Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Large-Scale Simulation Studies in Image Pattern Recognition[1]

Tin Kam Ho, Henry S. Baird
Bell Laboratories, Lucent Technologies

Abstract

Many obstacles to progress in image pattern recognition result from the fact that per-class distributions are often too irregular to be well-approximated by simple analytical functions. Simulation studies offer one way to circumvent these obstacles. We present three closely related studies of machine-printed character recognition that rely on synthetic data generated pseudo-randomly in accordance with an explicit stochastic model of document image degradations. The unusually large scale of experiments — involving several millions of samples — that this methodology makes possible has allowed us to compute sharp estimates of the intrinsic difficulty (Bayes risk) of concrete image recognition problems, as well as the asymptotic accuracy and domain of competency of classifiers.

1 Introduction

In most of the literature on pattern recognition, image data sets, used to train and test classifiers, constitute the *only* specification of the problem offered. These image data are usually unavailable to readers, and have often been gathered unsystematically, without following a published protocol. In such cases, the reported results can not be exactly replicated, and so it is difficult or impossible for readers to generalize from them to new problems without carrying out fresh trials.

This methodological problem has been chronic in the field since its inception, and has repeatedly stimulated the collection and dissemination of public available image data sets of increasing size. However, there are still serious drawbacks in using these databases: the image collections are often unsystematic, not extensible, and, above all, still much too small.

We believe that there are fundamental disadvantages in exclusively relying on *implicit* descriptions of image recognition problems. In this paper, we focus on an explicit, quantitative, stochastic model of degradations that occur in images of documents as a result of the physics of printing and imaging. We attempt to specify a problem precisely using the model, and then study various issues related to the problem. We will emphasize methodological questions, and we will not attempt to validate the particular degradation model used: this important question is being actively researched, and preliminary results are published elsewhere [3] [16].

The first study is an experiment in estimating the Bayes error of a difficult two–class problem. The Bayes error gives the 'intrinsic difficulty' of the problem since it is the minimum error achievable by any classification method. For many realistically complex problems, deriving this analytically appears to be hopeless, so we approach the task empirically.

[1] Parts of this paper have appeared in [12], [13], and [14]

We proceed first by expressing the problem precisely in terms of ideal prototype images and the image defect model, and then by carrying out the estimation on pseudorandomly simulated data. The study of the data reveals many interesting statistics, which allow the prediction of the worst–case time/space requirements for any given classifier performance.

Poor quality — sparse or unrepresentative — training data is widely suspected to be one cause of disappointing accuracy of isolated–character classification in OCR machines.[2] We conjecture that, for many trainable classification techniques, it is the dominant factor affecting accuracy. In the second study, we tested this conjecture by comparing the asymptotic accuracy of three classifiers on the same two–class problem. Using the defect model, the problem is represented as a stochastic source of an indefinitely long sequence of simulated images labeled with ground truth. Using this sequence, we were able to train all three classifiers to high and statistically indistinguishable asymptotic accuracies (99.9%). This result suggests that the quality of training data is the dominant factor affecting accuracy.

In the third study we perform an experiment in which a classifier is evaluated on synthetic character images created using the defect model. The use of synthetic data permits control of the quality of the test data and quantitative analysis of the effects of image quality on OCR accuracy. For that particular classifier, we show that good performance — accuracy above a given threshold — occurs within contiguous regions of the space defined by the range of key parameters of the defect model.

2 A parameterized image defect model

We use a parameterized model of document image defects [2] [4] to describe the problem domain. The model describes effects of: (1) the nominal text size of the output (`size`), (2) the output spatial sampling rate (both horizontally (`xresn`) and vertically (`yresn`)), (3) the point spread function (the standard error of a Gaussian blurring kernel) (`blur`), (4) the digitizing threshold (`thres`), (5) the variation of sensitivity among the pixel sensors (`sens`), (6) the variation of jitter among the pixels (*i.e.* discrepancies of the sensor centers from an ideal square grid, in units of output pixel) (`jitt`), (7) rotation (skew angle) (`skew`), (8) stretching factors (both horizontally (`xscl`) and vertically (`yscl`)), and (9) translation offsets with respect to the pixel grid (`xoff` and `yoff`). Each parameter has a range of values that is determined by the physics of printing and imaging.

Taking an 'ideal' (a black and white image at high digitizing resolution, obtained from bitmaps or scalable outline descriptions purchased from typeface manufacturers) bitmap of a character as input, the generator creates one image when the value of each parameter is fixed and when `sens` and `jitt` are zero. The effects of pixel sensitivity and jitter are randomized per pixel, and the parameters `sens` and `jitt` specify the variance of a normal

[2] Accuracy of reading a complete page image, however, depends heavily on layout analysis and character segmentation, which are topics outside the scope of this paper. Interested readers are referred to [17] and [18] for details of evaluating complete OCR systems.

distribution around zero mean. Multiple images can be created when `sens` or `jitt` is positive. Reasonable lower and upper limits can be set for most parameters including `skew`, `thrs`, `size`, `xresn` and `yresn`. Constraints can also be put on certain parameter values. For instance, we can require `xresn` = `yresn` or `xresn` = 2 × `yresn`. By sampling pseudo-randomly from this multi-variate distribution, we can generate an indefinitely long sequence of distorted images for a given prototype.

This defect model is designed to produce shape distortions similar to those occurring in real-world document images. A rough validation of the model has been carried out in experiments in which a page reader, using a classifier trained only on synthetic data from this model, scored 99.7% accuracy on real images of English books [1]. More thorough validation of such models is currently an active research topic.

3 The Intrinsic Difficulty of a Recognition Problem

The machine vision scientific and engineering communities suffer from a scarcity of performance guarantees. Potential users of the technology can rarely expect definite answers to questions about possible best accuracy and resource requirements both in principle and in practice. The intractability of these questions results from many causes, including the difficulty of unambiguously specifying vision problems, the immature state of methods for estimating the difficulty of problems, and the internal complexity of machine vision systems.

In this section, we describe first steps towards a methodology for answering such questions, and its application to a concrete, realistically complex, practically important problem in character recognition. We hope that this work lays the foundation for a method of automatically constructing classifiers that are guaranteed to achieve any user–specified accuracy, limited only by the problem's intrinsic difficulty and available computational resources.

3.1 Data Generation

The recognition problem is to distinguish images of the symbols 'c' and 'e' in the Adobe[3] Times–Roman typeface (Figure 1) degraded by our model of image defects. The Times–Roman typeface was selected because of its frequent usage in American technical documents [1]. The two symbols were selected because: (1) commercial OCR machines often confuse them ([17] ranks 'e'→'c' and 'c'→'e' as the 2nd and 14th most common mistakes); (2) they are easily distinguishable without noise; and (3) they have identical height, width, and height above baseline, and cannot be easily distinguished by geometrical context. Thus the problem is of practical interest, difficult but not hopeless.

We assume that the two classes are equally probable, and that their images are isolated. These are to simply the experimental design. In a practical OCR environment, the language-

[3]Artwork defining the ideal images of these is available from Adobe Systems, Inc., 1585 Charleston Road, P.O. Box 7900, Mountain View, CA 94039.



Figure 1: Ideal images of 'c' and 'e' in the Adobe Times–Roman typeface.

specific character frequencies can be considered after shape-based classification (in English prose, 'e' and 'c' occur approximately in the ratio 3.7:1).

Text size has a marked effect on OCR accuracy [12]. To study the difficulty of recognition as a function of text size, we created 1,000,000 images (500,000 for each class) at each of six text sizes (5, 6, 7, 8, 9 and 10 point), at a spatial sampling rate of 100 pixels/inch (ppi). Thus they are similar to those occurring in challenging real–world OCR problems, such as those arising in low–resolution FAXes. Figure 2 shows some sample images.

The vast majority (99.98%) of images at this resolution and these text sizes fit within a grid of 9×9 pixels, and thus can be represented by an 81-component binary vector (we center images in the grid). The empirical distribution of samples within this 81-dimensional space is the principal object of our study. We are interested in those characteristics of the distributions that affect both the accuracy and time/space demands of a classifier.

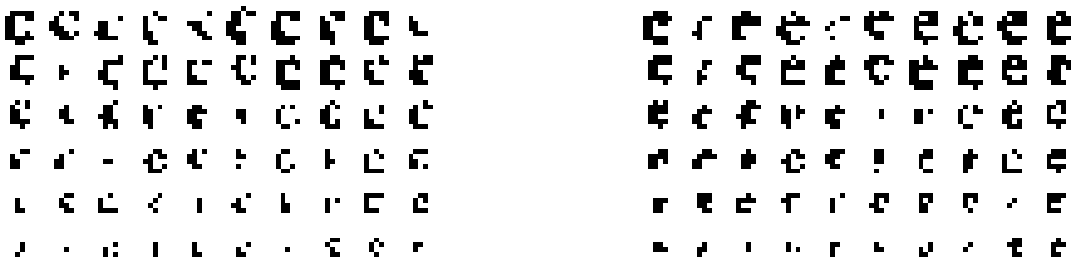


Figure 2: Sample images of c's (left) and e's (right) at 100 ppi and in 10, 9, 8, 7, 6, and 5 point (from top to bottom).

3.2 Characteristics of Sample Distributions

We first investigated the frequency of occurrence of an image in each class. In an 81-dimensional binary space, there are at most $2^{81} \approx 2.4 \times 10^{24}$ distinct points each corresponding to an image. The large volume of the space has led to a belief that classifying by exact match is infeasible due to the hopelessly large number of prototypes needed. Yet not every one of these images has an equal probability of occurrence in each class. If the distributions have a strong central tendency, a limited number of prototypes might be sufficient to represent a large fraction of all images that are likely to occur, so that at least a partial solution to the recognition problem can be obtained within practical limits.

We summarize our findings as follows. The distribution of 5 point images is the most compressed — out of the 500,000 samples of 'c's, only 5,939 distinct images were seen; and of the 500,000 samples of 'e's, only 6,431. On average the ratios of overdetermination are 84 and 78 respectively. Even for the 10 point images, which is the most dispersed, a significant number of repetitions are observed. Out of the 500,000 samples of each class, there are 230,054 distinct 'c's and 287,452 distinct 'e's, and the ratio of overdetermination is 2.17 and 1.74 respectively. Figure 3(a) shows the increase in the number of distinct images as a function of the sample set size.

The occurrence of each distinct image in the sample set is highly nonuniform. Figure 4 shows the cumulative frequencies of the samples, where distinct images are ordered by their frequency of occurrence. For instance, 90% of the 5 point samples are identical to one of less than 4% of the distinct images at that size. This central tendency is stronger for the 'c's than for the 'e's. At 10 point, 60% of the 'c's are included in 18% of the distinct images, but the same number of the 'e's are distributed over 30% of the distinct images.

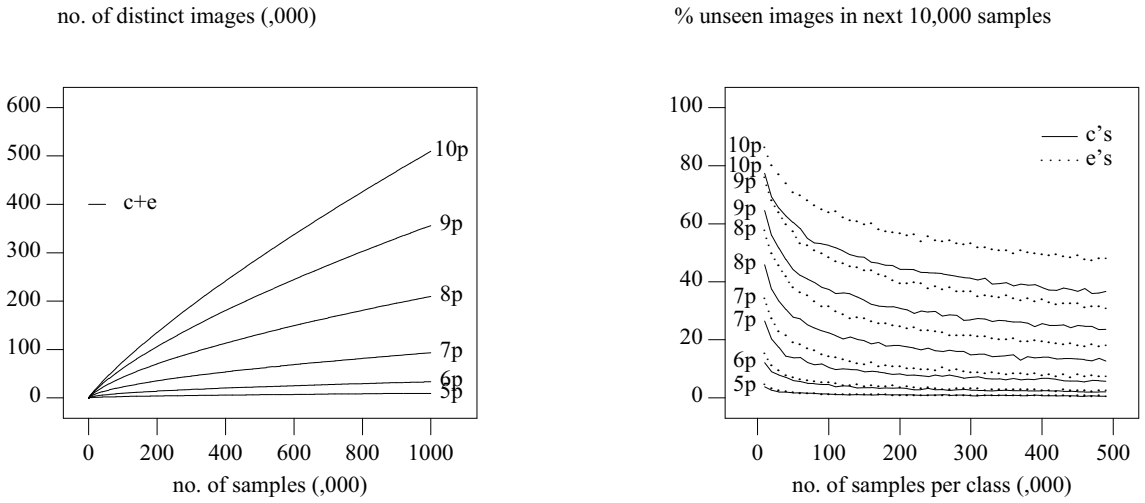
Assuming that the images are generated randomly, we can ask, at each step, 'what is the probability that the next image will be identical to one already seen?' We estimate this probability by a sliding-window method. Starting with an empty set, we add new samples in batches of fixed size (10,000). Before each new batch is added, we determine how many of the new samples have been seen in any of the earlier batches. The frequency of previously unseen images declines quickly as the sample size grows (Figure 3(b)).

To assess the difficulty of discrimination it is important to measure the overlap of the class-conditional distributions. One measure of this is the frequency of occurrence of images shared by both classes. For these images, if frequency information is not used, the class can be decided only at random. Table 1 summarizes the overlap statistics for our entire collection of samples.

Table 1 shows that, at 5 point, the distributions overlap heavily. Identical pairs can be found between 99.0% of the 'c's and 98.5% of the 'e's (lines 7 and 8), and only 1.3% of the samples can be uniquely classified. On the other hand, at 10 point, only 9.1% of all samples (line 6) are potentially confusable. This gives a measure of the difficulty of the problem, and it is clearly affected by the text size.

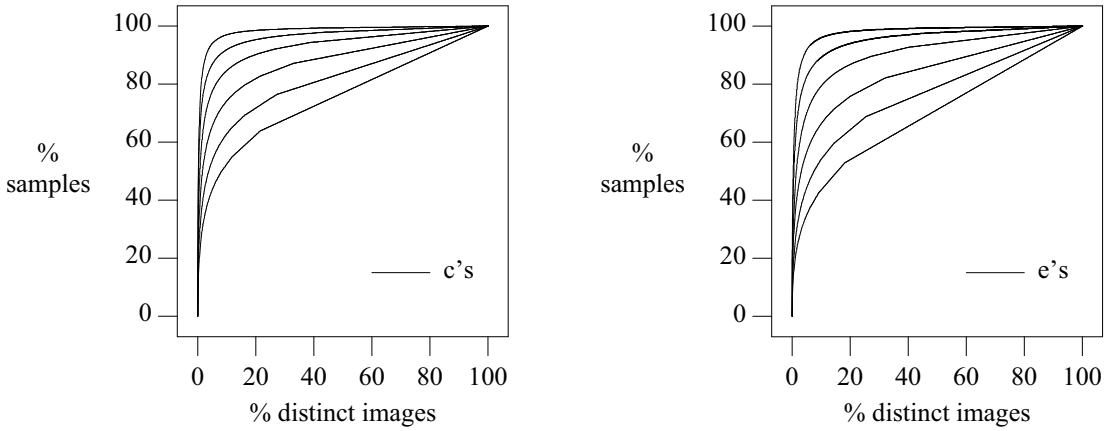
Another interesting observation from Table 1 is the asymmetry between 'c's and 'e's. For all text sizes, larger fractions of 'c's are confusable with 'e's than vice versa. In other words, 'e's are less likely to be identical to a sample of 'c's. This could be caused by the fact that there are more shape variations in 'e's (a larger number of distinct images) than in 'c's, and by the characteristics of our defect model.

Notice that although the distributions overlap heavily, the frequency of occurrence of the images shared by both classes could differ significantly between the two classes. For instance, at 5 point, only 475 (17%) of the 2825 images shared by the two classes occur with equal probability in both classes. This suggests that, by using the frequency information,



(a) (b)

Figure 3: (a) Number of distinct images found in the sample collection; (b) proportion of unseen images as sample size grows.



(a) (b)

Figure 4: Cumulative frequency of images of (a) c's and (b) e's at 5,6,7,8,9 and 10 pt (from top to bottom).

Table 1: Frequency of occurrence of identical images in both classes (measured with 500,000 samples of each class).

Text size	5 pt	6 pt	7 pt	8 pt	9 pt	10 pt
1) Number of shared images in two classes	2,825	8,127	14,113	17,288	14,446	7,917
2) % all distinct images	29.6%	24.3%	15.1%	8.2%	4.1%	1.6%
3) % all distinct images of 'c's	47.6%	42.8%	30.4%	18.0%	8.9%	3.4%
4) % all distinct images of 'e's	43.9%	36.0%	23.1%	13.2%	6.9%	2.8%
5) Corresponding number of samples	987,314	935,117	743,299	478,838	236,578	90,882
6) % all samples	98.7%	93.5%	74.3%	47.9%	23.7%	9.1%
7) % shared in all samples of 'c's	99.0%	95.3%	81.8%	57.3%	32.9%	12.7%
8) % shared in all samples of 'e's	98.5%	91.8%	66.9%	38.5%	14.4%	5.5%

classification of the confusable images can be more accurate than random guesses. In the next section, we attempt to estimate the minimum probability of error for these samples when frequency is used.

3.3 Empirical Estimate of Bayes Error

The Bayes probability of error is the minimum probability of error achieved when the Bayes decision rule is used. For discrimination between two classes c_1 and c_2, the Bayes decision rule for a sample x is to

$$decide\ c_1\ if\ p(c_1|x) > p(c_2|x);\ otherwise\ decide\ c_2.$$

For discrete x, the Bayes error is given by [6]

$$P(e) = \sum_{x \in R_1} p(x, c_2) + \sum_{x \in R_2} p(x, c_1),$$

where $R_1 = \{x|p(c_1|x) \geq p(c_2|x)\}$ and $R_2 = \{x|p(c_2|x) > p(c_1|x)\}$.

We construct a Bayes classifier using this decision rule, where $p(c_1|x)$, $p(c_2|x)$, $p(c_1, x)$ and $p(c_2, x)$ are estimated by frequency substitution with a training sample set of size n. For samples not included in the training set, the probabilities are estimated by the frequencies at their nearest neighbor (under Hamming distance). As n increases, the number of unseen images approaches zero, and so the error rate of this classifier approaches that of an ideal classifier that possesses perfect knowledge of $p(c_1|x)$ and $p(c_2|x)$, that is, it approaches the Bayes probability of error.

Using our data model it is possible in principle to generate samples indefinitely, and so drive up accuracy until it reaches the Bayes limit. This includes exhausting all the unique patterns as well as getting more precise estimates of their probabilities. However, in practice we are constrained by finite computational resources; thus the rate of change of the accuracy estimate as a function of n is of practical importance. As for the choice of an estimator, many have been proposed in the literature [8] [19]. Here we choose to apply the resubstitution method and the holdout method. In the resubstitution method (we denote it R), the sample set that is used to estimate the posterior probabilities and to construct the classifier is also used to measure the error; this is biased optimistically. In the holdout method (H), a reserved sample set of images (10,000 for each class and each point size), disjoint from those used in constructing the classifier, is used to measure the error; this is biased pessimistically. As sample size n increases (and the number of features is held constant), it has been conjectured that the difference between these two estimates vanishes in the limit [19]. It has also been suggested that a linear combination of the two estimates gives an essentially unbiased estimate for the true error[8].

Figure 5(a) shows the estimates and the corresponding 95% confidence intervals obtained for different text sizes, and Figure 5(b) shows the same data on log scale of n

(estimates with $n < 10,000$ are omitted for clarity). The confidence intervals were calculated by considering the correct rate p/q as p successes over q independent Bernouilli (binary-valued) trials, where q is the number of test samples. For a better approximation of normality, the confidence intervals were computed on the transformed statistic $\log(\hat{p}/(1-\hat{p}))$ (where $\hat{p} = 1-R$ or $1-H$) and then back transformed. Since the H estimates were computed using fewer samples, their confidence intervals are larger.

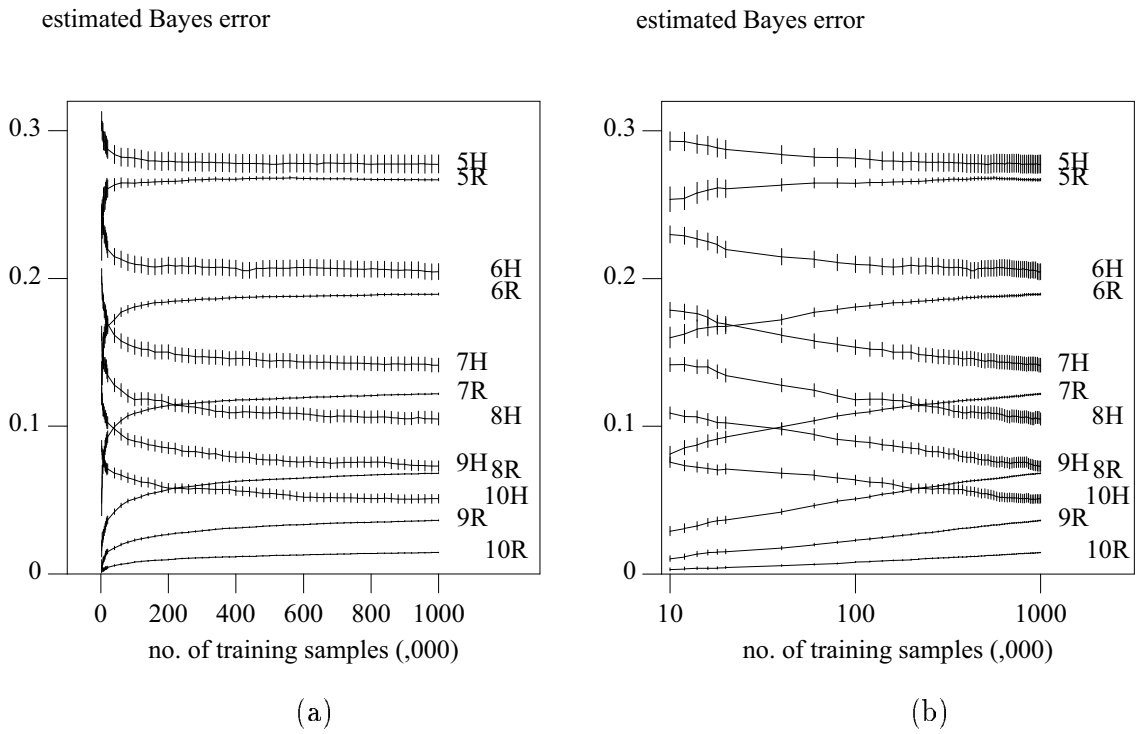


Figure 5: (a) R (resubstitution) and H (holdout) estimates of Bayes error for different text sizes (5,6,7,8,9,10) of Times–Roman 'c','e' at 100 ppi, shown with 95% confidence intervals; (b) the same estimates shown on log-scale of n.

We have seen that even the optimistically biased R estimate of Bayes error is nonzero for all our text sizes. Therefore the goal of building perfect classifiers for them is unrealistic. On the other hand, these problems are less difficult than they have been imagined. For instance, the best achievable accuracy for 5 point images (mostly smaller than 5×5 pixels), under our defect model, is between 71.6% to 73.4% (26.6% to 28.4% error), which is far better than a random guess. The impact of text size on the error rate is obvious.

The R and H estimates at $n = 1,000,000$ differ by only 1.03% for the 5 point images, but they differ by 3.63% for the 10 point images. This is so even though we enjoy a large ratio of n to the dimension of the feature vector ($1,000,000/81$). Although the dimension of the feature vector is the same for all text sizes, the smaller images have constant zeros in most of the dimensions, and thus the larger images have a higher *effective* feature dimension, and this may help explain the differences in the gaps. Another explanation is that new samples

of larger text sizes are less likely to have occurred in the training set, and thus more often have to resort to nearest-neighbor matching.

Achieving sharper estimates may be costly. An extrapolation based on Figure 5 suggests that, for R and H to converge to within 1% on 10 point text, between 10 and 100 million samples may be needed. Of course, more nearly unbiased estimators would help. It seems that extending these experiment to images of higher resolution, and to more classes, may prove challenging.

3.4 Implications for Classifier Design

The classifier being studied is a brute-force design: it simply stores all training samples and records with each the frequency of occurrence of each class. A new image is compared to all the stored samples. If no match is found then the case can be rejected or a nearest-neighbor is located instead. Given our essentially unlimited source of training samples, and assuming we are given sufficient memory, we can make the error rate of this classifier approach the Bayes risk, and the reject rate approach zero (as indicated by the declining fraction of unseen images in Figure 3(b)).

Figures 5(a)(b) and their extrapolations show the range of accuracy achievable with a given number of samples. For example, in our problem, the estimated Bayes error for the 10 point images is between 1.43% and 5.39% with 1,000,000 training samples. If we reject when no exact match is found, then from Figure 3(b) we know that a new sample of 'c' has a 37% chance of being rejected, and for 'e's it is 48%. To achieve this accuracy one would need to store 509,589 unique training samples (Figure 3(a)) and their corresponding class decisions, which takes $509,589 \times (81(\text{the image}) + 1(\text{the class decision}))$ bits $= 5.2$ Mbytes.

4 Asymptotic Accuracy of Two–Class Discrimination

Automatically trainable classifiers sometimes yield a disappointingly low accuracy on isolated–character recognition. It is often unclear whether this is due to flaws in the classification methodology, or inadequacies in the training sets, or both. Given this uncertainty, and the expense of acquiring large and representative training sets, most OCR research in the last few decades has focused on novel methods for classification. If, however, we believed that the quality of training sets, rather than classification methodology, was the determining factor in achieving higher accuracy, then we might choose to devote more effort to improving the quality of training sets.

We have investigated this question through large–scale empirical studies of the asymptotic accuracy of three statistical classifiers, applied to the same 'c' and 'e' discrimination problem as in the first study. By pseudo–random sampling from the parameter distributions specified by our defect model, training and testing sets of arbitrary sizes can be generated. Thus there are no limits, other than those imposed by our computing environment, on the size of these data sets. And, since the training and test sets are both selected at random

from the same distribution, the training set is representative by construction. In this way we control what we feel are the two most important aspects of the 'quality' of image data sets: their size and representativeness.

The three types of classifiers we consider are: nearest neighbors, decision trees, and distribution-map classifiers[11]. These classifiers are similar in that, during training, their 'capacity' can grow indefinitely: that is, their VC–dimension [21] can increase as they are exposed to more training samples. In other respects they are quite different.

4.1 Experimental Design

The values of the defect parameters in this experiment is summarized in Table 2. Note that we generated the training and testing sets from slightly different distributions on the `size` parameter (nominal text size in units of points): the training data is distributed uniformly among sizes {5,7,9,11,13}, and testing data among {6,8,10,12,14}. This is a safeguard against the danger of generating long subsequences of images that are identical in both the training and testing sets. The images were generated at the resolution of 300 ppi, so there are considerably more shape variations than in the previous experiment.

Table 2: Distribution on the parameters of the image defect model.

Parameter	Distribution	Units
randomized per-character ...		
`resn`	fixed (= 300)	pixels/inch
`size`	fixed (= 5,7,9,11,13 for training, 6,8,10,12,14 for testing)	points (1/72 inch)
`blur`	normal ($\mu = 0.7, \sigma = 0.3$)	pixels
`thrs`	normal ($\mu = 0.25, \sigma = 0.04$)	intensity
`skew`	normal ($\mu = 0, \sigma = 1.33$)	degrees
`xscl`	uniform in [0.85,1.15]	dimensionless
`yscl`	normal ($\mu = 1, \sigma = 0.0167$)	dimensionless
`xoff`	uniform in [-0.5,0.5]	pixels
`yoff`	normal ($\mu = 0, \sigma = 0.06$)	ems
randomized per–character and per–pixel...		
`sens`	normal ($\mu = 0.125, \sigma = 0.04$)	intensity
`jitt`	normal ($\mu = 0.2, \sigma = 0.1$)	pixels

Using this model, we created a training set of 500,000 samples (250,000 'c's and 250,000 'e's), and a testing set of 100,000 samples (50,000 'c's and 50,000 'e's). The bilevel images were normalized to 48×48 images by first centering, and then linearly scaling (in X and Y separately) so that width and height just fit.

The great variety of images generated may be appreciated by considering that within the testing set only 26 images of 'c' and 4 images of 'e' were repeated. Further, none of the 'c's were identical to any 'e's. We have not examined the training set exhaustively for

ambiguities, since the computation would be excessive.

4.2 Nearest–Neighbor Matching

Nearest–neighbor matching is appealing for several reasons: the method is relatively simple to implement; and there exists a theorem that, under certain conditions on the class–conditional distributions, its asymptotic error rate is bounded above by twice the Bayes risk [5]. The proof depends on the fact that, as the number of prototypes increases, the nearest neighbor of a sample chosen from the class distributions is, in the limit, identical to the sample itself. In practice, given finite sets of prototypes and the large size of high-dimensional feature spaces, this limiting condition may not apply: Fukunaga [7] has shown that in this case there can be substantial bias in the estimate of the nearest–neighbor error.

The most serious practical drawbacks of nearest–neighbor classification are the potentially exorbitant time and space requirements of naive implementations. Most prior work focuses on pruning the prototypes and speed–optimizing the search. Still, accuracy is bounded above by the results of brute–force matching to the entire training set. Given a relatively unbounded training set, we are now in a position to examine the asymptotic effects of the number of training samples on the classification accuracy.

The metric we use to compare samples is Hamming distance, normalized by the number of black pixels in the prototype. Table 3 summarizes the results of the test. The training set referred to in each entry is a subset of that in the next entry. A reject occurs when a sample's minimum distances to both classes are the same.

The number of errors quickly decreases until 60,000 prototypes are used, and thereafter shows no clear downward trend. So, an accuracy of 99.9% seems to be the best achievable with 500,000 prototypes under normalized Hamming distance.

It should be noted that the choice of a metric has a significance influence on the results of nearest-neighbor matching. For instance, when we used unnormalized Hamming distance instead, the accuracy is significantly lower, and the errors are not symmetric: substantially more 'e's are misclassified as 'c's than 'c's misclassified as 'e's. Arguably there may exist an even better metric than the one we have chosen, but such a metric is nontrivial to find.

To detect possible error concentrations, we analyzed the distribution of the defect parameters associated with errors. We found that errors are most strongly correlated with `size`: 93% of the errors in both tests involve 6 point images, and the rest involve 8 point images. The other model parameters, examined independently, do not correlate strongly with errors. We have not tested for correlations of errors with pairs, triples, etc of parameters, since the sample set is still sparse. Yet this suggests future experiments in which we enrich the set of prototypes with samples of low `size`. Another possibility is to generate additional prototypes by slightly perturbing the parameter vectors associated with the error cases. Both may yield more refined decision boundaries near the current errors.

4.3 Decision Trees

The second type of classifier we examine is decision trees. Their distinctive advantage is speed, and their weakness is rapid error accumulation with depth. We have conjectured that the use of unusually large training sets may circumvent this problem.

Our focus is on accuracy (generalization of discrimination to the test set) rather than on space or time optimization, so we choose to build trees by a greedy heuristic that is sensitive to the quality of training data and is simple to implement. The trees are correct on the training set by construction.

Using each training set, we construct a deterministic, non–backtracking, binary classification tree in which each interior node owns a linear discriminant, and each leaf owns a single class except when there are identical samples shared by two classes. The linear discriminant calculates distance to a hyperplane that is not necessarily parallel to any axis. For this reason this type of trees is referred to as *oblique decision trees.*

Each non-terminal node owning two nonempty sets c_1 and c_2 of training samples is split as follows. We compute the sample mean m_i of each class c_i ($m_i \in \mathbf{R}^{48\times 48}$). A line is drawn from m_1 to m_2. The family of hyperplanes $\{h_1, h_2, ...,\}$ perpendicular to this line, parameterized by their distances to m_1 ($d(m_1, h_j)$), are then examined. The parameter is quantized by fixed increments of $0.05 \times \|m_2 - m_1\|$. For each h, the error

$$e_h \;=\; |\{x | x \in c_1 \wedge d(x,h) > 0\} \;\cup\; \{x | x \in c_2 \wedge d(x,h) \leq 0\}|$$

is calculated, and the h with minimum error e_h is chosen. This proceeds recursively until all leaves own a single class or only identical samples.

Table 3 summarizes the classification results. With small training sets, there are more errors than nearest–neighbor classification. However, accuracy improves quickly. Comparing the results in line 15 of Table 3, we can see that the accuracy is comparable to that of nearest–neighbors. Because of insufficient computational resources, we have not yet grown bigger trees. Nevertheless, it is clear that the increase in training samples helps improve the generalization power of the trees.

As before, more 'e's than 'c's are misclassified. The results using 40,000 training samples agree closely with the apparent accuracy limit with nearest-neighbor classification. Yet only 32 of the 86 errors are identical.

4.4 Distribution-Map Classifiers

In [11][15] we have described a method for using a compact representation of class-specific distributions for classification. The method uses a metric $d(x, c) \geq 0$ which measures the dissimilarity of an image x to a (description of a) class c. Ambiguity arises when there is an x such that $d(x, c) = 0$ for more than one class.

The metric is represented as a 'distribution map' where the occurrence in the training set

of each value of each feature is explicitly recorded. These classifiers tend to be more prone to ambiguities than to errors. In an extreme example of this, if all values of all features occur for all classes, then all images are ambiguous.

Ambiguities will be eliminated if for each pair of classes c_i and c_j, there exists at least one feature whose distribution maps for c_i and c_j do not overlap. Such a feature set may be difficult or impossible to find. A more easily satisfied, but less efficient, condition is: for each pair of classes c_i and c_j, and for each x in c_i, there exists at least one feature in which x has a value that does not occur in the distribution map of c_j. We say that such a feature is an *i,j–discriminating feature for x*.[4]

We used the following heuristic to discover discriminating features for classes 'e' and 'c'. Again, quantized distances to hyperplanes were used as features. At the outset, all the training samples are in a single undifferentiated set. A feature is generated and all samples that it discriminates are removed from the set. The algorithm iterates until all samples of each class are discriminated or no more features can be found to discriminate any of the remaining samples.[5] In the resulting classifier all the features are evaluated. Using this method, we obtained a classifier for each of the training sets 1 through 12. Table 3 summarizes their performance on the test set. We treat all ambiguities as rejects.

Errors and rejects decline rapidly through 5000 prototypes, and then stabilize at values lower than those for nearest–neighbors and decision trees. The asymmetry in errors is also less obvious. Of the 47 errors, 22 are identical to those with nearest-neighbors and 17 are identical to those with decision trees. Only three errors are common to all three methods.

4.5 Comparisons

We have experimentally estimated the asymptotic accuracy of three trainable classification methods, as applied to the same precisely specified recognition problem. Through the use of a parameterized image defect model, we were able to supply a training set that was representative and of unlimited size.

In all three methods, the capacity of the classifier was permitted to expand during training. For all three, under training with larger and larger prototype sets, accuracy rose to an apparent asymptote (Figure 6); this occurred more rapidly for some methods than others. None of the methods required exorbitant time or space resources to approach the asymptote closely.

In other important respects, however, such as time/space demands at runtime, the three classification methods are quite dissimilar. Table 4 summarizes their space and time demands (when training set 10 with 10,000 training prototypes was used). The run time was measured on a Silicon Graphics Power Series Model 4D/480S.

[4] However, in the case when the same pattern is shared by more than one classes, there will not be any discriminating features for the pattern.

[5] This occurs when the two classes share some identical samples.

Table 3: Results of three classifiers on the test set.

train-ing set	size (,000)	nearest-neighbor c #err (#rej)	nearest-neighbor e #err (#rej)	nearest-neighbor total #err (#rej)	decision tree c #err (#rej)	decision tree e #err (#rej)	decision tree total #err (#rej)	distribution map c #err (#rej)	distribution map e #err (#rej)	distribution map total #err (#rej)
1	1	90(0)	95(2)	185(2)	30(0)	635(0)	665(0)	34(204)	139(202)	173(406)
2	2	78(1)	85(1)	163(2)	16(0)	477(0)	493(0)	14(66)	37(140)	51(206)
3	3	60(2)	82(1)	142(3)	22(0)	169(0)	191(0)	19(13)	41(112)	60(125)
4	4	53(2)	81(2)	134(4)	17(0)	364(0)	381(0)	34(8)	37(50)	71(58)
5	5	44(2)	80(0)	124(2)	36(0)	285(0)	321(0)	16(10)	47(27)	63(37)
6	6	47(1)	71(0)	118(1)	38(0)	256(0)	294(0)	31(14)	64(12)	95(26)
7	7	48(0)	76(0)	124(0)	30(0)	266(0)	296(0)	26(6)	52(22)	78(28)
8	8	45(0)	83(0)	128(0)	24(0)	179(0)	203(0)	15(1)	58(12)	73(13)
9	9	47(0)	87(0)	134(0)	15(0)	120(0)	135(0)	29(7)	53(16)	82(23)
10	10	48(0)	90(0)	138(0)	13(0)	138(0)	151(0)	20(2)	58(16)	78(18)
11	20	24(0)	84(0)	108(0)	16(0)	169(0)	185(0)	14(6)	53(47)	67(53)
12	30	23(0)	76(0)	99(0)	10(0)	100(0)	110(0)	16(11)	31(42)	47(53)
13	40	19(0)	77(0)	96(0)	14(0)	81(0)	95(0)			
14	50	19(0)	76(1)	95(1)	14(0)	94(0)	108(0)			
15	60	21(0)	67(0)	88(0)	8(0)	78(0)	86(0)			
16	70	20(0)	70(0)	90(0)						
17	80	20(0)	73(0)	93(0)						
18	90	20(0)	71(0)	91(0)						
19	100	19(0)	69(1)	88(1)						
20	110	23(0)	69(1)	92(1)						
21	120	22(0)	68(1)	90(1)						
22	130	21(0)	70(0)	91(0)						
23	140	20(0)	70(0)	90(0)						
24	150	18(0)	73(0)	91(0)						
25	200	17(0)	71(0)	88(0)						
26	250	16(0)	71(0)	87(0)						
27	300	15(0)	72(0)	87(0)						
28	350	16(0)	72(0)	88(0)						
29	400	16(0)	81(1)	97(1)						
30	450	15(0)	78(2)	93(2)						
31	500	13(0)	79(1)	92(1)						

Table 4: Summary of time/space demands when 10,000 prototypes are used.

classifier	prototype/feature storage	run time per image
nearest-neighbor	28,800 K bytes	18 seconds
decision tree	166 K bytes	0.024 seconds
distribution maps	1,110 K bytes	0.096 seconds

The accuracies achieved (99.9%) are remarkably high, considering the well–known practical difficulty of the problem. Apparently the distribution-map method was more accurate than the nearest–neighbors and decision–tree methods. However, if we count rejects as

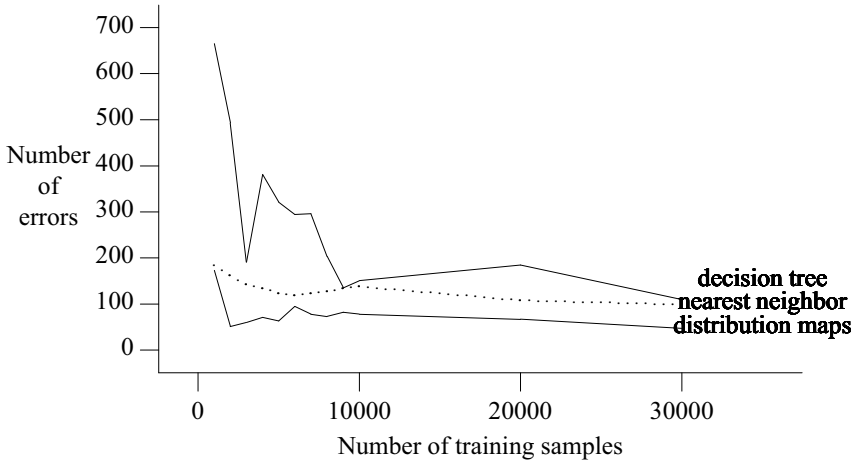


Figure 6: Comparison of number of errors by three classifiers (excluding rejects).

errors, the differences were not significant[6] at 95% statistical confidence.

It would be surprising if this remarkable consistency were a coincidence: therefore, we speculate that the asymptotic accuracy of all three methods is determined more by the characteristics of the training data that they shared, than by the details of their methodology which differ. In other words, we believe that, as long as the training data are representative and sufficiently many, a wide range of classifier technologies can be trained to equally high accuracies. Moreover, impractically expensive computing resources need not be required for this to succeed.

It should be noted that some aspects of the classifiers' performance may depend largely on the nature of the imaging defects. For instance, an asymmetry in errors could be due to the defect model rather than the classifiers: if defects such as scratches and coffee stains were modeled, we might have seen more 'c's classified as 'e's.

We have noted that there is little overlap among the errors made by the classifiers. This suggests that further performance improvements might be possible, for example, through combining their results.

Our results suggest a promising two–part research strategy to build highly accurate classifiers. On one hand, more research is called for to develop image defect models that fit reality as closely as possible. On the other hand, classification methods are needed that can take advantage of unlimited training samples obtained through a precise problem definition.

[6]We estimate the 95% statistical confidence intervals of errors plus rejects, after training on 30,000 samples and testing on 100,000 samples, to be as follows (in per cent): nearest-neighbors with normalized Hamming distance, [0.08,0.12]; decision trees, [0.09,0.13]; and perfect metrics, [0.08,0.12]. Note that each pair of intervals overlaps.

5 Evaluation of OCR Classifier Accuracy Using Synthetic Data

OCR accuracies reported in the literature are often measured on essentially arbitrary data sets, with image defects that cannot be quantitatively specified. Also, due to the high cost of collection and 'truthing,' these data sets are usually too small to provide uniform coverage of defects. Aside from the cost, it has been difficult to talk precisely about image quality. For instance, in [18] image samples are ranked in quality according to the median character recognition accuracy of six OCR systems. Such a ranking often does not agree with human judgement, and will change as OCR technology evolves. The resulting biases in performance measurements limit the projectability of accuracy estimates to unseen data.

In this section we describe early exploration of a methodology for evaluating an OCR classifier using synthesized image data created with our image defect model. Evaluation on synthetic images has several advantages: defect parameters are known precisely for the test data; comprehensive and uniform coverage of the range of defects is achievable; the test can be automated; and the sample size is not limited by the costs of manual 'truthing.'

We foresee numerous applications of such a methodology. If we could map the weaknesses of a classifier, in terms of image defects, then perhaps we could improve it by further training guided by this knowledge. If we could determine that the image quality of a given problem is capable of being narrowly characterized, we could perhaps build or select the best classifier for it, and perhaps estimate the achievable accuracy immediately without further testing. If we could determine that a problem consists of a mixture of image qualities, then perhaps we could combine several classifiers, each matched to one quality [9].

5.1 The accuracy function

Our goal is to map the accuracy of a given classifier in the 12–dimensional parameter space of our defect model. More precisely, we want to locate the region or regions within which the classifier's accuracy is above a certain threshold. The parameters are real numbers and there are infinitely many points in the space. Of course we can hope to estimate accuracy only at a finite number of these points.

The design of search strategies within parameter space will be aided if accuracy, as a function of the parameters, exhibits continuity and monotonicity properties. Informally, we say that a function is continuous if a small change in a parameter value does not cause large fluctuations in accuracy. It is monotonic if there is a single peak at the boundary of the parameter range. It is bi–monotonic if a single peak occurs in the interior. Many optimization search strategies require the function to be no more complicated than bi–monotonic. We will investigate whether or not our classifier, parameterized by our image defect model, exhibits these useful properties.

Strictly speaking, to establish continuity or monotonicity we need measurements at all scales, but this is computationally infeasible. We have adopted the following approximations: (1) choose a few parameters that are of greatest interest; (2) set a fixed sampling

rate in the space, ignoring smaller details; and (3) select a region of interest and examine the function only within that region.

5.2 Choosing parameters of interest

Some image defect parameters are well known to have marked effects on the recognition accuracy: these include text size and spatial sampling rate [18]. However, in many applications, their values are known in advance. We will assume that they have a fixed value. Variations in `xoff` and `yoff` are due to random errors in spatial sampling which are almost impossible to control in practice, and thus we allow them to vary randomly.

In our experience, aside from these, the key contributors to scanning and printing noise are the three parameters `blur` (standard error of the Gaussian blurring kernel), `thrs` (binarization threshold), and `sens` (pixel sensor sensitivity). These will be our primary concern. Figure 7 illustrates these effects.



Figure 7: Effects of blurring, thresholding, and pixel sensitivity. Images are created with `blur` varying from 0.0 to 3.6 by 0.4 (top row); `thrs` varying from 0.9 to 0.0 by -0.1 (middle row); and `sens` varying from 0.0 to 0.9 by 0.1 (bottom row).

The other parameters of the model are of secondary interest. The parameters `skew`, `xscl`, `yscl`, and `jitt` may have more or less effect depending on the features used in the classifier. Skew may have no effect with classifiers using rotation–invariant features. X and y scaling have little effect with classifiers which normalize character size. Jitter has effects similar to pixel sensitivity error. We defer studying these parameters.

Thus we designed our experiment as follows. We chose one fixed value for each of `size`, `xresn`, `yresn`, `skew`, `xscl`, `yscl` and `jitt`. We let `xoff` and `yoff` vary at random. We allow `blur`, `thrs`, and `sens` to vary in a controlled way over a range, and study the accuracy of the classifier as a function of these three.

5.3 Choosing values of the parameters

We fix the spatial sampling rate in both x and y directions (`xresn` and `yresn`) at 300 pixels per inch, and the text size `size` at 10 point. Skew is zero, scaling factors are 1.0, and jitter is disabled. We let `xoff` and `yoff` vary randomly uniformly within the range [-0.5,

0.5) pixels. Parameters `blur`, `thrs` and `sens` vary within a range at fixed intervals: `blur` varies from 0.5 to 2.5 (inclusive) by a step of 0.5; `thrs` varies from 0.2 to 0.4 (inclusive) by 0.05; and `sens` varies from 0.0 to 0.1 (inclusive) by 0.025. These ranges were selected so that for most symbols the images neither vanish nor become shapeless black blobs. They are also consistent with the values used in the public–domain *Bell Labs BLidm0 Character Image Database* [20]. There are five values for each parameter, and thus 125 triples. We explore the behavior of the classifier over this lattice of 125 points in parameter space.

5.4 Classifier Evaluation

For each parameter point, we pseudorandomly generated 50 sample images for each of the 94 printable–ASCII symbols:

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

in the Adobe Times–Roman typeface. Thus a total of 6,250 samples were generated for each symbol. Half of these were used for training and the rest for testing. During testing, the 94 symbols are assumed to occur equiprobably (in a specific application, the distribution may be highly non–uniform: modifying our methodology for this should be straight–forward).

We will illustrate our methodology on a binary decision tree classifier where each internal node is associated with an oblique (i.e. non–axis–parallel) hyperplane [12]. The tree is derived automatically from the 293,750 ($25 \times 125 \times 94$) samples of training data.

5.4.1 Accuracy as a function of the parameters

Figure 8 shows the measured accuracy as a function of `blur`, `sens` and `thrs`. Except for a few cases where some images vanished, at each point the accuracy is averaged over 2,350 samples (25×94). Accuracy varies from 83.74% (occurred at {`blur`, `sens`, `thrs`} = {2.5,0.1,0.4}) to 98.17% (at {1.5,0.0,0.2}).

From Figure 8 we can see that over the region and at the coarse scale that we established, the accuracy appears to be continuous as a function of all three parameters. As a function of `blur` or `thrs`, it is monotonic (decreasing). Parameter `sens` has little effect except at extreme lower values of `blur` or `thrs`.

The fact that accuracy peaks at less than 100% is due to intrinsic ambiguities of shape such as case confusions and 'comma' *versus* 'single quote': these cannot be resolved in the absence of geometric and linguistic context. However, the consistently low accuracies at higher values of `blur` and `thrs` are certainly due to degradation in image quality.

The individual effect of each parameter is shown in Figure 9. At higher values of `blur` ($e = 2.5$) or `thrs` ($t = 0.4$), the function fluctuates non–monotonically as `sens` changes,

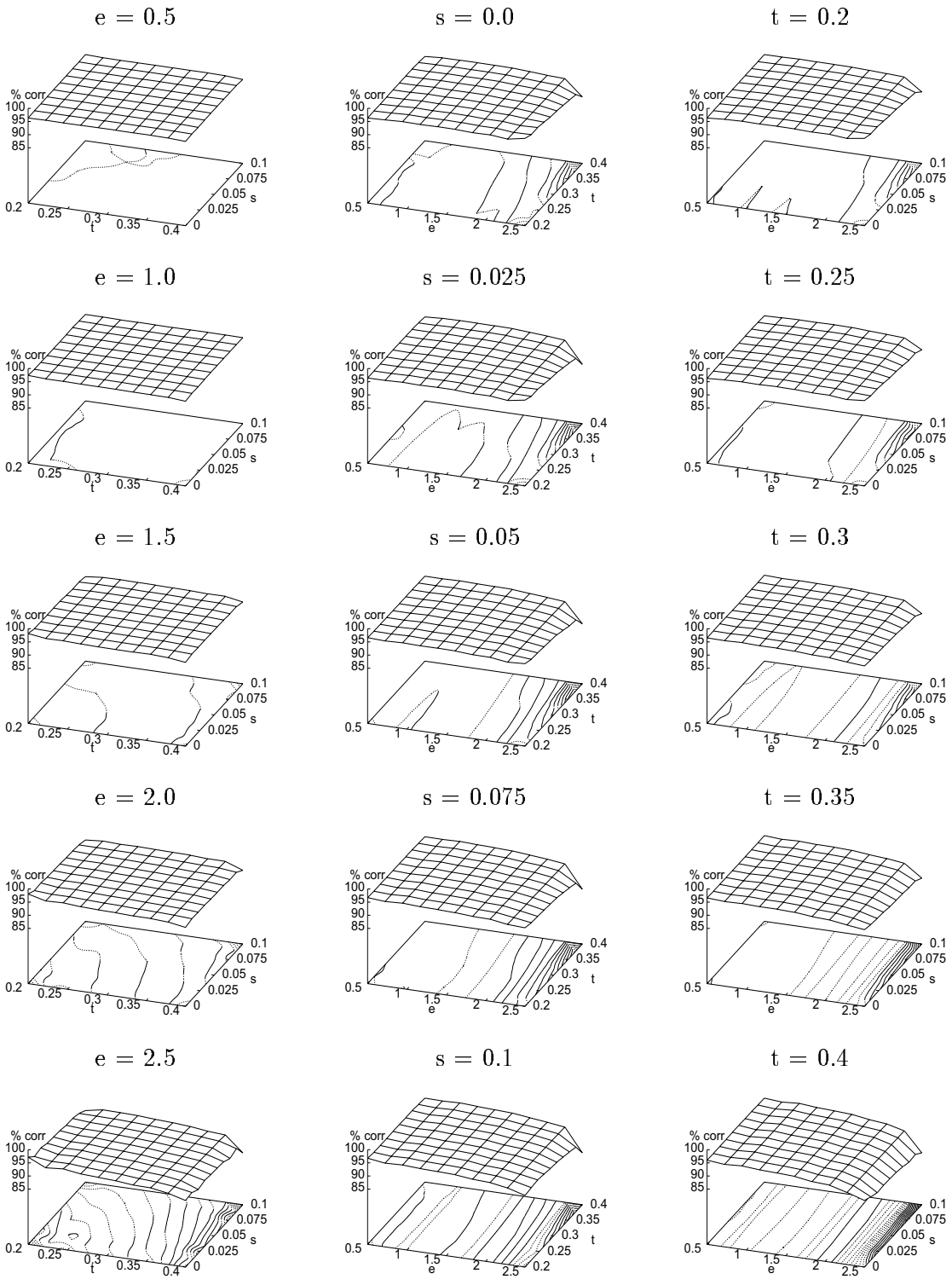


Figure 8: Accuracy plotted against parameters e (`blur`), s (`sens`), and t (`thrs`). The contours are at intervals of 0.5%.

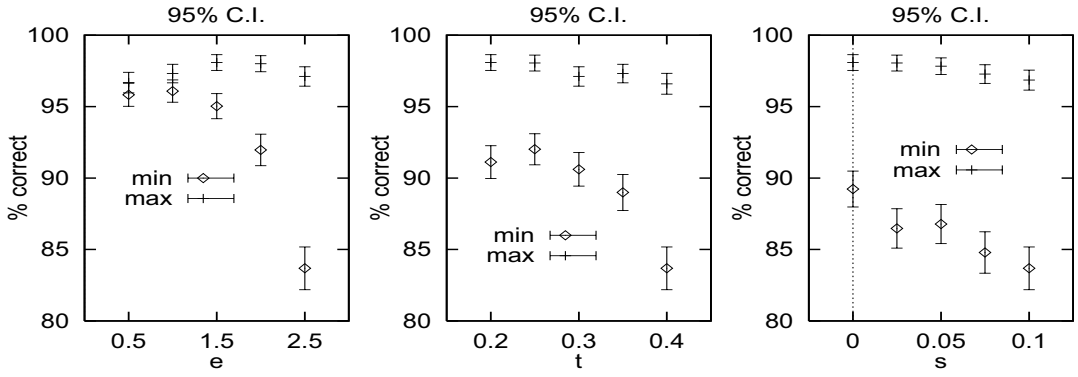


Figure 9: 95% confidence intervals of the maximum and minimum correct rates at fixed values of each parameter.

but the confidence intervals suggest that this is mostly due to measurement error. One conclusion from this experiment is that this particular classifier is sensitive to changes in `blur` and `thrs`, which become good predictors of its accuracy.

The effect of the parameters on accuracy varies among symbols. In other words, accuracy on some symbols is more sensitive to defects than others. For symbols `2469ACMkm` perfect (100%) accuracy is maintained throughout the entire range of defects we examined. The other symbols can be ordered as follows according to increasing difference between the highest and lowest correct rates over the region:

```
378JKPQXYbdghpqr#&$5BDEGHNRSVWZacnuy@\+FLUe
svx/^0OoTtz=%({w>"?}f!~<j)]*[I:1l';i-'_|,.}
```

5.4.2 Domain of competence in the parameter space

The domain of competence of the classifier in the chosen parameter space can be shown as regions where the accuracy is above or below certain thresholds (Figure 10). As noted before, the parameters associated with accuracies below each particular threshold tend to locate in contiguous regions in the space.

Similar maps can be made to compare the accuracies of different classifiers. For instance, the tree-based classifier we evaluate in this study can be modified to use multiple trees and votings of their decisions. It has been observed that the accuracy improves asymptotically as more trees are added to the classifier[10]. To compare the accuracies, we map the values of the parameters associated with accuracies below 96% when different numbers of trees are used. Figure 11 shows such a map. It can be seen that the weak regions shrink with the increase in the number of trees used.

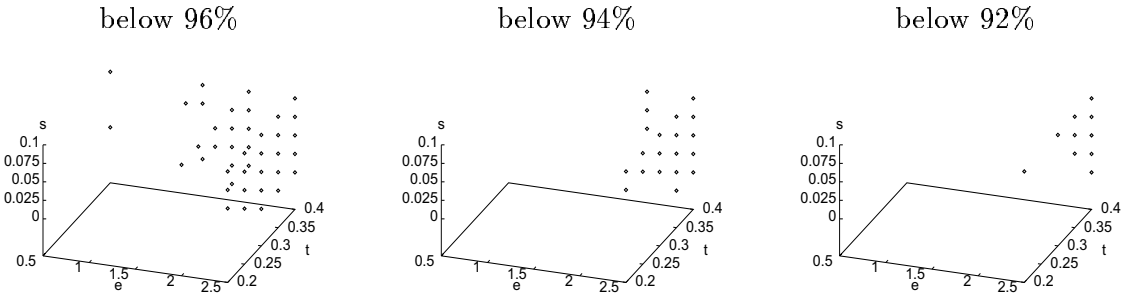


Figure 10: Parameter values associated with accuracies (% correct) below various thresholds.

5.5 Discussions

We observed that blurring and thresholding, and to a lesser extent pixel sensor sensitivity, affect accuracy nearly continuously and monotonically, within measurement error. Thus good classifier performance is constrained to a contiguous region in parameter space. Blurring and thresholding have strong effects on classifier accuracy, and affect some symbols much more than others. Pixel sensor sensitivity plays an important role only when either of the other two parameters are at the margins of good performance.

In our evaluation we have assumed all symbols and all defects in the interesting range are equally probable. In real–world applications those probabilities vary, which need to be taken into account when accuracies are projected to realistic page images. The projection will also rely heavily on successful calibration of the model for real data.

6 Conclusions

We have experimented with a new methodology for studying the character recognition problem. Large-scale simulations using the defect model permitted exploration of many interesting aspects of the problem under a controlled environment. The more important observations are that the problem has non-zero intrinsic error under specific defect conditions, that accuracies of classifiers ultimately depend on the quality of the training data,

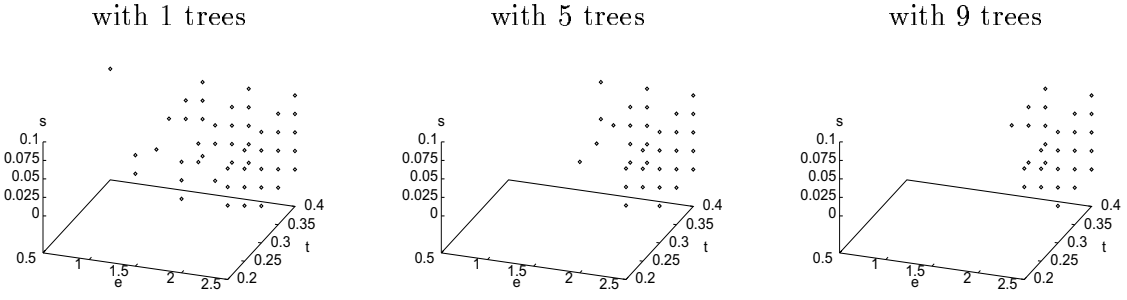


Figure 11: Parameter values associated with accuracies (% correct) below 96% as the complexity of the classifier grows.

and that two defects (blurring and thresholding) affect classification accuracy significantly, continuously, and monotonically. We believe that these results, as well as the methodology, can be further exploited to guide the training of individual classifiers and coordination of multiple classifiers.

We have focused our attention on methodologies for making use of explicitly defined image degradation models. We have not attempted here to justify the design of the particular model we have used. This important question is being actively researched, with preliminary results available in [3] [16]. We recognize that successful engineering applications of such models depends significantly on the validity of the models. Certainly these models can be improved: but, we present our experiments as examples of interesting, useful, and even illuminating results that can be obtained based on the present state of the art.

Acknowledgements

We are thankful to Mark Hansen, David Ittner, Dz-Mou Jung, George Nagy, Arnold Neumaier, Daryl Pregibon, and Margaret Wright for their helpful comments.

References

[1] H.S. Baird, R. Fossey, "A 100-Font Classifier", *Proceedings of the first International Conference on Document Analysis and Recognition*, St.-Malo, France, pp. 332-340, September 20–October 2, 1991.

[2] H.S. Baird, "Document Image Defect Models", in H.S. Baird, H. Bunke, K. Yamamoto (Eds.), *Structured Document Image Analysis*, Springer–Verlag, pp. 546-556, 1992.

[3] H.S. Baird, "Calibration of Document Image Defect Models", *Proceedings of the 2nd Annual Symposium on Document Analysis and Information Retrieval*, pp. 1-16, April 26-28, 1993.

[4] H.S. Baird, "Document Image Defect Models and Their Uses", *Proceedings of the Second International Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, pp. 62–67, October 20–22, 1993.

[5] T.M. Cover, P.E. Hart, "Nearest Neighbor Pattern Classification", *IEEE Transactions on Information Theory*, **IT–13**, 1, pp. 21–27, January 1967.

[6] R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Addison-Wesley, New York, 1973.

[7] K. Fukunaga, D.M. Hummels, "Bias of Nearest Neighbor Error Estimates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI–9**, 1, pp. 103–112, January 1987.

[8] D.J. Hand, "Recent Advances in Error Rate Estimation", *Pattern Recognition Letters*, **4**, pp. 335-346, 1986.

[9] T.K. Ho, *A Theory of Multiple Classifier Systems And Its Application to Visual Word Recognition*, Doctoral Dissertation, Dept. of Computer Science, SUNY at Buffalo, 1992.

[10] T.K. Ho, "Random Decision Forests", *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, Canada, pp. 278-282, August 14-18, 1995.

[11] T.K. Ho, H.S. Baird, "Perfect Metrics", *Proceedings of the Second International Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, pp. 593–597, October 20–22, 1993.

[12] T.K. Ho, H.S. Baird, "Asymptotic Accuracy of Two–Class Discrimination", *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, pp. 275-288, April 11-13, 1994.

[13] T.K. Ho, H.S. Baird, "Estimating the Intrinsic Difficulty of A Recognition Problem", *Proceedings of the 12th International Conference on Pattern Recognition*, Jerusalem, Israel, pp. 178-183, Oct 9-13, 1994.

[14] T.K. Ho, H.S. Baird, "Evaluation of OCR accuracy using synthetic data", *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pp. 413-422, April 24-26, 1995.

[15] T.K. Ho, H.S. Baird, "Pattern Classification with Compact Distribution Maps", *Computer Vision and Image Understanding*, to appear, 1998.

[16] T. Kanungo, R.M. Haralick, H.S. Baird, "Validation and Estimation of Document Degradation Models", *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pp. 217-225, April 24-26, 1995.

[17] S.V. Rice, J. Kanai, T.A. Nartker, "An Evaluation of OCR Accuracy", in *Information Science Research Institute, 1993 Annual Research Report*, University of Nevada, Las Vegas, pp. 9–20, 1993.

[18] S.V. Rice, J. Kanai, T.A. Nartker, "The Third Annual Test of OCR Accuracy", in *Information Science Research Institute, 1994 Annual Research Report*, University of Nevada, Las Vegas, pp. 11–38, 1994.

[19] G.T. Toussaint, "Bibliography on Estimation of Misclassification", *IEEE Transaction on Information Theory*, **IT-20**, 4, pp. 472-479, July 1974.

[20] English Document Database I & II CD–ROM Set, The Intelligent Systems Laboratory of the Department of Electrical Engineering, University of Washington, Seattle, WA, Fall 1993.

[21] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer–Verlag, 1982.