, E. Barney Smith, P. Sarkar, H. Veeramachaneni, J. Nonnemaker, and P. Xiu.)***

***Keywords*-Rule-based recognition, model-driven recognition, learning models, synthetically generated data, strong versus weak models, joint recognition, style-conscious recognition, adaptive recognition, anytime algorithms, whole-book recognition.**

I. Introduction

Theo Pavlidis advises us to "approach pattern recognition as an engineering problem and try to solve important special cases" rather than "looking for silver bullets that will solve [...] general problems," and he emphasizes "the need to understand the nature of [each] problem and the desirability of models" [1]. This rings true to me: I've often seen that the more application-specific knowledge that is designed into a system, the more accurate it becomes. Of course, such modeling efforts can be engineering-intensive and even unaffordable; sometimes we can't find enough data; or our best algorithms for exploiting models may be disappointingly suboptimal, or run too slowly. So how closely to model a given problem (say, recognizing printed text within a certain family of document images) is a critical engineering issue. I find it helpful to contrast "strong" and "weak" models:

- *strong models:* application-specific, an exact fit to the problem, often formal and detailed; and
- *weak models:* generic, broadly applicable to other problems too, often informal and imprecise.

It's no surprise to find that strong models support high accuracy but are often expensive to implement, and weak models tend to be cheaper but often yield lower accuracy. Usually, I suspect, we feel forced to choose one over the other—but I will argue here that we may be able to have the best of both. A series of insights, often surprising to me at the time, by dozens of researchers, accumulating for over twenty-five years, make me optimistic that affordable weak models can be made to combine, support, and even criticize one another, adapting and becoming stronger and so driving accuracy higher—all fully automatically. In short, this line of research suggests how high-performance document image recognition systems can be built with only small investments in application-specific modeling.

II. Models in Document Recognition

Models widely used to guide the recognition of text within document images include of course at least:

- *iconic* models of shape and ideal image formation given by writing systems and character (glyph) formation rules, typeface artwork and handwriting styles;
- *linguistic* models of language such as morphology, inflection, character and word n-gram statistics, dictionaries, and syntax rules;
- *segmentation* models for how to break words into characters and text lines into words (varying with writing system and typeface);
- *layout* models of the physical (geometric) and logical (functional) organization of page images (a higher level of segmentation); and
- *image quality* models of distortions and defects arising during printing and imaging.

One would like to add *pragmatic* and *semantic* models, as computational linguists do, but with rare exceptions (I discuss a couple below) the document recognition community has not yet been able to do so.

The distinction between "strong" and "weak" models that I make throughout this paper may be easier to appreciate—I doubt that it can be precisely formalized—through some examples:

- *iconic*: a model of exactly (& only) the typefaces occurring in the input document, I call strongest; if of different faces, or if it covers many irrelevant faces, I call weaker.
- *linguistic*: a dictionary containing exactly the words found in the input (a "perfect lexicon"), is strongest; a dictionary that is a subset or superset of the document's word-list, is weaker.
- *image quality*: a model of just those the degradations that affected the input is strongest; if of a range of degradations any of which do not occur in the input, weaker.

So, a "strong" model, in my usage[1], is a *close fit to the particular input* that the recognition system happens to be working on: it is customized or specialized to it. And a "weak" model is a poor fit, either because it is incomplete or even over-complete (covering a wider range of characteristics than are found in the input).

In this view, a model which is more versatile than it needs to be for its present input is weak, even though that may meet many of its design goals: for example, commercial page readers must perform reasonably well on a wide range of documents, so for any particular document, its default models are inevitably weak. So strength depends on the expected input, which may be as unbounded as all postal code addresses, or as constrained as a single machine-printed character.

Different degrees of strength can be found among competing models for any given input (whether the input is long or short, diverse or unvarying). I will argue below that the stronger the model is for an input, the higher the accuracy of the system running on that input is likely to be; but also the higher the engineering cost of acquiring that model. And consistent with this: relatively weak models are often easier to acquire but yield lower accuracy.

III. Acquiring Models

Our R&D community has benefited greatly from the statistical machine-learning revolution of the last twenty years [2], [3], [4]: as a result we rely less often on hand-crafting special-case rules than on learning models from labeled samples (the reverse of the usual practice in the 80's and much of the 90s). I will not talk (much) about hand-crafted rule-based models because they are an unattractive extreme case: they are not only costly to create, but difficult (and eventually impossible) to improve as the expected input changes.

There is of course a rich history of document image recognition guided by probabilistic models of several types, well typified by the literature on word recognition, *e.g.* Decerbo, Natarajan *et al* [5], Breuel *et al* [6], Weinman *et al* [7], Susuki *et al* [8], and Hamamura *et al* [9]. Most of these assumed perfect knowledge of the models, and focused principally on how to apply them to reduce errors. But, interestingly, Susuki *et al* and Weinman *et al* attempted to detect model imperfections, but not to adapt the models themselves. I felt for a long time that there should be ways not only to estimate error rates and even to identify candidate errors, but then to correct these errors automatically; this was an increasingly urgent open question through the 2000s.

IV. Strong Models: High-performing but Costly

At the risk of belaboring the obvious, I will discuss a couple of cases which illustrate the point that strong models tend to be costly to acquire but support high accuracy.

A. National Postal Code Readers

The most ambitious, and arguably the most successful, document recognition R&D projects have all been national-scale efforts to automate postal address reading. Three projects with strikingly similar goals were launched in the 1980s: one in Germany, led by Dr. Jürgen Schürmann (AEG Telefunken/ElectroCom); one in the U.S., led by Professor Sargur Srihari (SUNY Buffalo); and one in Japan, led by Dr. Hiromichi Fujisawa (Hitachi CRL). All three leaders were simultaneously researchers and administrators; all were deeply involved in ICDAR professional activities; their careers were all nearly consumed by their huge projects; and all their projects were triumphantly successful. The huge scale of these technical efforts is suggested by Srihari's record of advising 34 Ph.D. students and over 100 Masters students, and publishing well over 300 refereed publications (I omit selecting references).

We are fortunate that the staff of all three projects published generously: we know a lot about their inventions and engineering approaches. It is clear that although they pushed machine learning as far as they could, a large fraction of their effort was devoted to accumulating vast databases of problem-specific data, and not simply "training data" for classifiers but operational, data-flow, and financial information provided by their partner postal services or acquired painstakingly by the document recognition technical staff in numerous site visits.

The models driving these systems were plainly extremely costly to acquire, and also extremely strong since they were made up of millions of application-specific details. The

[1] The terms "tight" and "loose" might have been more intuitive, and less normative.

range of models they used was also unprecedentedly wide, including all of the types I itemized above, including image quality and layout models for perhaps the first time; one could argue that in applying zipcode/state/city/street/number constraints they were using semantic models. By all accounts the combination of many strong models was essential to success.

There is discussion in their literature of methods for detecting special cases and then applying pre-trained models particular to them: an early example of adaptation to the input. But I do not find that models themselves were ever adapted on-the-fly: they remained essentially constant. I suspect that this is a consequence of the fact that the inputs—single address images—were individually quite short, even though the sequence of inputs was of course long and highly variable.

B. Reading Chess Books

In another project depending crucially on strong models, but at an opposite extreme in scale, in 1989 two Bell Labs researchers built, in a few weeks, a custom system [10] to read several volumes in the Chess Informant series [11] which contain descriptions of chess games selected (and meticulously proofread) by chess masters and published by the International Chess Federation (FIDA).

My partner in this project was Ken Thompson, co-inventor of Belle [12], the first chess-playing machine to earn an international Master rating. He brought his existing algorithms for checking the legality of chess moves, and specially developed code to parse the regular-grammar syntax of FIDA's game descriptions. I contributed my experimental multilingual page reader, which was already able to analyze the single-column page layouts and which I quickly trained on the non-ASCII characters of the (single) FIDA chess typeface. We hand-crafted layout rules to extract game descriptions from sequences of pages.

The books were poorly machine-printed (letterpress on soft paper) which challenged the then state-of-the-art of layout analysis and character and recognition and segmentation. The results of recognition, including alternative interpretation of characters, were analyzed with the help of our layout and syntactic models, and thus encoded games—sequences of chess moves—were extracted. By applying knowledge of the rules of chess (our "semantic" model), each move was checked for legality directly in the context built up by prior moves and indirectly through the consistency of later moves. If after all these checks an interpretation remain illegal, legal alternatives (that did not occur to OCR) were generated, again by invoking semantic rules. The semantic analysis was fully backtracking and unprotected against exponential explosion, which in fact occurred but fortunately only rarely (in which case we manually shut down and restarted on the next game).

Of the games with no typographical errors, 97% were assigned a complete legal interpretation (later verified manually) for an effective success rate of 99.995% of characters correct on approximately one million characters (2850 games, 945 pages, four volumes). The character error rate due to iconic (and segmentation) models was about 0.5%; the syntactic model cut this only in half (to 0.2%); finally, the semantic model drove it down to 0.005%: fifty errors per million. The success rate on *complete games* was 42% after the iconic model was applied, 76% after the syntactic model, and 97% after the semantic model. Ken felt that correctly reading 97% of the games fully justified his effort, and he put many more volumes through the system and added their hundreds of games, not otherwise available in machine-readble form, to Belle's "opening book."

Of course Ken's syntactic and semantic models were strongly specialized to these books, and required highly skilled engineering that would rarely be available or affordable; my iconic model was also strong (and reasonably accurate) and required a few days of skilled semi-manual training; but my layout and segmentation models were weak and generic (and yet worked fine without modification).

Prior to this, OCR linguistic models had been applied to short passages, often a single word at a time. The tractability, compared to natural language, of the syntax and semantics of chess offered us an early opportunity to assess the value of high-level models on long passages, here of the hundreds of characters making up each complete game. The fact that this semantic model operated on long passages struck me as the essential reason for the astonishing 100-fold reduction in character error rate compared to iconic recognition alone. This never-forgotten lesson would bear different, and less narrowly specialized, fruit twenty years later in whole-book recognition research with at Lehigh University with Pingping Xiu.

V. Generative Models

Pattern recognition professionals often suspect that the classifier trained on the most data wins [13], [14], [15]. A serious chronic obstacle to the broad application of machine learning remains the difficulty or impossibility of collecting and labeling large representative data sets of real data to drive training. An appealing circumvention is to complement real data (or supply their total lack) with synthetic data using generative models.

Among the earliest of these were hand-crafted parameterized models of bilevel and grey-level degradations of images of text (characters, then words, and finally whole pages) and their implementation as pseudorandom generators. I first heard this proposed by Theo Pavlidis in about 1984, and by the late 80's I had convinced myself, in experiments with a ten-parameter model [16], that it offered synthetic data of quality good enough to systematically expand the range of cases subject to automatic training. This enabled four engineers (Theo, Simon Kahan, Reid Fossey, and myself) to build a high-performance 100-font page reader in a few months [17][18].

A 100-font classifier—trained on many fonts and ready to attempt recognition on any of them, even when the fonts are mixed together— is admirably versatile, but if the input happens to be in only a single font, I would say this system is relying on a weak model. George Nagy and I trained 100 distinct classifiers[19], each on a different single font: they had strong models, of course, when run on text only in their custom font. We then tested the 101 classifiers: the 100-font classifier, run on character-images from all 100 fonts, had an average error rate of 4.2%; when each single-font classifier was run on its *own* font, the error rate (averaged over all 100 tests) fell to 0.81%, an improvement of a factor of 5.2. Naturally we had expected specialized systems to excel on their specialities, but I was amazed at the large improvement: in those days (perhaps still today) an OCR engineer would be proud of an improvement of a factor of only 1.1 (a 10% absolute improvement). This was another striking revelation of the power of strong models.

An important open problem in the 1990s, especially for degradation models, was how to assess their accuracy (their "realism"). Tapas Kanungo [20] made important progress in his dissertation under Robert Haralick with a well-founded statistical method for evaluating such models using bootstrap tests on sets of image samples. He applied it to find values of the free parameters of such a model to best match a collection of real images. Unexpectedly, his test was also exquisitely sensitive, capable of discriminations finer than untrained human eyes can manage [21].

Li *et al* [22] validated image degradation models using knowledge of OCR errors, but depended on ground-truth being known.

Elisa Barney Smith *et al* investigated methods [23], [24], [25] for estimating the parameters (including blurring and thresholding) of image-quality models from real images of known characters. Aside from her work, as far as I know, no model-validation test has yet been used to compare competing image quality models with collections of real text images to determine which is the best: this remains a long-standing open question.

A. When is Synthetic Data Safe?

Synthetic training data is now used in our community widely and almost routinely. But unrepresentative data, even data that is only a little too degraded, can yield inaccurate classifiers, so it is reasonable to ask when it is safe. Jean Nonnemaker [26] has pointed out that synthesis can be carried out in at least three natural "spaces" of images: *sample* space (the set of all real samples as they are originally described), *parameter* (or, generator) space (vectors of values determining how samples are generated), and finally, *feature* space (vectors by numerical feature values). Synthetic data can be (and has been) generated in all three spaces: in sample space by combining parts of samples or averaging; in parameter space by varying the parameters that control generation; and in feature space by perturbing feature values. In her dissertation she explored the parameter spaces of two generative models: an image degradation model [16]; and a parameterized model of typeface design (adapted from Knuth's Metafont system). She showed how to generate previously unknown (but still legible) typefaces by interpolation between well-known typefaces in the Metafont parameter space, and then, within these synthetic typefaces, degraded character images were generated, also by interpolation between legible images, in the image-degradation parameter space.

In this rich two-tier synthesis, of both typefaces and character images, Nonnemaker found, in large-scale systematic tests, that training on synthetic data that results from interpolation between or among real images in parameter spaces is safe: that is, classifiers trained on such synthetic samples never performed worse on real samples. Furthermore, classifiers trained on synthetic data often improved (about one third of the time) in accuracy on never-before-seen synthetic cases: that is, such training was effective in extending the domain of competence of the classifiers to include cases for which real typefaces and real images would be unavailable. To summarize: training on synthetic data which is generated by interpolation (between and among) real samples is both safe and effective, and over both typeface and image quality variations.

More work of this sort is clearly needed, in order to supply a firmer foundation for the safe use of data generated by models.

B. Synthetic Data & Human Interactive Proofs

Methods for generating synthetic images of text across a wide range of image defects has found a serendipitous use. In 1997 Andrei Broder and his colleagues [27], then at the DEC Systems Research Center, developed a scheme to block the abusive automatic submission of URLs to the AltaVista web-site. Their approach was to present a potential user with an image of printed text formed specially so that machine vision (OCR) systems could not read it but humans could. In September 2000, Udi Manber, Chief Scientist at Yahoo!, challenged Prof. Manuel Blum and his students [28] at The School for Computer Science at Carnegie Mellon University to design an "easy to use reverse Turing test" that would block 'bots' (computer programs) from registering for services including chat rooms, mail, briefcases, etc. In January 2002, Prof. Blum and I ran, at PARC, the first workshop on 'human interactive proofs' (HIPs), defined as "challenge/response protocols which allow a human to be authenticated as a member of a given group: as human (vs. machine), as a particular individual (vs. everyone else), as an adult (vs. a child), etc."

Richard Fateman and I built the PessimalPrint [29] HIP using an image degradation model. Within its generator parameter space, it is possible to locate the margins of good performance of any OCR system by systematic testing on synthetic data. Psychophysical testing on human subjects similarly maps their capabilities. Wherever this reveals a gap in ability—where OCR systems fail but human readers succeed—an unbounded series of word images can then be generated to serve as HIPs.

All large-scale commercial uses of HIPs still exploit the gap in ability between human and machine vision systems in reading images of text, and synthetic images containing pseudorandomly generated defects (of a wide variety) are key to many of them.

Systematic testing using generative models have the potential, still mostly untapped, to reveal the margins of good performance of many document recognition systems. Knowing exactly where weaknesses lie hidden is surely a good thing: for example, it may be possible to repair weaknesses by generating training data in and around the detected margins of failure; a stab at this by Tin Kam Ho and myself [30] was too brief to be conclusive: it remains a promising open problem.

VI. Automatically Adaptive Models

Research into document image analysis over the last two decades has demonstrated that automatically adaptive recognition algorithms can, in some circumstances, improve accuracy substantially without human intervention.

Tao Hong [31] showed that in books printed in a single typeface, an adaptation strategy that alternates between applying "visual constraints" and "linguistic constraints", can reduce errors. This was perhaps the first adaptive recognition technique to exploit both iconic and linguistic models on a roughly equal basis, and to operate on long (multi-page) passages: my imagination was strongly excited by the thought of alternating between two distinct models (iconic and linguistic) within long passages; this memory was one of the driver's for Xiu's and my recent work on whole-book recognition.

Nagy, Shelton, and I [32][19] showed that a character classifier trained on many typefaces can "self-correct" when adapting to text in a single unnamed typeface. The method was risky: run a classifier known to be imperfect but then *accept its classifications as true* and, using them, retrain. I confess it felt also sinful, since training on poisoned data violated core precepts of supervised classification. So we were astonished at how safe and effective it turned out to be: experiments on 6.4 million pseudo-randomly distorted character images showed improvement on 95 out of 100 typefaces. Character error rate fell by a factor of 2.5, averaged over all 100 typefaces on an alphabet of 80 ASCII characters at body-text size and quality. The self-correcting method complements, and does not hinder, other methods for improving accuracy including the application of linguistic models.

While these results were exciting and suggestive, they were merely empirical, lacking any analytical insight into the causes of the improvements.

A. Style-Conscious Recognition

Prateek Sarkar's dissertation [33] gave the first rigorous analysis of classifiers able to exploit uniformity in the input to enhance recognition. Sarkar's theoretical framework is admirably clear and compelling and his methods are potentially widely applicable (far beyond document image recognition). Sarkar, Nagy, and Veeramachaneni [34], [35], [36] went on to investigate a family of "style-conscious" algorithms able to improve recognition on "isogenous" documents—that is, documents produced in a uniform manner so that they contain only a few of the many typefaces, image qualities, and other variations that can occur in diverse collections. They also showed (to my surprise) that it is not necessary to train on styles in order to reap some (but typically slight) benefit.

Isogeny and exploitable uniformities turn out to be widespread and highly varied, potentially embracing not only typefaces, handwriting styles, and image qualities, but languages and domains of discourse.

Style consistency has usually been applied to "fields" of up to a dozen or so characters, but on longer passages it performs nearly identically to simple style-first recognition [37].

A key challenge of adaptive recognition remained: how to proceed when it is known (or presumed) that the input document is isogenous, but the particular models which generated it are not known.

VII. Joint Recognition over Multiple Models

If we can't manage to craft (or train) a single complete, near-perfect "strong" model to drive recognition, can we make progress by combining several imperfect or incomplete "weak" models?

A. Document Image Decoding

PARC's Document Image Decoding (DID) technology finds a sequence of characters that best explains an observed document image in terms of models of printing, scanning, and language. It is based on a communication-systems interpretation that views the generation of a printed page in terms of transitioning through a probabilistic finite-state machine, or Markov model. In this iconic model of the image source, a mark or template (typically corresponding to an individual character) is printed upon every state transition, and the current printing location on the page is then advanced by the corresponding 'set-width' (actually, a 2-D displacement vector; thus the iconic and segmentation models are integrated). The observed image is viewed as a possibly degraded version of the ideal image produced by the Markov model.

The task performed in recognition is to work backwards from the observed image to reason about what path must have been followed through the Markov source, and what the degradation must have been, and the language, to produce that image. A variant of dynamic programming was used to accomplish this in the original work on DID [38], under the assumption that the Markov source is amenable to causal processing via scheduling. In later work ([39], [40], [41], [42], [43], [44], [45], [46], [47], [48]), this assumption was maintained but its significance somewhat obviated by an independent restriction to individual text lines which are one-dimensional and hence trivially schedulable: this line-by-line decoder was fully implemented.

Several models must be specified for DID to work on any particular document. Three of these are learned from training data: (1) *iconic*, the shapes and identities of the character templates printed by the Markov source, (2) *image quality*, the manner in which the ideal image is corrupted on the way to observation, and (3) *linguistic*, a prior description of which recognized strings are valid in the given language. One remaining DID model is structural: these are the states and transitions in the Markov model itself; in the special case of an isolated text-line decoder this is a simple, fixed three-state Markov model (in omore complicated cases, this model would be specified manually).

Given training images, learning the iconic model becomes essentially a character segmentation problem, which was addressed by a graph-theoretic independent-set formulation [40], an iterative procedure [41], and finally as an instance of the expectation-maximization algorithm [41] [42].

The second modeling problem, learning image quality (or, degradation parameters) was initially addressed using a binary asymmetric pixel-flip model [38], then generalized to allow different pixel-flip probabilities depending on position within the template [43], and finally to grayscale observations with the possibility of spatial dependence in the deterministic component of the degradation [39].

For the third class of models needed by DID, the linguistic model, a unigram (simple letter frequency) model was initially used [38]; this was then extended to fixed [44] and variable [45] character n-grams (statistical characterization in terms of groups of n adjacent characters). When integrating an n-gram language model into Viterbi template-matching search, conventional dynamic programming is no longer feasible due to a potentially exponential blow-up in the search space. In response to this problem PARC researchers invented a procedure (iterated complete path optimization) to find an optimal path that is, on average, far faster [45] and compared it with an approximate (sub-optimal) technique that is often faster still [46].

Large reductions (up to a factor of forty) in runtime were thus realized without compromising optimality by iterative decoding using heuristic upper bounds and segmental dynamic programming [47], and judicious subsampling [48].

DID research first emphasized retargeting: that is, supervised training of decoders, requiring manual effort to prepare training images and synchronized ground truth [41]. Later, PARC reduced the manual effort of DID training significantly by obviating manual pre-segmentation of images of text lines into words or characters during both training and testing [42]. They also showed that DID decoders are trainable to high accuracy across a wide range of explicitly parameterized image degradations [49].

Some DID models, including the iconic model are, in our sense, necessarily strong, but can be learned automatically. Other models are arguably quite weak: the character n-gram models are seldom perfect for any given input; and the bit-flip image-quality model is simplistic; but these also can be inferred automatically from corpora. Therefore it interested me that when this mix of strong and weak models were optimized jointly during recognition, accuracy improved dramatically [44], [49]; this may be connected to the remarkable fact that joint recognition in DID was provably optimal in a precise and realistic sense. It was also striking that joint optimization over multiple models could be carried out rapidly. A fascinating set of open problems is suggested by the evolution of the DID technology. Each extension, from one to two to three models, and to linguistic models of increasing complexity, has maintained provable optimality and high speed, while at each step achieved significantly higher accuracy. This consistent improvement seems to me to result largely from the invention of the iterated complete path algorithm: why stop at three models?—I suspect that it can be extended to higher orders of linguistic models such as word n-gram occurrence models, and—who knows—perhaps pragmatic and semantic models. This is deserves several Ph.D. dissertations.

A general characteristic of the document image recognition literature on joint optimization over multiple models is its restriction to extremely short passages: usually, isolated characters; occasionally, isolated words; and only rarely passages longer than a single word. While DID technology successfully operated on images of entire text lines, the text lines were processed independently (starting afresh on each) for the most part, and never across page boundaries.

B. *Deciphering OCR*

The literature on “deciphering” OCR has shown repeatedly [50][51][52][53][54][55] that clustering character images (in which the iconic model is weak, merely an image-similarity metric), together with pruning by linguistic constraints, can improve recognition. While deciphering-OCR algorithms operate on long passages and use strong linguistic models, they have not, as far as I can tell, adapted their linguistic models on the fly.

VIII. AUTOMATICALLY CHOOSING A STRONG MODEL

Can a recognizer automatically pick or construct a strong model of its input? I'll describe two approaches: if we pre-train models for each style of inputs that might be seen (taken together this makes for a weak initial model), it can be possible to choose the strongest model for any given input; and in a system driven by several weak models, they can be made to correct one another and so strengthen all of them.

A. Selecting Among Pre-trained Strong Models

Prateek Sarkar, Dan Lopresti, and myself investigated high-accuracy, fully automatic recognition of machine printed text across a wide range of challenging image qualities, without requiring manual intervention or supervised training [56]. This approach was made possible by two properties of the DID technology: (1) it is trainable for high accuracy across a wide range of explicitly parameterized image degradations; and (2) decoders for arbitrary parameter settings can thus be generated automatically. Large-scale experiments on synthetic images showed that, when many pre-trained decoders are applied in parallel to an input image of unknown but fixed image quality, the decoder that yields the highest accuracy is often the one that exhibits the highest DID *posterior* 'Viterbi score' which is computed as a side-effect of the decoding process. By choosing this decoder, the need for manual document–specific training is eliminated with little or no loss in accuracy. When implemented naively, in a brute–force manner, decoder banks can be computationally intensive: it is an open question how their cost may be reduced with no loss of versatility, automation, or accuracy.

B. Mutually Correcting Models

Can a recognition algorithm detect that it is operating on a type of document different from those that its models were trained on? That is, can it judge the strength of its models without knowing ground-truth? Can an unsupervised algorithm which changes its models on the fly know whether those changes will be advantageous or not?

Pingping Xiu and I, believing that model adaptation operating on long passages might shed light on these questions, explored "whole-book" recognition [57], a strategy that operates on the complete set of a book's page images using automatic adaptation to improve accuracy.

Nagy and Sarkar's work sensitized us to the implications of the fact that, within long isogenous books, identical (or highly similar) character images occur multiple times and therefore in a variety of word contexts. Thus, a defect in the iconic model (say, the wrong shape for a particular character) can cause multiple errors on instances of this character, and so damage the recognition of more than one word. Similarly, defects in the linguistic model (say, a word not in the lexicon but occurring in the document) can damage more than one character's interpretations. Models' errors affect one another: can these effects be separated?

Let us assume that the models which implicitly generated all the images of an isogenous book include at least (1) an iconic (character-image formation) model and (2) a linguistic (word-occurrence) model and that recognition is performed jointly with respect to both models. Then we expect that most errors will be due to imperfections in one or both of the models. If a particular error is caused by one model but not the other, then it may provide evidence of a "disagreement" between the models.

We discovered that such disagreements are real and can be detected automatically (using cross entropy [58])—further, these disagreements, when summed over long passages (of many pages), correlate significantly with character and word error rates. Thus disagreement, a statistic which the algorithm can estimate, turns out to be a reliable proxy for error rate, which in an unsupervised setting is of course unavailable to the algorithm. Furthermore disagreements can help identify candidates for model corrections at both the character and word levels [59]. Some model corrections will reduce the error rate over the whole book (while others won't), and these successful corrections can be identified with a useful degree of confidence by comparing model disagreements, summed across the whole book, before and after the correction is applied. If implemented naively, the algorithm runs in time quadratic in the length of the book; but random subsampling and caching techniques speed it up by two orders of magnitude with negligible loss of accuracy [60]. The longer the passage operated on by the algorithm, the more reliable this adaptation policy becomes, and the lower the error rate achieved. Experiments on passages up to one hundred and eighty pages long show that when a candidate model adaptation reduces whole-book disagreement, it is also likely to correct recognition errors [61]. error rates are driven down by nearly an order of magnitude fully automatically without supervision (or indeed absent any user intervention or interaction). Improvement is nearly monotonic, and asymptotic accuracy is stable, even over extremely long runs. Best results occur when the iconic and linguistic models mutually correct one another [62]: flawed critics locked in a virtuous embrace.

The method requires little application-specific engineering: it expects to be initialized with approximate iconic and linguistic models—derived from (generally errorful) OCR results and (generally imperfect) dictionaries—and then, guided entirely by evidence internal to its evolving test set, corrects the models which, in turn, yields higher recognition accuracy. Whole-book recognition has potential applications in digital libraries as a safe unsupervised anytime algorithm.

It is an open question whether mutual correction can be extended to more than two models: it's technically challenging to find statistics which allow criticism among models in an even-handed manner, without favoring (trusting the accuracy of) one model over the others. If however this is possible—if, say, image quality, character n-gram, or layout models can be added—there are good reasons, as we have noted earlier, to expect that corrections will be more reliable and improvement greater [63].

IX. Conclusion

Drawing on over twenty-five years of research by dozens of researchers, I have pieced together the case that high-performance document image recognition systems can be built without detailed knowledge of the application, starting with models to drive recognition which are what I have termed "weak" (incomplete or generic) but which are often easy to acquire. The distinction between weak and strong models, which I use to unify my argument, shifts attention away from our customary concerns with engineering cost, accuracy, and speed, and towards the degree of "fit" of a variety of models (on which cost, accuracy, and speed depend) to the particular input the system is attempting to recognize.

Statistical machine learning allows us to train models from real labeled-sample data which, if they are too few, can be complemented or even completely supplied by samples synthesized by hand-crafted and statistically calibrated generative models. Training on synthetic data appears to be safe and effective when it is carried out by interpolation (in generator parameter space) between or among real data samples. In addition, synthetic data can map the margins of good performance and perhaps guide automatic repair of failure regions. If we can't afford to acquire complete application-specific "strong" models, we can make significant progress by combining several weak models, the more the better. Joint recognition over weak models can be proved optimal and still run fast. A recognizer can automatically pick a strong model of its input and thus adapt to it and improve, without supervision. We do not always need to pre-train models for every style of input the system may encounter: it can adapt to unknown styles. In this way models that start out weak can grow stronger and so drive accuracy higher, without any human intervention. In a system driven by more than one initially weak model, each of the models can criticize and correct the other models—without knowledge of ground truth—even while it is being criticized and corrected by them at the same time; for this to work, it seems to be necessary to operate over long passages repeatedly within the inner-loop of the recognition process. Improvements by mutual correction can be nearly monotonic and highly stable over extremely long runs, and so they support anytime recognition systems that can reliably run unattended.

Taken together, these research results suggest a model-intensive engineering methodology: start with manageably small collections of real training data, amplify them with synthetic data using generative models, train weak models, jointly optimize recognition over several weak models, and adapt (and so strengthen) models by style-conscious recognition applied to short passages and mutual correction over long passages. In this way, high-performance document image recognition systems can be built with only small up-front investments in application-specific modeling.

Such engineering techniques promise to extend the range of affordable and highly accurate document recognition systems.

Acknowledgment

I would like to express my gratitude to the many researchers whose ideas influenced me, and most keenly to those who agreed to collaborate with me, especially: Theo Pavlidis, Tin Kam Ho, David Ittner, Ken Thompson, George Nagy, Robert Haralick, Tao Hong, Tapas Kanungo, Phil Chou, Daniel Lopresti, Gary Kopec, Dan Bloomberg, Ashok (Kris) Popat, Thomas Breuel, Elisa H. Barney Smith, Prateek Sarkar, Sriharsha Veeramachaneni, Jean Nonnemaker, and Pingping Xiu. I am sensible of the rare honor granted me by the organizers of ICDAR 2011 in inviting me to deliver a keynote address: this is a once-in-a-lifetime chance to open my mind without restraint to my research community on matters which have absorbed me—I should admit, obsessed me—through most of my career. I would like also to extend thanks, in advance, to the people who will hear the address and read this companion paper (for which the talk's slides provide Figures and illustrations), for their forbearance with a train of thought which, while perfectly natural to my way of thinking, may, I can easily imagine, seem arbitrary and idiosyncratic, even eccentric, to others'.

References

[1] T. Pavlidis, "Thirty-six years at the pattern recognition front," Barcelona, SPAIN, September 2000, King-Sun Fu Prize Lecture delivered at the 11th ICPR (`www.theopavlidis.com/technology/KSFuLecture.htm`).

[2] T. M. Mitchell, *Machine learning*. New York: McGraw Hill, 1997.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, 2nd Edition*. New York: Wiley, 2001.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, New Jersey: Springer-Verlag New York, Inc., 2006.

[5] M. Decerbo, P. Natarajan, R. Prasad, E. MacRostie, and A. Ravindran, "Performance improvements to the BBN Byblos OCR system," in *Eighth Int'l Conf. on Document Analysis and Recognition (ICDAR'05)*, vol. 1, 2005, pp. 411–415.

[6] "The OCRopus(tm) open source document analysis and OCR system," October 2007, the alpha release. [Online]. Available: http://code.google.com/p/ocropus/

[7] J. Weinman, E. Learned-Miller, and A. McCallum, "Fast lexicon-based scene text recognition with sparse belief propagation," in *Proceedings, IAPR 9th Int'l Conf. on Document Analysis and Recognition (ICDAR'07)*, Curitiba, BRAZIL, September 2007.

[8] A. Susuki and S. Miyahara, “Word recognition coping with undefined words,” *Institute of Electronics, Information, and Communication Engineers*, vol. J76-D-II, no. 3, pp. 464–473, March 1993.

[9] T. Hamamura, T. Akagi, and B. Irie, “An analytic word recognition algorithm using a posteriori probability,” in *Proceedings, IAPR 9th Int’l Conf. on Document Analysis and Recognition (ICDAR’07)*, Curitiba, BRAZIL, September 2007.

[10] H. S. Baird and K. Thompson, “Reading chess,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI–12, no. 6, pp. 552–559, June 1990.

[11] *Chess Informant (Sahovski Informator)*. International Chess Federation (FIDA).

[12] J. H. Condon and K. Thompson, “Belle,” in *Chess Skill in Man and Machine*, P. Frey, Ed. Springer-Verlag, 1982.

[13] T. K. Ho and H. S. Baird, “Large-scale simulation studies in image pattern recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1067–1079, 1997.

[14] P. Simard, Y. A. Le Cun, J. S. Denker, and B. Victorri, “Transformation invariance in pattern recognition - tangent distance and tangent propagation,” in *Neural Networks: Tricks of the Trade*, G. B. Miller and K.-R. Orr, Eds. Springer, 1998, vol. Chapter 12.

[15] T. Varga and H. Bunke, “Comparing natural and synthetic training data for off-line cursive handwriting recognition,” in *9th Int’l Workshop on Frontiers in Handwriting Recognition*, IEEE, Ed., 2004.

[16] H. S. Baird, “Document image defect models,” in *Structured Document Image Analysis*, H. S. Baird, H. Bunke, and K. Yamamoto, Eds. New York City, New York: Springer-Verlag, 1992, pp. 546–556.

[17] S. Kahan, T. Pavlidis, and H. S. Baird, “On the recognition of printed characters of any font and size,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI–9, no. 2, pp. 274–288, March 1987.

[18] H. S. Baird and R. Fossey, “A 100-font classifier,” in *Proc., 1st Int’l Conf. on Document Analysis & Recognition (ICDAR’91)*, St.-Malo, FRANCE, September 1991.

[19] G. Nagy and H. S. Baird, “A self-correcting 100-font classifier,” in *Proc., IS&T/SPIE Symp. on Electronic Imaging: Science & Technology*, San Jose, California, February 1994.

[20] T. Kanungo, “Document degradation models and a methodology for degradation model validation,” Ph.D. dissertation, University of Washington, Seattle, 1996, (Advisor: R. Haralick).

[21] H. S. Baird, “Document image quality: Making fine discriminations,” in *Proc., IAPR 5th Int’l Conf. on Document Analysis and Recognition (ICDAR1999)*, Bangalore, INDIA, September 1999.

[22] Y. Li, D. Lopresti, G. Nagy, and A. Tomkins, “Validation of image defect models for optical character recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 99–108, 1996.

[23] E. H. B. Smith, “Characterization of image degradation caused by scanning,” *Pattern Recognition Letters*, vol. 19, no. 13.

[24] E. H. B. Smith and X. Qiu, “Relating statistical image differences and degradation features,” in *Proc., 5th IAPR Int’l Workshop on Document Analysis Systems*. Princeton, New Jersey: Springer-Verlag, August 2002, pp. 1–12, lNCS 2423.

[25] ——, “Statistical image differences, degradation features and character distance metrics,” *International Journal of Document Analysis and Recognition*, vol. 6, no. 3.

[26] J. Nonnemaker, “The safe use of synthetic data in classification,” Ph.D. dissertation, Lehigh University, December 2008, (Advisor: H. S. Baird).

[27] M. D. Lillibridge, M. Abadi, K. Bharat, and A. Z. Broder, “Method for selectively restricting access to computer systems,” U.S. Patent No. 6,195,698, Issued February 27, 2001.

[28] M. Blum, L. A. von Ahn, and J. Langford, “*The CAPTCHA Project:*, Completely Automatic Public Turing test to tell Computers and Humans Apart,,” Dept. of Computer Science, Carnegie-Mellon University, `www.captcha.net`.

[29] A. L. Coates, H. S. Baird, and R. Fateman, “Pessimal print: a reverse turing test,” in *Proc., IAPR 6th Intl. Conf. on Document Analysis and Recognition*, Seattle, Washington, 2001, pp. 1154–1158.

[30] T. K. Ho and H. S. Baird, “Large-scale simulation studies in image pattern recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1067–1079, October 1997.

[31] T. Hong, “Degraded text recognition using visual and linguistic context,” Ph.D. dissertation, State University of New York at Buffalo, 1995, (Advisor: J. Hull).

[32] G. Nagy and G. L. Shelton, “Self-corrective character recognition system,” *IEEE Trans. on Information Theory*, vol. IT-12, no. 2, pp. 215–222, April 1966.

[33] P. Sarkar, “Style consistency in pattern fields,” Ph.D. dissertation, Rensselaer Polytechnic Institute, May 2000, (Advisor: G. Nagy).

[34] P. Sarkar and G. Nagy, “Style consistent classification of isogenous patterns,” *IEEE Trans. on PAMI*, vol. 27, no. 1, January 2005.

[35] P. Sarkar, “An iterative algorithm for optimal style conscious field classification,” in *Proc., IAPR 16th Int’l Conf. on Pattern Recognition (ICPR2002)*, vol. 4, 2002, pp. 40–43.

[36] S. Veeramachaneni and G. Nagy, “Style context with second order statistics,” *IEEE Trans. on PAMI*, vol. 27, no. 1, January 2005.

[37] ——, “Analytical results on style-constrained bayesian classification of pattern fields,” *IEEE Trans. on PAMI*, vol. 29, no. 7, July 2007.

[38] G. Kopec and P. Chou, “Document image decoding using Markov source models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI–16, pp. 602–617, June 1994.

[39] K. Popat, “Decoding of text lines in grayscale document images,” in *Proc., 2001 Int’l Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2001)*. Salt Lake City, Utah: IEEE, May 2001.

[40] M. Lomelin, “Character template estimation from document images and their transcriptions,” Master’s thesis, MIT, Cambridge, Massachusetts, June 1995, M.S. Thesis.

[41] G. Kopec and M. Lomelin, “Supervised template estimation for document image decoding,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI–19, no. 12, pp. 1313–1324, December 1997.

[42] G. Kopec, “An EM algorithm for character template estimation,” submitted to *IEEE Trans. on PAMI* in March 1997; returned for revision, but never revised due to the author’s death; available from PARC by request.

[43] ——, “Multilevel character templates for document image decoding,” in *Proc. of Document Recognition IV, SPIE vol. 3027*, L. Vincent and J. Hull, Eds., 1997.

[44] G. Kopec, M. Said, and K. Popat, “N-gram language models for document image decoding,” in *IS&T/SPIE Electronic Imaging 2002 Proc. of Document Recognition and Retrieval IV*, San Jose, California, January 2002.

[45] K. Popat, D. Bloomberg, and D. Greene, “Adding linguistic constraints to document image decoding,” in *Proc., 4th Int’l Workshop on Document Analysis Systems*. Rio de Janeiro, Brazil: International Association of Pattern Recognition, December 2000.

[46] K. Popat, D. Greene, J. Romberg, and D. S. Bloomberg, “Adding linguistic constraints to document image decoding: Comparing the iterated complete path and stack algorithms,” in *Proc. of IS&T/SPIE Electronic Imaging 2001: Document Recognition and Retrieval VIII*, San Jose, California, January 2001.

[47] T. P. Minka, D. S. Bloomberg, and K. Popat, “Document image decoding using iterated complete path heuristic,” in *Proc. of IS&T/SPIE Electronic Imaging 2001: Document Recognition and Retrieval VIII*, San Jose, California, January 2001.

[48] D. Bloomberg, T. Minka, and K. Popat, “Document image decoding using iterated complete path search with subsampled heuristic scoring,” in *Proc., IAPR 2001 Int’l Conf. Document Analysis and Recognition (ICDAR 2001)*, Seattle, Washington, September 2001.

[49] P. Sarkar, H. S. Baird, and X. Zhang, “Training on severely degraded text–line images,” in *IAPR 7th Int’l Conf. on Document Analysis and Recognition (ICDAR03)*, Edinburgh, SCOTLAND, August 2003.

[50] G. Nagy, S. Seth, and K. Einspahr, “Decoding substitution ciphers by means of word matching with application to OCR,” *IEEE Trans. on PAMI*, vol. 9, no. 5, pp. 710–715, September 1987.

[51] C. Fang and J. J. Hull, “A modified character-level deciphering algorithm for OCR in degraded documents,” in *Proc. of SPIE/IS&T Conf. on Document Recognition II*, February 1995, pp. 76–83.

[52] T. K. Ho and G. Nagy, “OCR with no shape training,” in *Proc., IAPR 15th Int’l Conf. on Pattern Recognition (ICPR2000)*, vol. 4, 2000, p. 4027.

[53] C. Fang, “Deciphering algorithms for degraded document recognition,” Ph.D. dissertation, State University of New York at Buffalo, 1997, (Advisor: S. N. Srihari).

[54] S. Leishman, “Shape-free statistical information in optical character recognition,” Master’s thesis, Computer Science, University of Toronto, 2007.

[55] G. Huang, E. Learned-Miller, and A. McCallum, “Cryptogram decoding for OCR using numerization strings,” in *Proceedings, IAPR 9th Int’l Conf. on Document Analysis and Recognition (ICDAR’07)*, Curitiba, BRAZIL, September 2007.

[56] P. Sarkar and H. S. Baird, “Decoder banks: Versatility, automation, and high accuracy without supervised training,” in *Proc., IAPR 17th Int’l Conf. on Pattern Recognition (ICPR2004)*, Cambridge, UNITED KINGDOM, August 2004, pp. 646–649.

[57] P. Xiu, “Whole-book recognition,” Ph.D. dissertation, Lehigh University, December 2010, (Advisor: H. S. Baird).

[58] P. Xiu and H. S. Baird, “Whole-book recognition using mutual-entropy-based model adaptation,” in *Proc., IS&T/SPIE Document Recognition & Retrieval XII Conf.*, San Jose, California, January 2008.

[59] ——, “Towards whole-book recognition,” in *Proceedings., 8th IAPR Document Analysis Workshop (DAS’08)*, Nara, Japan, September 2008.

[60] ——, “Scaling-up whole-book recognition,” in *Proceedings, IAPR 10th Int’l Conf. on Document Analysis and Recognition (ICDAR’09)*, Barcelona, SPAIN, July 2009.

[61] ——, “Analysis of whole-book recognition,” in *Proceedings., 9th IAPR Document Analysis Workshop (DAS’10)*, Boston, Massachusetts, June 2010.

[62] ——, “Incorporating linguistic model adaptation into whole-book recognition,” in *IAPR 20th Int’l Conf. on Pattern Recognition (ICPR’10)*, Istanbul, TURKEY, August 2010.

[63] ——, “Multiple-agent adaptation in whole-book recognition,” in *Proc., Document Recognition and Retrieval XVIII Conf. (DR&R XVIII)*, San Francisco, California, January 2011.