THE SAFE USE OF SYNTHETIC DATA IN CLASSIFICATION

by
Jean E. Nonnemaker

A Dissertation
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Doctor of Philosophy
in
Computer Science & Engineering Dept.

Lehigh University
December, 2008

This dissertation is accepted in partial
fulfillment of the requirements for the degree of
Doctor of Philosophy.

(Date)

Henry S. Baird

Edwin J. Kay

Roger N. Nagel

John. R. Spletzer

Wei-Min Huang

Acknowledgments

I would like to thank Professor Henry Baird for all his help, Without him this work would not have been possible. I would like also to thank my committee members, Professors Edwin J. Kay, Roger N. Nagel, John R. Spletzer, and Wei-Min Huang for their careful comments and encouragement. I am grateful to my friend and co-worker, Gerald Lennon, for his models of shell scripts and assistance with the sed editor and commands. I would like to thank my co-worker, Pauline Chu, who encouraged me by her persistence and example in obtaining her own doctorate after many years. Pauline was the friend who was always available for a talk. I would also like to offer a hearty thank you to Wayne Mery for his assistance in printing the many iterations of this document. Last but certainly not least, I would like to express my appreciation for the support and patience of my husband, Michael, who spent many days hiking alone with our dogs so that I might be able to work without distractions at home. He was ever patient when I worked into the wee hours of the night, and didn't complain too much when some of the day to day things did not always get done.

Contents

List of Figures

Abstract

When is it safe to use synthetic data in supervised classification? Trainable classifier technologies require large representative training sets consisting of samples labeled with their true class. This is in the context of supervised classification in which classifiers are designed fully automatically by learning from a file of labeled training samples. Acquiring such training sets is difficult and costly. One way to alleviate this problem is to enlarge training sets by generating artificial, synthetic samples. Of course this immediately raises many questions, perhaps the first being "Why should we trust artificially generated data to be an accurate representative of the real distributions?" Other questions include "When will training on synthetic data work as well as — or better than — training on real data?"

We distinguish between sample space (the set of all real samples), parameter or generator space (samples that can be generated synthetically), and finally, feature space (samples described by numerical feature values). Synthetic data can be produced in what we call parameter space by varying the parameters that control their generation. We are interested in exploring how generator and feature space relate to one another. Specifically, we have explored the feasibility of varying the generating parameters for typefaces in Knuth's Metafont system to see if previously unseen fonts could also be recognized.

Generally, we have attempted to formalize a reliable methodology for the generation and use of synthetic data in supervised classification. We have designed and carried out systematically a family of experiments in which pure typefaces already widely used are supplemented with synthetically generated typefaces interpolated in generator or parameter space in the Metafont system. We also vary image quality widely using a parameterized image defect generator.

We have found that training on interpolated data is for the most part safe, that is to say

never did worse when tested on the pure samples. Furthermore, the classifier trained on interpolated data often but not always improved (about one third of the time) classification when tested on previously unseen interpolated samples.

Chapter 1

Introduction

It has been widely accepted in pattern recognition research that the classifier trained on the most data wins Ho, T. K. and Baird, H. S. [HB97], Simard, P. Y., LeCun, Y. A., Decker, J. S. and Victorri, B. [SV98], Varga, T. and Bunke, H. [VH04]. Of course, putting this strategy into practice can be troublesome, since large training sets are expensive or impossible to obtain, and may not be representative - or sets may be imbalanced, where one class is represented by too few samples, and others have too many. This is in the context of supervised classification in which classifiers are designed fully automatically by reading in files of labeled training samples so that the classifier can learn from example patterns.

Good results in pattern recognition have been achieved by the use of supervised classifiers such as nearest neighbors algorithms(kNN) [1] [DA91]; however these results require large amounts of training data together with carefully labeled g*round truth* (the true classes of each sample) which can be even more expensive to provide than the data themselves. Often the acquisition of such data becomes a problem, as much time and labor must be spent to locate existing training data, or alternatively, to classify new training data properly.

[1] The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors.

We believed that one way out of this impasse is to *amplify* the training data: that is to increase it artificially by generating more. We called such generated data *synthetic* in contrast to data collected in the field, *real* data. The generation of data can be completely automated and is often inexpensive, but when is it safe? That is, when will synthetic data perform as well as real data? It is easy to imagine how it could perform worse! We posed the following questions. Given a set of training data, can we generate more that is still representative of the underlying distribution? When can you reliably use such data? What are valid ways of generating these new data?

We thought it would be useful if a way could be found to amplify existing data so that it could be used in classification programs with results equivalent to that of data gathered in the usual, tedious and time-consuming fashion. To this end, we conducted experiments to explore the validity and uses of artificial data.

In practice, researchers tend to use datasets because a) others have used them, b) they are freely available, c) they possess a documented collection protocol, and d) they are uncommonly large. A synthetic data set may enjoy most or all of these properties. It can be offered for use by others at low cost. The underlying model of the generated data may be arguably a carefully documented protocol, and of course, perhaps the overwhelming advantage of such a data set is that the size is effectively unbounded! The most serious drawback of synthetic datasets is that people don't trust them to be representative of real distributions.

We made the assumption, as is done classically in Bayesian decision theory, that we had a classifier technology which could be trained on a training set and tested on a separate set. Typically the training samples arise in a natural *sample space*. In our domain, the sample space was the set of all document images.

Usually the first step in crafting a classifier is to choose numerical features (say d of them) that can be algorithmically extracted so each sample is represented by a set of features: that is, as a point in d-dimensional space. In this sense, data also live in *feature space*.

However, since we know how images are generated, then the parameters that control the generation process are yet another way of describing the data – so synthetically generated data can be said to live in *parameter space* also.

It may be informative to give an example of a method of generating artificial data which 'lives' in each space. For example, in sample space, one might generate synthetic data by taking several real images, cutting them up, pasting them together in a random fashion, and so, making a collage of them.

To generate data synthetically in feature space, one might extract features for several samples, giving several points in feature space, then interpolate between the points to generate a new point. An example of feature space data generation is outlined in more detail in our small experiment described in the following pages. In this example, the features reddish, greenish and blueish have numerical values assigned to them, and new feature points are generated by interpolating between the values to obtain new reddish, greenish and blueish values.

Samples that are generated in parameter space were synthesized by varying the generating parameters. In the case of document images, these generating parameters were among the following; type, size, image degradations such as blur, threshold, additive noise, etc. and layout dimensions such as gutter width, line spacing. We can say that pairs of real images span ranges in parameter space and thus allow the generation of synthetic training data densely within that range.

We explored the relationship between parameter space, sample space and feature space which can be thought of as follows:

$$\text{parameter space} \rightarrow \text{sample space} \rightarrow \text{feature space}$$

One can think of samples as living in a natural space, but for pattern recognition purposes we tend to extract a small number of features, so each image is represented by these. Starting from any given sample or samples, we can say that we extract features from the sample(s), and that we infer parameters of generation for the

sample(s). Given that, you can imagine three different ways to generate data 1) take the sample and add noise (sample space), 2) change the generating parameters and generate new samples (parameter space), and 3) modify the individual features (feature space).

As we explored methods for generating synthetic data we saw that some methods were naturally more suited to one of these three spaces. In our work, we will concentrated on parameter space or feature space. As we will see later, much of the prior work has concentrated either on adding data in sample space, or is limited to feature space. As part of our research, one of our intentions was to formalize and explain the use of parameter space.

Given two real images, we wanted to identify a continuum in some parameter space with the properties that image one is generated by some value (say a) of the parameter, which we can estimate, and image two is generated by some other value (say b) of the same parameter. All images generated by values in between a and b can occur in real images, and therefore should be recognized correctly by our system. Note that we do not speak of probabilities in this context, just possibilities, so this argument sounds like a primitive uniform likelihood function.

Chapter 2

Literature Review

We have researched four topics, choosing to summarize papers which concern 1) image transformation methods, 2) boosting methods, 3) theoretical papers and 4) relevant papers from other disciplines, such as a subset of papers from the software testing and engineering fields. Among the papers are also some interesting studies of the problem of imbalanced training sets and papers that measure classifier improvements as a function of sample size.

2.1 Image Transformation Methods

Most of the first group of papers use the addition of synthetic data in their experiments to test various aspects of pattern recognition. Several of them use parameterized image degradation methods which are applied to existing samples to create new samples. Several of the applications use interpolation to determine the parameters to be applied. However, none of the experiments use interpolation to create the original samples. Among the domains explored are handwritten lines of text, individual Japanese Kanji characters, and the generation of synthetic features derived from the domains of medical cell samples and satellite images.

Varga, T. and Bunke, H. [VH03]

The performance of a recognition system is strongly affected by the size and quality

of the training data (Baird 2000). In an effort to further investigate this assertion, the authors have conducted experiments to examine under what circumstances a larger and more varied training set improves the accuracy of handwriting recognition systems. The paper discusses how synthetic generation of new training samples can be achieved through perturbation of, or interpolation between the original samples.

The authors maintain that the use of synthetic training data doesn't necessarily lead to an improvement of the recognition rate, because even though the variability of the training set improves, leading to a potentially higher recognition rate, the synthetic training data may bias a recognizer towards unnatural handwriting styles. The paper examines the use of parameters which govern aspects of handwriting style such as slant, the number of Gaussians used for distribution estimations, distortion strength, and training set size. Each geometrical transformation is controlled by a continuous nonlinear function (based on the cosine) which determines the strength of the transformation at each horizontal or vertical coordinate position of the textline. Shearing and vertical scaling are performed with respect to the lower baseline, as well as thinning, thickening and greyscale operations using a perturbation model.

In their experiment, the authors investigated the effects of three parameters when using a Hidden Markov Model (HMM) [RA89] based cursive handwriting recognizer. The first parameter was the number of Gaussian components, while the second parameter was the distortion strength. The number of natural textlines in the training set was the third parameter. A better recognition rate was anticipated as the size of the training was increased. The experimenters tested training sets of 81, 162, 243 and 324 textlines, as well as both six Gaussians and single Gaussians.

The authors found that single Gaussians caused greater variation in recognition rates than six Gaussians, possibly because unnatural looking synthetic textlines in the training set may cause serious damage in the parameter estimation. Also, they found that for larger training sets, the positive effect of adding synthetic data becomes smaller, and the negative effect of unnatural looking textlines dominates. They conclude that a sufficiently large number of parameters in the HMM output

distribution is vital, so unnatural looking synthetic textlines cannot cause damage in the estimation of parameters in the training phase. Furthermore, they found that when using larger training sets with great variability, only rather weak distortions could be expected to produce improvements in the recognition rate.

In our work, we also vary the distortion of the synthetic data. While we did not vary the size of the training sets we did alter the makeup of the training sets, that is to say the percentage of synthetic data versus non-synthetic, or naturally found data. The authors also varied the features they chose, in their case 6 versus 1 Gaussian.

Chawla, N., Bowyer, K., Hall, L. and Kegelmeyer, W. [CH02]

The cost of misclassifying an abnormal (interesting) example as a normal example is often much higher than the cost of the reverse. Undersampling of the majority (most frequent or normal) class has been proposed as a good means of increasing the sensitivity of a classifier to the minority (less frequent or abnormal) class. The authors use a combination of over-sampling the minority class and under-sampling the majority class to achieve better classifier performance. They also create synthetic minority class examples. As the minority class is over-sampled by increasing amounts, the effect is to identify similar but more specific regions in the feature space as the decision region for the minority class. The authors describe experiments on nine different datasets which vary in their size and class proportions. Some examples are a diabetes dataset, A yeast anticancer drug screen, and a forest cover cartographic dataset.

The authors maintain that if you merely replicate the minority class, the decision region for the minority class becomes very specific and will cause new splits in the decision tree, leading to overfitting. Instead they propose to over-sample the minority class by creating synthetic examples, rather than over-sampling with replacement. They generate synthetic examples in a less application-specific manner, by operating in ‘feature space’ rather than ‘data space’.

Here is an example of their algorithm: If the amount of oversampling is 200%, two

neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each neighbor from the original sample. Synthetic samples are generated by taking the difference vector between the feature vector(sample) under consideration and its nearest neighbor. They multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features.

They claim that by applying a combination of under-sampling and over-sampling, the initial bias of the learner towards the negative (majority) class is reversed in favor of the positive (minority) class.

Unlike this work, in which the authors generate synthetic samples in feature space, we have chosen to generate our synthetic samples by interpolation in parameter space. In other words, we modify the generating parameters to produce our synthetic samples. Additionally, we do not have minority and majority class, as in our domain the classes may occur with similar frequency.

Varga, T. and Bunke, H. [VH04]

This paper makes the point that the performance of a handwriting recognition system is strongly affected by the size and quality of the training data. As in their previous paper, the authors emphasize that the classifier that is trained on the most data wins, a viewpoint with which we agree.

The authors discuss the fact that synthetic generation of new training samples can be achieved in different ways, such as interpolation between the original samples. Most existing methods are related to isolated character recognition. In this work, the authors tested the use of continuous nonlinear functions that control a class of familiar geometrical transformations applied on an existing handwritten text line. Thinning and thickening operations were also used.

Variation in handwriting is due to letter shape variation and also the large variety of writing instruments. Their perturbation model incorporates some parameters

with a range of possible values, from which a random value is picked each time before distorting a textline. The authors conducted a series of experiments to compare improvements by expansion of the training set with synthetic textlines versus expansion of the training set with natural textlines only.

Our work differs from this in that we use training sets all of the same size. In our experiments we test the effects of replacing much of the naturally occurring data with synthetic samples. We also interpolate between the original samples, in part methodically and systematically, and in part randomly.

Ho, T. K. and Baird, H. S. [HB97]

In this paper, the authors present three studies that rely on synthetic data generated pseudo-randomly in accordance with an explicit stochastic model of document image degradations. The authors maintain that the image data sets used to train and test classifiers are often gathered unsystematically without following a published protocol. Among the drawbacks with using these databases are that the image collections are often unsystematic, not extensible, and much too small. Thus, they 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 to generate test samples.

The authors use a parameterized model of document image defects to generate documents. Among the parameters are size, spatial sampling rate, blur, sensitivities among the pixels sensors (sens) and variations of jitter. Their generator creates one image when the value of each parameter is fixed and when sens and jitt (jitter) are zero. The effects of pixel sensitivity and jitter are randomized per pixel. By sampling pseudo-randomly from this multivariate distribution, they are able to generate an indefinitely long sequence of distorted images for a given prototype. This model is designed to produce shape distortions similar to those occurring in real-world document images.

Given their essentially unlimited source of training samples and sufficient memory,

they can drive down the error rate of the classifier to approach the Bayes risk (this is the irreducible error rate of a particular recognition problem given a set of features) [DU01] . The authors pose the idea that if the quality of training sets, rather than classification methodology was the determining factor in achieving higher accuracy, then one might choose to devote more effort to improving the quality of training sets.

One of the types of classifiers the authors use to test their ideas is the nearest neighbors because the asymptotic error rate is bounded above by twice the Bayes risk. Using their generated data, they found that the number of errors quickly decreases until 60,000 or so prototypes are used under normalized Hamming distance [HA50]. Likewise, using decision trees, they found that an increase in training samples helps improve the generalization power of the trees. They 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. They have also found that there is little overlap among the errors made by the classifiers, suggesting that further improvements might be possible through combining their results. They believe that classification methods are needed that can take advantage of unlimited training samples obtained through a precise problem definition.

They have found that 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. However, in their evaluation they have assumed that all symbols and all defects in the interesting range are equally probable, while in reality, those probabilities vary, which need to be taken into account when accuracies are projected to realistic page images.

In our work, we utilize the same parameterized model of document image defects. We have extended this work by generating synthetic images of individual characters on which we can apply the document image defects. By using our synthetic character generation model along with the document image defect generator we are

able to systematically and repeatedly create a set of images on which to conduct our experiments.

Sun, J., Hotta, Y. and Naoi, Y. K. S. [SU04]

The recognition rate for highly degraded characters is still a bottleneck for most OCR machines. The authors maintain that this is because most feature extraction methods are based on binarization. Binarization removes the noise while preserving the structural information of the characters. Unfortunately, as the degradation is increased, structure information is lost as this information can be embedded in the noise. The authors propose a new feature extraction method based on a dual eigenspace decomposition which is used along with degradation model.

They generate low resolution character images of Japanese Kanji characters using a video degradation model, common in facial recognition systems, based on perspective transformation and super-sampling. A mask is generated for creating synthetic images, and by controlling parameters such as size of scene plane, view point distance, and focal length, characters with different degradation levels are generated and rendered.

In their experiments with the dual eigenspace method of feature extraction, the authors have obtained better results than with other methods such as contour directional feature and principal component analysis.

The primary concern of this paper is with the results obtained using their new feature extraction method, however the use of the degradation model in this experiment was of more interest to us. As in our experiments, the authors apply a parameterized degradation model to individual, normalized characters in several specific fonts. Our work, however, differs in that we also generated characters in new fonts created by interpolating between existing fonts.

Cano, J., Perez-Cortes, J., Arlandis, J. and Llobet, R. [CL02]

The authors examine the expansion of the training set by synthetic generation of handwritten uppercase letters via deformations of natural images. They performed

extensive experiments adding distorted characters to the training set and the results were compared to directly adding new natural samples to the set of prototypes. The authors used the kNN rule because of its good results, ease of use, and theoretical properties.

The temporal cost of a kNN method can be reduced by reducing the number of prototypes (editing, and condensing, etc.) without degrading classification power, or alternatively, by using a fast nearest neighbors search algorithm. The power of reducing the number of prototypes resides in the smoother discrimination surfaces created by eliminating redundant and noisy prototypes. Cleaner discrimination surfaces reduce the risk of overtraining.

The authors maintain that in most pattern recognition applications, there is no need to guarantee that the exact nearest neighbor has been found. Since the temporal cost of the classification grows approximately with the logarithm of the size of the training set (slowly), an interesting approach to improving the accuracy while keeping high recognition speeds, is to insert new prototypes by performing controlled deformations on the characters to insert them into a larger training set. In this work, they perform experiments to validate that approach.

They test four kinds of transformations, slant and shrink to account for geometric distortions, and erosion and dilation to deal with different writing implements, etc. All transformations were applied to the original binary images. They found that the insertion of artificially slanted, eroded and dilated images produced significant improvements, however, the inclusion of shrunken images did not. The authors also found that the best value of k seemed to only gradually increase as the database grew.

The authors performed another experiment to test whether the addition of real data instead of ‘deformed’ images is preferable. In their experiment they found that this is indeed so, however, real data can only be added if if the additional images are available.

This work is similar to several of the other papers in that image degradation parameters are used to transform original, existing images. Unlike our work, parameters are not used to generate entirely new font types.

Mori, M., Suzuki, A., Shio, A. and Ohtsuka, S. [MO00]

In the recognition of handwritten characters, increasing the number of training samples increases the recognition performance since the training samples can cover wider variations in deformation. The authors propose using, for each category of samples, the pattern correspondence between the training samples and a template pattern. The template is derived from the average of all training samples of the same category.

Once the authors have determined a template, they set a standard displacement path for each pixel pairing between a sample and the template. They create new displacement paths by varying the standard ones with a variation parameter. Using a value of 1 as a parameter generates a pattern that is almost identical to the template. Minus values yield distortion away from the template.

Their results indicate that the recognition rate increases as the number of the training samples increases. Prior to adding generated samples to the original ones, they test which training samples were recognized using a dictionary holding only generated samples. Generated samples that degraded recognition performance were removed, and the remaining samples were used in the experiment.

Their results show that a slight degree of distortion can improve recognition, however large levels of distortion (both positive and negative) lower the recognition rate. The addition of generated samples yields 40-50% of the improvement achieved by doubling the number of original samples. Deforming the samples away from the template can degrade recognition performance. They reason that all samples that are deformed towards the template lie inside the class boundary in the feature space, and some may describe the boundary in more detail. On the other hand, samples deformed away from the template may cross the true class boundary and lead to

recognition error. In other words, a generated sample that leads to correct recognition of original samples does not always contribute to correct recognition of test samples and vice versa. They feel that a better method of selecting samples to be used in the training is needed.

Training samples that are deformed towards the template might be thought of as interpolated between the sample and the template. Samples that are deformed away from the template might then be thought of as extrapolated. It is not surprising, but very interesting, that performance degrades as the new samples approach and even cross the class boundaries. This is one reason why we have chosen not to attempt to explore whether extrapolation is safe. Our work deals with interpolated samples also, however the interpolations are between two fonts and not a sample and a template. Our work also differs in that deformations are applied to the generated samples.

2.2 Boosting and Imbalanced Data Sets

This next group of papers speaks about methods used to change the composition of classes in the training sets. In some cases there are too many samples of one class (the majority class), and two few of another class (the minority class). This might happen in a domain such as the recognition of relatively rare cancer cells among normal cells. In these cases, the authors might attempt to balance out the data sets by adding more of the relatively rare data samples. Alternatively they may remove samples from the majority class. In several of the papers, samples are removed from data sets because they are outliers, outside their true class boundaries. By removing them, better performance can be obtained from the classifier.

Guo, H. and Viktor, H. [GV04a]

Boosting is an ensemble method (ensembles are sets of individually trained classifiers whose predictions are combined to classify new instances) in which the performance of weak classifiers is improved by focusing on *hard* examples, those which

are difficult to classify or examples on which the classifier frequently fails. The output is combined using weighted voting. Boosting concentrates on classifying the hard examples correctly.

In this paper on boosting methods, the authors maintain that boosting frequently suffers from over-emphasizing the hard training examples. In their experiments, they first identify hard examples, and keep the weight of each *seed*, with the seed example to be used for assigning proper weight to the synthetic examples generated. The mean value and variance of attribute A, with respect to class C is determined and the distribution of the values of A is constrained by generating values without changing the original mean value or the variance of the attribute. The mean and variance produce a fixed range of values for the attribute to be assigned to the new sample.

In these experiments, the authors seem to concentrate on the generation of data based on the minority class. They discuss in great detail their method of generating new samples and the precautions they have taken to insure that the mean and variance of the new attributes do not vary from those of the original.

The authors concentrate on generating the new samples in feature space, by varying the attributes of the features of the original samples. They have attempted to insure that the resulting values are similar to the real or original samples by using the mean and variance as constraints. In this way, it is much more likely that these new samples could be found in *nature*. In our work, by way of contrast, we have concentrated on generating the samples in parameter space, and have shown that such samples could actually occur in the real world by generating the actual sample images.

Viktor, H. and Guo, H. [VG04]

In this later work, the authors aim to generate additional synthetic instances to add to the original training set. Training samples are sorted by 'hardness', or difficulty, and then a subset of the minority class is selected as seeds, as well as a subset of

the majority class. The authors define nominal or discrete attributes, and continuous ones. For nominal attributes, they choose N values for each seed in training set i.e., to reflect the distribution of discrete probability means contained in the original training attribute with respect to the particular class. Using continuous attributes, the data generation produces N attribute values chosen by considering the range [min, max] of the original attribute values with respect to the seed class. Before training, the total weights of the examples in different classes are rebalanced, forcing boosting to focus on hard as well as minority class examples. They include tables which show that their method improves performance.

In this work, the authors generate synthetic samples by varying the attributes of the features selected from the training set samples. Minority and majority class examples are generated separately based on the distribution of attributes in the original samples.

Again, this work differs from ours in that the authors create new examples of data in feature space. The results from the first classifier are ranked and new samples are created by selecting certain samples as seeds from which the attributes are selected. These attributes are varied to produce new samples to be used in training the subsequent classifier. Unlike our work, in which interpolation is accomplished smoothly over a range of parameters, each attribute appears to be treated separately from every other attribute with no consideration of how its adjustment might effect the value of the other attributes in a real-life sample. They do not discuss how to prevent a combination of attributes from being generated which could never be found in the real world.

Gou, H. and Viktor, H. [GV04b]

This paper is a continuation of the experiments started in above two works. Using their method, the *hard* examples are used to separately generate synthetic examples for the majority and minority classes. The authors maintain that their results indicate that their approach produces high predictions against both minority and majority classes.

First they separately identify hard examples and generate synthetic examples for the minority as well as the majority classes. Secondly, they generate synthetic examples with bias information toward the hard examples on which the next component classifier in the boosting procedures needs to focus.

In one example, for a Hepatitis data set, they identify 27 hard examples, 2 corresponding to the majority class and 25 from the minority class. The high occurrence of examples from the minority class is due to the fact that, for imbalanced data sets, the minority class is harder to learn. They used a parameter N to specify the amount of synthetically generated examples, and it was set to 100, 300, and 500, respectively. In their experiment, in many cases the improvement for the minority class was quite significant.

They conclude that additional synthetic data provides complementary knowledge for the learning process, and also that rebalancing the class frequencies alleviates the classifiers' learning bias toward the majority class. They state that rebalancing the total weight distribution of different classes forces the boosting algorithm to focus on the hard examples as well as rare examples, and also that the synthetic data prevent boosting from over-emphasizing the hard examples. In this paper, the authors discuss in much more detail their algorithm for generating the synthetic samples and provide examples of the seeds and some of their features.

As before, this work deals with the generation of the synthetic samples in feature space as opposed to sample or parameter space, unlike our work in which we generate the samples by varying the generating parameters.

Mao, J. and Mohiuddin, K.M. [MM97]

These authors examine boosting techniques used in conjunction with synthetic training data to classify characters. They point out that boosting algorithms requires an 'oracle' to produce a large number of independent training patterns. They define boosting and introduce three document degradation models in their boosting algorithm, affine transformations (translation, scaling, rotation and shearing),

a nonlinear deformation (in which they pad the size-normalized character bitmap with a 2-pixel wide boundary and then subject each pixel in the image to a displacement), and a the noise model. In the noise model, a black-white flip occurs independently at each pixel

They run an experiment in which they created three new sets of data for training (27636 characters), validation(12,000) and test(12,000). In an 'epoch', they go through the entire training set in a random order and use either the original pattern (with 'weight' .5) or one of three degradation models (with weight .167 for any of the three). At the end of five epochs, the network is evaluated on the validation data set and the weights corresponding to the best accuracy on the validation set are finally selected. The process is repeated 10 times for each classifier.

In this work, the authors compare the accuracy of different types of classifiers, boosting ensembles, standard ensemble and a single network, and do not perform any testing the whether or not the addition of the of synthetic data is helpful. This paper was interesting to us in that it described the use of synthetic data in a series of experiments designed to test the performance of boosting algorithms.

Japkowicz, N. [JA00]

The class imbalance problem corresponds to domains for which one class is represented by a large number of examples while the other is represented by a few. This is encountered by a large number of domains and can cause a significant bottleneck in the performance attainable by standard learning methods which assume a balanced distribution of classes. This paper discusses the question of whether imbalances are always damaging.

The author compares three methods of re-balancing the classes and investigates which methods work best on what types of imbalances. In the first method, the class represented by small data sets gets oversampled, while in the second, the class represented by the large data set is undersampled. The third method ignores one of

the two classes and uses a recognition-based scheme in place of a discrimination-based one. An artificial domain was created in order to better control the various domain characteristics. The domains were created with one-dimensional inputs in the [0-1] range which were associated with one of two classes.

The best results were obtained by the undersampling method, particularly if the number of training samples was large and the data was not complex. The author maintains that future research should concentrate on finding ways to decrease the complexity of imbalanced domains and re-balance the imbalanced domains even if that means decreasing the overall size of the training set. Oversampling is generally less accurate than random resampling and random downsizing. The results indicate that it is better to learn how to recognize the majority class than the minority one.

While this paper does not directly touch on our research in the area of generation of synthetic data, it does raise some interesting questions regarding the addition of new samples to create balanced data sets. The methods employed in these experiments involve re-sampling the same data points two and sometimes three times. We propose that perhaps better results might be achieved by creating entirely new data points within the minority class.

Sanchez, J. S., Barandela, R., Marques, A. I., Alejo, R. and Badenas, J. [SB02]

This paper describes how to improve the quality of training sets by eliminating mislabeled and atypical samples, or outliers. An outlier is traditionally defined as a prototype that does not follow the same model as the rest of the data. They maintain that a quality training set can be seen as a data set without outliers. Consequently the decision boundaries will be much simpler.

In many practical situations the asymptotic classification error rate of the kNN rule (tending to optimal Bayes as $k \rightarrow \infty$) cannot be achieved because of weakness or imperfections in the training set. The process of cleaning overlapping among classes, and removing outliers is called editing. Koplowitz and Brown (1981) proposed a scheme in which some samples are discarded and some are re-labeled. An

alternative to this is generating new samples to replace the original training set.

The authors describe an algorithm called Depuration, used to remove suspicious samples from the training set and change the class labels of some instances. They use the concept of the nearest centroid which takes into account not only the proximity of prototypes to the given sample, but also their symmetrical distribution around it. They also suggest that future work address the potential of using their editing technique on training sets in which one class is more represented than another.

Although the authors speak about generating new samples to replace samples in the original training set, the do not actually discuss algorithms to do so. They discuss methods to re-label existing samples, in essence moving them from one class to another, They also describe methods they have used to pick samples to be discarded from the training set.

This work differs from ours in that the authors are reducing the training set by eliminating samples near the boundaries, while we are increasing our training set by introducing samples within the boundaries. By using interpolation we insure that our samples fall well within the boundaries of the existing classes.

Jiang, Y. and Zhou, Z. [JZ04]

The authors point out that a kNN classifier is bounded by twice the Bayes error at worst, but this theoretical behavior can hardly be obtained because kNN is sensitive to outliers and noise contained in the training data set. This paper extends the work of the above paper which uses editing approaches such as the Depuration algorithm. This algorithm consists of removing some ‘suspicious’ training samples while changing the class labels of other samples. This is interesting, but risky. The authors propose using neural network ensembles to edit the training data sets for the kNN classifiers, and either removing data samples or relabeling, but not both together.

The paper discusses tests performed comparing the three methods, removal, relabeling, and depuration. It finds that best results are achieved by the removal of suspicious training samples.

In contrast to this study, our work adds samples which are well within the class boundaries. It would be interesting to observe the effects of a combination of adding interpolated samples along with removing outliers.

2.3 Theoretical issues.

The papers in this section address various novel ideas which have influenced our work, some more heavily than others. Several of the papers emphasize the importance of adding more and varied data to training sets, an idea with which we heartily agree. Knuth's paper describing the Metafont system of font generation, in particular, has provided a framework for much of our work.

Lopresti, D and Nagy, G. [LN01]

This paper discusses whether ground-truth is fixed, unique and static, or relative and approximate. The authors say the stochastic models can be used to generate synthetic data for experimentation. For example, appropriately calibrated character defect models can produce multitudes of patterns.

The model is in some sense an idealized representation of the digital object, and one could argue that instead of the object, we model the intent of its maker.

The authors point out some of the intrinsic ambiguities of the ground truthing process. For example, in the University of Washington database (UW1), over 1000 images were made available to the international community. It is interesting to note, that UW1 was oriented towards preventing data entry errors, and making sure there is a single, unique representation for every entity in the dataset. It does not allow for the possibility of multiple legitimate interpretations. There is one master arbiter to reconcile differences in the zoning process. The zoning process involves

the segmentation of a page into logical parts such as columns, titles, etc. This makes the data entirely appropriate for the applications UW1 was intended for, but may not be easily extensible for ground-truthing for more complex document analysis tasks.

In addition, the authors maintain that the effort required to prepare ground truth is almost always the limiting factor on the size of experiments. For instance, in the preparation of the UW1 database the ground truth was carefully constructed using a double data entry, double reconciliation procedure. In the final stage, a single person took the results from the two reconcilers and performed a final reconciliation. Each page required roughly six person-hours to produce.

Some questions they propose are: Is sufficient to have just one version of the ground-truth when the input admits more than one interpretation? Who is to decide whether a ground-truth is correct? What kinds of experiments could confirm that a proffered ground-truth is appropriate for the task at hand? What is the minimal context necessary for a ground-truthing task? Do elements of the ground truth have to be labeled with regard to their importance in some intrinsic hierarchy?

This article directly touches on our work by its emphasis on the difficulty and care necessary to prepare ground truth. If we agree that the more carefully prepared training data we use, the better the classifier performance, then we should be looking for methods of easily constructing reliable training data. One such method might be interpolating between already existing training data to create new, unseen samples.

Simard, P. Y., LeCun, Y. A., Decker, J. S. and Victorri, B. [SV98]

The authors state that classification techniques can be divided into two camps, according to the number of parameters they require – there are the “memory-based” algorithms, which use a compact representation of the training set, and the “learning” techniques, which require adjustments of a comparatively small number of parameters during training.

The main idea in memory-based algorithms is to approximate the surface of possible transforms of a pattern by its tangent plane at the pattern. With learning-based algorithms, rather than trying to keep a representation of the training set, it is possible to compute the classification function by learning a set of parameters from the finite training set.

With tangent distance, two sets of curves are constructed representing sets of points obtained by applying some chosen transformations on the samples. Tangent spaces are then constructed for the curves. Assuming that the working space has more dimensions than the number of chosen transformations then the tangent spaces do not intersect and the tangent distance is uniquely defined. Finding the minimum distance between the tangent spaces is then a simple least squares problem.

The authors find that the best strategy is to extract features first, smooth the features, and then compute the tangent distance on the smoothed features. They keep a pool of all the prototypes which could potentially be the k nearest neighbors of the unclassified pattern. Next, the distance *D1* between all the samples in the pool and the unclassified pattern is calculated along with a classification and a confidence score. If the confidence is good enough, say better than *C1*, the classification is complete; otherwise the k closest samples are kept while the remaining samples are discarded. The reduced pool is passed to the next stage. In tangent propagation, the invariance is incorporated directly into a classification function.

They conclude that the main reason for the success of their methods is the ability to incorporate a priori knowledge into the distance measure. A smart preprocessing enables them to measure the Tangent Distance in a more appropriate feature space instead of the original pixel space. Additionally, Tangent Vectors and the prototypes can be learned from the training data rather than chosen a priori. It is straightforward to derive a batch or on-line algorithm to train the tangent vectors.

The main advantage of Tangent Distance is that it is a modification of a standard distance measure to allow it to incorporate a priori knowledge that is specific to the

problem. However, the two drawbacks of tangent distance are its memory and computational requirements. The authors state that tangent propagation can be used for learning - instead of learning a classification function from examples of its values; one can use information about its derivatives. To a first approximation, using tangent distance or tangent propagation is like having a much larger database. If the database were plenty large to begin with, tangent distance or tangent propagation would not improve the performance.

This novel approach to classification differs from our approach in that they use the existing samples and a priori knowledge to find the closest match. They liken their algorithm to having a much larger database, unlike our approach in which we actually enlarge the variety of training samples which is available to the classifier. The authors admit that their approach has the disadvantage of requiring much memory and computational power.

Smith, S. J., Bourgoin, M. O., Sims, K. and Voorhees, H. L. [SV94]

This article addresses the notion that statistical techniques may achieve superior performance on a wide range of perceptual tasks, compared to the artificial intelligence (AI) approaches. Such methods include gradient descent search for relative weightings of penstroke and other features.

The authors discuss a conference which was sponsored by the National Institute of Standards and Technology (NIST) at which twenty-nine groups came together to compare the performance of their OCR systems on a common set of segmented handwritten characters. The results showed that in general, systems that were trained only on the Census database had poorer performance than those trained on data sets which incorporated large additional databases. However, their system, which used the kNN algorithm and three metrics, Hamming (counts of number of mismatched pixels), pixel distance, and a penstroke feature metric, performed surprisingly well.

Additionally, the authors found that in their experiments, for every tenfold increase

in database size the error rate was cut by half or more, although the performance seemed to be leveling off slightly for larger database sizes. They point out that if the database is large enough it will include all possible samples and will only fail to perform perfectly due to errors or noise in the database. They postulate that there is good reason to believe that performance will continue to improve as the training database grows even larger, suggesting that researchers might better spend their time collecting data than writing code.

In our research we increase the variety of data available to the classifier as opposed to just adding more of the same data. In this way we include more samples that may not have been previously found in the real world data. Although it is beneficial to increase the database size, it is also important to include a large variety of samples, a task which is made easier with interpolation in parameter space.

Hofstadter, D. and McGraw, G. [HM93]

Hofstadter's Letter Spirit cognitive science project attempts to understand the creative process of artistic letter-design; how the 26 lowercase letters of the roman alphabet can be rendered in different but internally coherent styles. The authors started with *seed letters* and attempted to generate the rest of the alphabet automatically in such a way that all letters share the same style or *spirit*.

The authors insist that for a design algorithm to be called creative, it must 1) make its own decisions, 2) have rich knowledge, 3) have concepts and their interrelations which are not static, 4) must perform at a deep conceptual level, 5) be able to judge its own output, and lastly, 6) converge iteratively on a satisfactory solution.

In this paper, the term *letter-concept* is used to refer to the most abstract idea for drawing a letter. Next, a *letter-plan* or specification of how each role should be realized is drawn up. Lastly, the *letterform*, or actual shape is drawn upon the paper. The conceptual pieces into which a letter is broken in the mind's eye are its *roles*. The stylistic appropriateness of a shape is judged in terms of how the roles are filled, or in other words, how norms are violated. They ask how letters in a given

style are related to one another and discuss the concepts of *categorical sameness* (possessed by instances of a single letter in various styles), and *stylistic sameness* (possessed by instances of various letters in a single style).

In this work, the authors speak of having generated 600 gridfonts so far. The process of gridfont creation is governed by the interrelatedness of letter categories. Modeling the ability to find conflicts, diagnose them, and convert the diagnoses into reasonable suggestions for solutions is a key aspect of the project. The authors maintain that this temporally-extended serial process of integration and gradual tightening of internal consistency is an indispensable part of true creativity.

The work of Letter Spirit is carried out by four program modules: (1) the *Imaginer*, whose job is to make suggestions regarding roles, (2) the *Drafter*, which attempts to implement the roles concretely, (3) the *Examiner* which takes the grid-letter specification and determines which of the 26 letter-categories it belongs to, and (4) the *Adjudicator* which is concerned with stylistic consistency. We direct the reader to the referenced work for the details of the implementation.

The authors state that a system which can recognize letters in many typefaces that it has never seen or been trained on has yet to be developed. In an effort to address this problem, we attempt to create just such a system, one which is both safe and effective, by using existing fonts to create interpolated samples which have never been seen.

Knuth, D. E. [KN86]

Donald Knuth developed virtual fonts to assist those who had been struggling with interfaces between differing font conventions. His *Metafont* is a way to specify a mapping from TEX's notion of a font character to a device's capabilities for printing. Knuth describes 62 parameters, which divide naturally into several groups, that define a Computer Modern typeface.

Among the parameters are those which define vertical measurements, overshoot, or how much a character descends or ascends above the height, one parameter which

defines width, and four additional parameters for finer adjustments. There are also parameters which define the darkness or *weight,* for both upper and lower case, stem corrections, and the *softness* of corners in letter shapes. Serifs and arms and diagonal strokes can also be varied by other parameters.

In addition, there are numeric parameters to describe characteristics such as slant and bowl thickness, as well as true/false parameters such as square_dots and low_asterisk.

Knuth states that many of these parameters depend on each other in subtle ways so that you cannot obtain a good font by selecting 62 values at random. Proper balance between parameters such as *curve* and *stem* is very important. In fact certain minimum conditions must be satisfied or the resultant font will not be pleasing. For example, the *asc_height* must always be larger than the *x_height* and stem weights must not be less than the corner diameters. The parameters for fine adjustments should be very small. Additionally, he points out that the program will fail at very low resolutions, or those with fewer than 100 pixels per inch.

In our work, we use Knuth's Metafont system to create new fonts by interpolation. Proper balance is maintained among the parameters, because we smoothly interpolate among the parameter values. For example, if the *asc_height* is larger than the *x_height* in the starting fonts, they will continue to be so in the interpolations, as they are increased or decreased proportionately to each other. In this fashion, we create new fonts which have never been seen in the real world, but shich are still highly legible.

2.4 Other Relevant Issues

The next two papers deal with issues which are relevant to our topic. The first of the papers, while not directly addressing classifiers, gives some broad guidelines for software testing and test design in general. The second paper in this section deals directly with the question of which statistical test is appropriate to our experiment and how to design an experiment to employ our chosen statistic.

Zhu, H., Hall, P. and May, J. [ZM97]

Software testing has been formalized over the years in an effort to provide consistency and structure for deciding when a program is correct. We believe that the same sort of rigor in defining training samples would be useful. Questions such as "When does a training set provide adequate coverage?" "What sort of data should be included in a training set?" and "When can you trust a training strategy?" should be formalized so that consistent answers can be expected.

In this paper, the authors address the question of "what is a test criterion?". They define several types of testing as follows:

- Statement coverage: In software testing practice, testers are often required to generate test cases to execute every statement in the program at least once. A test case is an input on which the program under test is executed during testing. A test set is a set of test cases for testing a program. The requirement of executing all the statements in the program under test is an adequacy criterion. The percentage of the statements exercised by testing is a measurement of the adequacy.
- Branch coverage criterion requires that all control transfers in the program under test are exercised during testing.
- Path coverage requires that all execution paths from the program's entry to its exit are executed during testing.
- Mutation adequacy plants some artificial faults into the program to check if they are detected by the test. If a "mutant" (program with a planted fault) and the original program produce different outputs on at least one test case, the fault is detected, and the mutant is "dead". The percentage of dead mutants compared to the number of mutants that are not equivalent to the original program is called the mutation score, or mutation adequacy.

Additional terms defined:

Reliability requires that a test criterion always produce consistent test results. Validity requires that the test always produce a meaningful approach. However, it was soon recognized that there is no computable criterion that satisfies the two requirements and hence they are not practically applicable.

A test data adequacy criterion is considered to be a stopping rule that determines whether sufficient testing has been done that it can be stopped.

- Following a set of guidelines, one can produce a set of test cases by an algorithm which generates a test set from the software under test and its own specification. Such an algorithm may involve random sampling among many adequate test sets. Mathematically speaking, test case selection criteria are generators, that is, functions that produce a class of test sets from the program under test and the specification.
- The second role that an adequacy criterion plays is to determine the observations that should be made during the testing process. Before the testing, the objectives of that testing should be known and agreed upon and set in terms that can be measured.

Categories of test data adequacy criteria:

1. Specification based – specifies required testing in terms of identified features of the specification
2. Program-based, specify test requirements in terms of the program under test.
3. Random or statistical testing (select according to the usage of the software) in which test cases are sampled at random according to a probability distribution over the input space.

The author discuss three basic approaches to testing:

- Structural (coverage of a particular set of elements – program or specification), fault-based (focus on detecting faults in the software) and error-based

(check error-prone points). In structural testing, a test set is said to satisfy the decision coverage criterion if for every condition there is at least one test case such that the condition has value true when evaluated, and there is also at least one test case such that the condition has value false. Criteria can be redefined to obtain finite applicability by only requiring the coverage of feasible elements. There are two main roles a specification can play in software testing; to provide the necessary information to check whether the output of the program is correct, and also to provide information to select test case and to measure test adequacy. It is important to remember that only a finite subset of the paths can be checked during testing. The problem is therefore to choose which paths should be exercised.

- In fault based testing, error seeding is a method by which artificial faults are introduced into the program under test in some suitable random fashion unknown to the tester. This can show weakness in the testing. The first step in mutation analysis is the construction of a collection of alternative programs that differ from the original in some fashion. Each "mutant" is then executed on each member of the test set, stopping either when an element of the test set is found which on which the mutant(s) and the program produce different responses, or when the test set is exhausted. If a large proportion of the mutants live (i.e. do not stop) then it is clear that on the basis of the test data alone, we have no more reason to believe that the program is correct than to believe that any of the live mutants are correct. Mutation analysis systematically and automatically generates a large number of mutants. Measuring the adequacy of software testing by mutation analysis is expensive. In perturbation testing, we are concerned with possible functional differences between the program under test and the hypothetical correct program. The adequacy of a test set is decided by its ability to limit the error space defined in terms of a set of functions.
- In error-based adequacy criteria, the software input space is partitioned either according to the program or the specifications. One method, NX1 domain

testing requires N test cases to be selected on the borders in an N-dimensional space, and one test case just off the border. A stricter criterion is the N X N criterion, which requires N test cases off the border. The focal point of boundary analysis is to test if the borders of a subdomain are correct. N X 1 will detect an error of parallel shift of the border, while N X N will detect parallel shift and rotation of linear borders. (see algorithm in article)

Fault detecting ability is one of the most direct measures of the effectiveness of test adequacy criteria. The methods to compare test adequacy criteria are (1) statistical experiment, (2) simulation, and (3) formal analysis. Duran and Ntafos compared 100 simulated random test cases to 50 simulated partition test cases, and concluded random testing was superior. Performing 100 random tests was less expensive that 50 partition tests. However, confidence is more difficult to achieve for random testing than for partition testing, which they showed by computing the upper bounds in the two cases.

In the design of our experiments we have attempted to adhere to the practices defined above. We have made our system reliable and insured that our test sets produce consistent results (that is to say, every time our classifier is tested on the same test set using the same training set, the results are consistent and repeatable). The objectives of our testing were known and agreed upon prior to testing, and were measurable and identifiable.

Dietterich, T. G. [DI98]

In this example the authors consider the single-domain case in which the primary goal is usually to find the best classifier and estimate its accuracy on future examples. In any particular application, the goal is usually to choose the best classifier from some set of available classifiers.

If we have a large set of data, then one can set some of it aside to serve as a test set for evaluating classifiers and much simpler statistical methods can be applied

in this case. However, in most situations, the amount of data is limited, thus, one needs to use it all as input to the learning algorithm and some form of resampling (cross-validation or bootstrap) must be used to perform the statistical analysis.

The authors make the assumption that all data points are drawn independently from a fixed probability distribution defined by the particular application problem

The authors pose 9 statistical questions, as follows

1. Suppose we are given a large sample of data and a classifier C. The classifier C may have been constructed using part of the data, but there is enough data remaining for a separate test set. Hence, we can measure the accuracy of C on the test set and construct a binomial confidence interval.
2. Suppose we are given a small data set S and suppose we apply algorithm A to S to construct classifier C, how accurately will C classify new examples?
3. Given two classifiers Ca and Cb and enough data for a separate test set, determine which classifier will be more accurate on new test examples.
4. Given two classifiers, Ca and Cb produced by feeding a small data set S to two learning algorithms, A and B, which classifier will be more accurate in classifying new examples?
5. Given a learning algorithm A and a large set of data, what is the accuracy of the classifiers produced by A when trained on new training sets of a specified size?
6. Given a learning algorithm A and a small data set S what is the accuracy of the classifiers produced by A when A is trained on new training sets of the same size as S?
7. Given two learning algorithms A and B and a large data set S, which algorithm will produce more accurate classifiers when trained on data sets of a specified size m?

8. Given two learning algorithms A and B and a small data set S, which algorithm will produce more accurate classifiers when trained on data sets of the same size as S?
9. Given two learning algorithms A and B and data sets from several domains, which algorithm will produce more accurate classifiers when trained on examples from new domains?

The authors state that questions 1, 2, 5 and 6 can all be rephrased in terms of determining the expected log loss of a classifier or algorithm, while questions 3, 4, 7 and 8 can be rephrased in terms of determining which predictor or algorithm has the smaller mean squared error. Question 1 can be addressed by constructing a confidence interval based on the normal or t distribution (depending on the size of the set). Question 3 can be addressed by constructing a confidence interval for the expected difference. Analysis of variance techniques have been developed for questions 5 and 7, however appropriate statistical tests are not well established for the small sample questions (2, 4, 6 and 8).

To design and evaluate statistical tests, the authors maintain that the first step is to identify the sources of variation that must be controlled by each test. For the case they are considering, there are 4 sources of variation

1. First there is random variation in the selection of the test data that is used to evaluate the learning algorithms. On any particular randomly-drawn test data set, one classifier may outperform another, even though on the whole population the two would perform identically. This is particularly a problem for small test sets.
2. Selection of the training data is the second source of random variation. On any particular randomly-drawn training data set, one classifier may outperform another, even though on the average, the two algorithms have the same accuracy. Even small changes to the training set may cause large changes in the classifier produced by a learning algorithm.

3. There may be internal randomness in the learning algorithm, for example, random starting weights in forward-feed neural networks.
4. Lastly, there may be random classification error. If a fixed fraction, n of the test data points are randomly mislabeled, then no learning algorithm can achieve an error rate of less than n.

A good statistical test should conclude that two algorithms are different if and only if their percentage of correct classifications would be different, on the average, when trained on a training set of a given fixed size and tested on all data points in the populations.

The paper goes on to describe 5 statistical tests bearing on question 8 (Given two learning algorithms and a small data set *S*, which algorithm will produce more accurate classifiers when trained on data sets of the same size as *S*?) The tests are as follows:

1. McNemar's test
2. a test for the difference of two proportions
3. the resampled t test
4. cross-validated t test
5. 5x2cv test or 5-fold cross validation

A simulation study is performed to measure the probability that each will incorrectly detect a difference when there is no difference (Type I error). In an applixation setting, a sample *S* is drawn randomly from *X* according to a fixed probability distribution *D*. A collection of training examples is constructed by labeling each $x \in S$ according to *f(x)*. A learning algorithm *A* takes as input a set of training examples *R* and outputs classifier *f*. The true error rate of that classifier is the probability that *f* will misclassify an example drawn randomly from *X* according to *D*. In practice this error rate is estimated by taking sample *S* and subdividing it into a training set *R* and a test set *T*. The error rate of *f* on *T* provides an estimate of the true error rate of *f* on population *X*.

The null hypothesis is that for a randomly drawn training set R of fixed size, the two algorithms will have the same error rate on a test example randomly drawn from X, where all random draws are made according to distribution D.

The authors found that the Type I error of the resampled t test was unacceptably bad and the test was expensive computationally so they did not study it further. The difference-of-proportions test also had high type I error, which would be unacceptable in some cases, but it is cheap to evaluate so they retained it for further study.

To obtain a better evaluation of the four remaining tests, the authors next conducted a set of experiments using real learning algorithms on realistic data sets. They tested question 8 above.

The authors found that McNemar's test, the cross-validated t test and the 5x2cv test all had acceptable levels of Type I error. The differences- of-proportions test had the lowest Type I error.

However, if the goal is to detect whether there is a difference between two learning algorithms, then the *power*, or the probability that a statistical test will reject the null hypothesis when it is false, is also important. The authors found that the cross-validated t test was the most powerful followed by the 5x2cv test. They suggest that if the goal is to be confident that there is no difference between two algorithms, then the cross-validated t test is the best choice, even though its Type I error is unacceptable.

Each of the statistical tests also has other shortcomings. For example, the derivation of the 5x2cv test requires a large number of independence assumptions that are known to be violated. McNemar's and the difference-of-proportions tests do not measure all the important sources of variation. Because of this, the authors suggest that all the statistical tests should be viewed as approximate, heuristic tests, and not as rigorously correct statistical methods. Based on their results, the authors recommend either the 5x2cv or McNemar's test for situations in which the learning

algorithm can only be run once. They conclude that the resampled t, which is currently the most popular test, should never be used.

After careful consideration of this article, we discarded the resampled t test as our design. We initially chose McNemar's test because of its low Type I error. Our goal was, in fact, to detect if there is a difference between our two classifiers, and thus power was important. McNemar's test is not as powerful as some of the other tests, but we felt that it was adequate for our experiments. As will be seen later, we actually found that the χ^2 was sufficient for our purposes, and simpler to implement.

Chapter 3

Questions We Explored

We make some claims about testing regarding reliability and validity. We explored what is the criterion for success, for example, we say that “No failure rate gets worse” using our interpolated data.

We have also said that a synthetic image may occur, and there is no compelling argument that it cannot occur. However, we are also merely guessing how often it occurs and in fact it may never occur. If so, then we have trained on samples not drawn from the underlying distribution. Additionally, what if the trainable classifier is sensitive to the balance between real and synthetic samples in that the more synthetic samples are used, the less well it will perform on real samples? [1]

One of the problems is that no one can agree on the parameters to use, and no one can collect a truly representative set of samples. We have attempted to craft an airtight argument that such generated images represent the underlying distribution and thus our classifier ought to handle them.

George Nagy has objected to the use of training data that are highly correlated, i.e. duplicative, and one question we took into account is whether the training data

[1] And what, ultimately, is the difference between ”real” and ”synthetic” data? Is your data real while mine is synthetic? Suppose the history of collection of a data set is lost: if it were originally synthetic, now mere ignorance of that fact makes it real.

were independent. To address this, we used random variables in the generation of our interpolated samples. Our efforts reflect an attempt to model the behavior of a sequence of independent samples drawn from an underlying, incompletely understood distribution.

This thesis attempts to address the following serious methodological problem in supervised classification: It is highly problematic and often unfeasibly expensive to collect and assign ground truth to a sufficiently large representative set of training data.

The approach this thesis takes is as follows. Given a set of real data samples, we automatically synthesize new samples complete with ground truth that are guaranteed also to be representative. We have attempted to make precise our claim that the resulting synthetic data are representative. While we do not claim that every synthetic datum must naturally occur, we do argue that it should be classified correctly as if it could have occurred.

Issues we have explored include

- Formalizing a methodology for the generation and use of synthetic data in training and testing classifiers.
    1. What makes a synthetic datum "safe" for use in training? When is classification improved by adding it to the training set?
    2. We have explored methods for training on synthetic data that is guaranteed never to increase confusion between any two categories.
    3. We have researched the use of parameter space in the generation of interpolated data by running a series of carefully crafted experiments.
        (a) Given two points in parameter space we explored if it always safe to interpolate between them (*i.e.* by forming convex combinations)?
        (b) Generalizing to a set of more than two points, we tested when it is it safe to generate convex combinations within the set?

4. We have investigated Knuth's Metafont system for generating typefaces as a concrete example of a richly parameterized model for shape. We have addressed the following questions.
   (a) Is it possible to train a classifier that will correctly distinguish characters, say *e* and *c*, across *all typefaces* within the Metafont system, while using only synthetic data?
   (b) We have investigated uses of Baird's document image degradation model together with Metafont to provide a generating model for Latin alphabet shapes over a wide range of image characters.
5. Note that this thesis is not about the design of classifiers, so we have just picked one that seems as good as any other, the nearest neighbor classifiers.
6. We have exerted ourselves to unearth evidence that under certain circumstances a classifier trained on synthetic data can produce results superior to one trained only on real data.

Chapter 4

A Small Scale Proof of Principle Experiment

We devised a small experiment to investigate if there is some minimum “seed” size of real data training points which, when interpolated in feature space, will give improved results. Our hypothesis was that classifier performance of the synthetic training data would increase as the seed size increased, however, at some point it would improve less rapidly until it approached the performance of the real data.

To test this hypothesis, we created an experiment in which the data to be classified consisted of small image files labeled as “reddish”, “blueish” or “greenish” by an outside observer. The image files were of one color apiece, each one having the three color components, (R,G,B) which ranged from zero to one. That is to say, each pixel in the image file had the same RGB value.

The experiment was designed to produce approximately 100 samples per color class (reddish, blueish, or greenish), which were randomly divided into a training and a test set. Much variation was introduced in the experiment, as the colors had a broad range in hue, intensity and saturation, and many of the colors were in fact not actually red, blue or green, but yellow, purple, etc. It was up to the observer to pick the closest of the three choices.

The steps of the experiment follow:

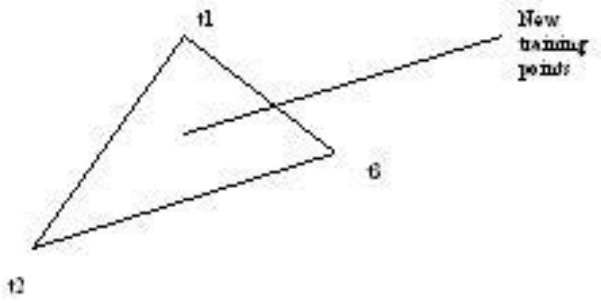
- A human observer labeled each of 300 image files, or "swatches", as either "reddish", "blueish", or "greenish". Each color swatch was displayed as a rectangle of a single color having its' red, green and blue components (real numbers ranging from 0 to 1) chosen at random. The observer's answer was recorded along with the color component values for each image. The end result of this step was a list of samples consisting of three real numbers (red, green, and blue), and a choice of "reddish", "blueish", or "greenish" (RG or B character).
- The image files were divided randomly into two groups, a training set and a test set. Each set had at least 150 samples with approximately 1/3 reddish, 1/3 blueish, and 1/3 greenish.
- The classifier was trained on the previously labeled training set, and tested using a "Nearest Neighbors" classifier on the test set for three features defined as the ratio of red/(red+green+blue), green/(red+green+blue) and blue/(red+green+blue). The results were recorded. This step resulted in an error rate of .144.
- A "seed" sample of the training data was chosen at random, for example three samples each from among the training points labeled as reddish, blueish, and greenish. Training data was generated by interpolation among this seed data using the following algorithm.

Steps of Experiment

1. For each color type (R,G and B), choose n seed points
2. For each of the n seed points of that color, randomly choose n weights w(i) which sum to 1
3. Multiply each seed point, s(i) by its weight w(i) using standard linear algebra
4. Calculate the new point, t, by summing s(i)w(i) over the n points
5. Repeat steps 2-4 to obtain each training points needed for each color

6. Choose n more seed points for the next color and repeat steps 2-5

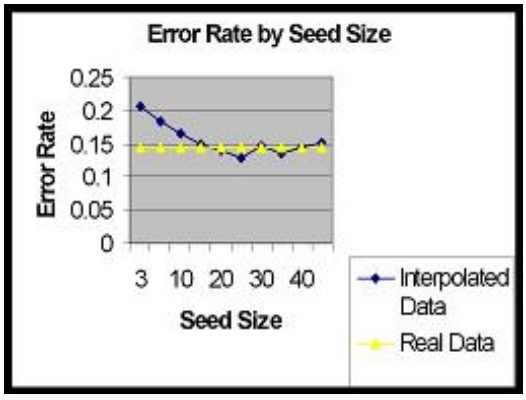
Figure 4.1: Training Points with Convex Space



- Since the weights sum to one, this creates a convex space enclosed by the training points.
- Enough training data was generated to arrive back at 50 training samples. For example, three samples for the “seeds” the program generated 47 training samples per color class.
- The classifier was again used on the test set using the interpolated data as training samples instead of the real data. The experiment was repeated ten times apiece for specific seed sizes ranging from three to 45 with the error rates averaged by seed size.
- An additional experiment was conducted using 10 seed samples to generate training sample sets ranging between 20 and 200. Once again the training sets were tested 10 times apiece and the resulting error rates averaged by training set size. Charts from both experiments are shown and discussed below.

Results (number of misclassified test samples) by “seed size” are shown on the chart below.

Figure 4.2: Error Rate by Seed Size

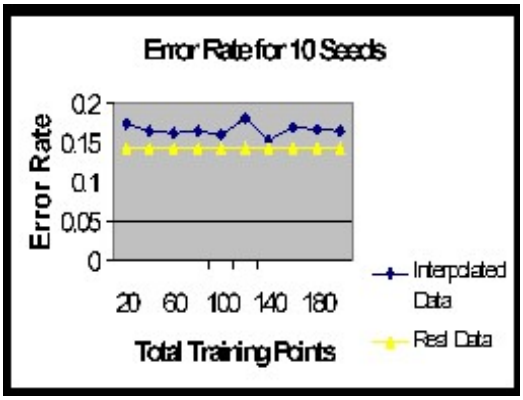


It is interesting to note that the error rate trended downward from when going from three to 15 seeds. The error rate more or less levels off at 25.

We had hypothesized that the error rate of the interpolated data would approach the true error rate, but never reach it, and indeed, the fact that it appears to dip below the line may be merely attributable to the fact that we happened to randomly choose 25 good test points for our ten runs of that seed size. Further experimentation should show whether or not the error rate truly drops below that of the real data at some point.

Results from our experiment with different training set sizes generated from 10 seed points are shown in the chart below.

Figure 4.3: Error Rate for 10 Seeds



In this experiment, we had expected that as the size of the training set increased, the error rate would be driven down. Surprisingly, this did not happen, nor did it happen when the experiment was conducted with 20 seeds.

Perhaps these results may have occurred because of the subjective nature of the data classification. Some of the data points may have been very far apart, due to the fact that some colors such as gray had to be labeled as either “reddish”, “blueish” or “greenish”, when in fact they might conceivably have been any of the three. This might have led to several training points very close to or actually outside of the edges of the selection boundaries, with labels that were somewhat suspect. If these points were chosen for interpolation, conceivably the errors could be actually magnified by the generation of a larger training set. It thus might be preferable to cull the training points close to the edges of the decision areas before applying our interpolation algorithm.

Conclusion

We do believe that the results, particularly from the first experiment are quite

promising. In particular, the experiment shows that it is possible, at least in a limited feature space, to interpolate data from real data, thus increasing the size of the training data. The error rate from the interpolated data was not extremely greater than that of the real data, and at times even appeared to fall below it.

While it was somewhat of a surprise that the error rate did not fall off with the generation of a larger training set, we believe that more experimentation would lead to a better sense of why that did not occur, and perhaps lead to better rules to apply before generating the data.

We direct your attention to the following page, in which we include sample confusion matrices for the interpolated data (sample individual test runs). In each table, the first row of numbers shows how many of the red samples were classified as red, green or blue. The second row shows how many of the green samples were classified as red, green or blue, while the last row shows how many of the blue samples were classified as red, green and blue. Correctly classified samples are counted in the (0,0), (1,1), and (2,2) cells which form the diagonal of the confusion matrix..

Table 1.

CONFUSION TABLE

Nearest Neighbor Results for

”Real Training Data”

Classified As:	R	G	B	Error Type II
True Class				
R	35	5	3	8
G	4	47	4	8
B	6	1	55	7
Error Type I	10	6	7	23

Error Rate 0.144

Table 2.

CONFUSION TABLE

Nearest Neighbor Results for

25 randomly picked training points

Classified As:	R	G	B	Error Type II
True Class				
R	32	5	6	11
G	3	47	5	8
B	4	2	56	6
Error Type I	7	7	11	25

Error Rate 0.156

Table 3.

CONFUSION TABLE

Nearest Neighbor Results for

3 randomly picked training points

Classified As:	R	G	B	Error Type II
True Class				
R	26	5	12	17
G	3	50	2	5
B	2	4	56	6
Error Type I	5	9	14	28

Error Rate 0.175

Chapter 5

Design of a Family of Experiments

5.1 Purpose

We were interested in seeing whether or not interpolation in parameter space was safe and effective in the context of supervised classification. To the best of our knowledge, interpolation has never even been attempted in parameter space. Our research uncovered several examples of interpolation in sample space, in which new samples were created by altering already existing images of characters. We also found examples of interpolation in feature space, in which certain features of samples were measured to create feature points, after which new features points were created by interpolating between the values of these feature points. One of the ideas we wanted to test was whether or not it was possible or practical to create interpolations in parameter space, by altering the generating parameters of our samples.

Once we had created such samples, we then wanted to test whether an interpolated training set created in parameter space is safe and effective in classifying samples both created from the original parameters as well as new samples created from our interpolated parameters.

We chose Knuth's Metafont system as a starting point in the creation of our interpolated

samples. Since the Metafont system is parameter-driven, in that it uses 62 different parameters to programmatically create a font, we felt it would provide a rich framework for generating our interpolated samples. By interpolating between specific Metafont parameters, for two or more fonts, we would be able to create new, previously unseen, fonts. However, not all of the Metafont fonts use every parameter, so it was important to choose carefully so that each of our starting fonts used the same set of parameters. For our series of experiments, all the starting fonts were from the Computer Modern family of fonts.

Once we saw that we were able to actually create interpolated fonts we tested whether samples created by these interpolations were safe and effective by performing a series of experiments. In describing our experiments the first, or starting, typefaces will be referred to as *pure*. These are well known, standard typefaces created from original Knuth's Metafont type styles which are widely used. Existing classifiers have been trained on them. The next set of typefaces we call *interpolated*. These typefaces have been created by interpolating between the parameters used to create the pure typefaces. They may never have been used but are legible and could be used.

First we tested whether a classifier trained on a set of images generated from interpolated typeface styles performed as well as a classifier trained on a same-size set of images generated from pure type styles when tested on pure test samples. It is important to note that the tests were performed with training sets of the same size. In this way improvements in performance could not be attributed merely to the fact that more training samples were involved. This tested the safety of our algorithm. We wanted to be sure that we did not lose accuracy by the introduction of the interpolated test samples.

Next we wanted to see whether our classifier trained on the set of interpolated training images performed better than the classifier trained on pure styles when tested on interpolated samples. Again, it is important to note that the training sets were the same size for each of the tests. This tested the efficacy of our algorithm when tested on images previously unseen among our test samples and not among the traditional fonts, but which could possibly occur.

Our purpose in designing this family of experiments involving pure and interpolated type styles is thus three-fold. First we wanted to discover if it is possible to create interpolations in parameter space. Secondly we wished to explore if the use of these interpolations is safe, at the very least in a controlled set of experiments. And thirdly, we wanted to determine if there is at least one instance of circumstances in which the use of samples created with the interpolated parameters leads to better performance.

The first three sets of experiments involved interpolations between two typefaces, while the remaining set of experiments implemented an interpolation among three typefaces. The first experiment used interpolations between CMR (Computer Modern Roman) and CMSS (Computer Modern Sans Serif), testing the letters *e* and *c*. The letters e and c were created using Knuth's Metafont with the fonts CMR (Computer Modern Roman) and CMSS (Computer Modern Sans Serif).

Nine interpolations between CMR and CMSS were constructed by smoothly interpolating the parameters used in creating both CMR and CMSS. Below is an example showing the letters in CMR, nine interpolations, and finally CMSS. Each of the nine interpolations uses successively more CMSS than CMR. For instance, the second character is 90% CMR and 10% CMSS, while the tenth character is 10% CMR and 90% CMSS. The letters thus vary smoothly from CMR to CMSS.

The second set of experiment tested a similar interpolation between CMR and CMFF (Computer Modern Funny Font), also on the letters *c* and *e*. The third set tested the letters *i* and *j* with the CMR and CMFF fonts, while the fourth set of experiments tested a three-way interpolation between CMR, CMFF, and CMSSI (Computer Modern Sans Serif Italic) fonts on the letters *c* and *e*.

Each set of experiments had approximately 12 tests involving differing degrees of blur, intensity, and brightness. Each test compared the performance of pure versus mixed (interpolated) training sets on both pure and mixed test sets. Some of the tests in each set

Figure 5.1: Letters e and c and their Interpolations

took test data from within the entire range of interpolated samples, while some took all the interpolated data from the midpoint between the two fonts. We have included seven of the most interesting tests from each set of experiments along with their results.

5.2 Description of the Automation

5.2.1 Font Creation

A Metafont program was created to interpolate between two or more fonts from the Computer Modern Roman families of fonts. I will discuss the case of a two-font interpolation. First, the parameter values were listed for each font and differences between the same parameter in each font were calculated. Since we were creating nine interpolations, the difference was divided by 10 and one tenth of this value was added to or subtracted from the first font until we arrived at the second font. Boolean values were either True or False. One example of a Boolean parameter would be the Serifs parameter (Serifs = True or Serifs = False). If both fonts had True, the interpolations were also True. If one was False

and one was True, the first half of the interpolations was set to False, while the second half were set to True and vice versa. A Metafont program was used to create the fonts and once they were created, LaTeX was used to create an image of the letters c and e (or i and j) as a .png file.

5.2.2 Test Set Creation

The IDMGEN [BA92] program created by Henry Baird as an image defect model generator was used to generate the images of a letter in a selected font. The program is given an image of an isolated character in PNG format, from which it generates a series of the same character in text-line format in an ascii file. The text characters are pseudo-randomly distorted using a quantitative model of the printing and imaging process. Alternatively the program will read/write HSLC, an enhanced HSL format developed by the DICE project at Lehigh. The details of this file format are beyond the scope of this dissertation, however the interested reader is referred to the website at snake-eyes.cse.lehigh.edu for details. The HSLC files are easily converted to PNG files for better display of the character images by using the DICE project utilities.

A parameter for seed (-S) is input at the start of each letter generation run to set the pseudo-random number generator and produce an image of a character. Input parameters are used to generate scalar random variables with defined distributions which are applied to the generated images. The interested reader may find the details of these parameters in the article referenced above. We briefly summarize below three of the parameters we used in our experiment to generate our images.

- BLURRING: This represents the point-spread (or impulse response) function of the combined printing and imaging process. In the IDMGEN implementation it is modeled as a circularly-symmetric Gaussian filter with a standard error of /textit-blur in units of output pixel size. The value of blur is passed to the program as the -e parameter, with its first argument being the blurring value, while the second argument is the variance from the blurring value during the pseudo-random generation process.

(default: -e.7,.9; Values greater than .7 will therefore produce more blurred characters).

- THRESHOLD: This parameter models binarization as a test on each pixel: if the pixel intensity is greater than the threshold, the pixel is black. The threshold parameter is passed into the IDMGEN program as the -t parameter. Again, its first argument is the threshold value while the second is the variance.

  (default: -t.25,.125; This default is halfway between the expected intensities at a black pixel just off center of a 1-pixel-wide line, and the white pixel next to it).

- SENSITIVITY: The sensitivity parameter randomizes each pixel's photo-receptor sensitivity in two stages. For each pixel, a sensitivity adjustment is chosen randomly, distributed normally, with mean 0 and standard error equal to the parameter, and then added to each pixel's intensity or brightness before the threshold test. The sensitivity parameter is passed to IDMGEN as the -s parameter. As before, the first argument is the sensitivity value while the second is its variance.

  (default: -s.125,.125)

Our end result was a series of images with variations in blur, brightness and intensity, some very readable, and some greatly distorted. Each of the experiments within a series of fonts used different values of -s, -t, and -e to control its sensitivity, threshold and blurring. The greater the values for each of the first parameters, the more potentially distorted the characters were. The greater the value of the second variable for each parameter, the larger the variance from the ideal character for each individual image.

.

5.2.3 Automation of Results

For each experiment, test results were placed in text files in experimental subdirectories. A single shell script was created which used multiple TeX file templates, graphics, textual experiment descriptions, and the test results themselves to generate a formatted four or

five page summary of the experimental results into a final TeX file. All the statistical tests were automatically performed and evaluated in this process as well.

5.3 Steps of the Experiment

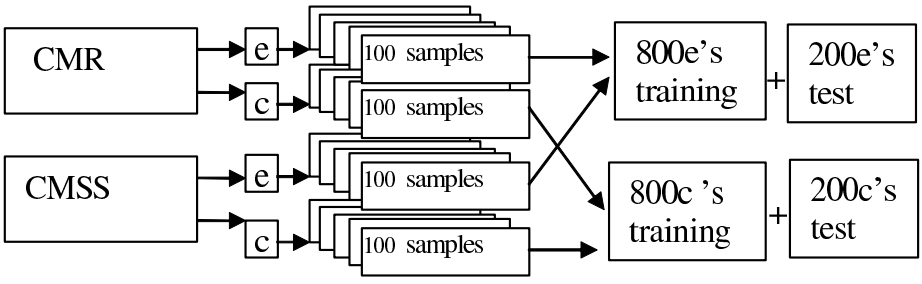
The first set of experiments is described in detail below while the remaining sets of experiments follow similar steps. The experiments differ in the amount of blur, distortion, variance from the ideal pattern, letters chosen, fonts interpolated, and various other ways.

5.3.1 Step 1:

Four hundred samples of the letter c and four hundred samples of the letter e were generated using the IDMGEN program described above. For the first group of experiments, the ideal prototype of each of the training samples was a machine print type form of these letters in Knuth's CMR (Computer Modern Roman) typeface (800 samples total).

Additionally, four hundred samples of the letter c and four hundred samples of the letter e were generated using the IDMGEN program and a machine print type form of the letter e and the letter c in Knuth's CMSS (Computer Modern Sans Serif) typeface as the ideal prototype (800 samples total). This provided a training set of 1600 samples, equally divided between CMR and CMSS.

Figure 5.2: Pure Samples



A kNN Classifier was trained to differentiate between two characters using the CMR and CMSS training sets. The classifier was then tested on a test set consisting of 200 CMR samples and 200 CMSS samples. Rates of accuracy for the 400 test samples were recorded.

5.3.2 Step 2:

Next, Ten sets of 80 samples of the letter c and ten sets of 80 samples of the letter e were generated using the IDMGEN program. The ideal prototype of each of the ten sets was as follows; a machine print type form of the letter e and a machine print type form of the letter c in Knuth's CMR (Computer Modern Roman) typeface for the first set, a machine print type form of the letter e and a machine print type form of the letter c in Knuth's CMSS (Computer Modern Sans Serif) typeface for the second set, a machine print type form consisting of an interpolation which is 90 percent CMR and 10 percent CMSS for the third set, and so on until the last set which consists of an interpolation which is 10 percent CMR and 90 percent CMSS. The final result of this process is a training set consisting of 160 CMR samples, 160 CMSS samples, and 1280 interpolated samples for a total of 1600 mostly interpolated training samples.

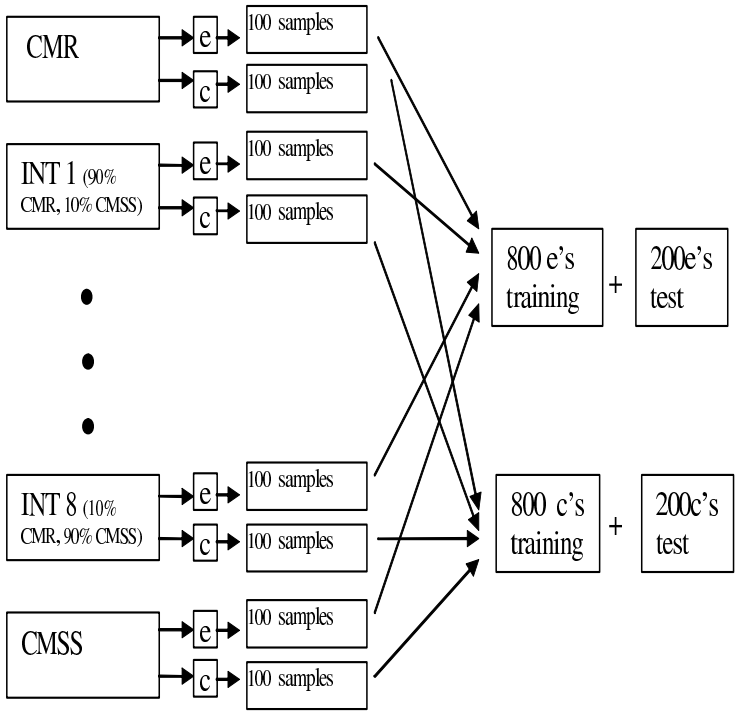
The classifier was trained on the above samples and then tested on the same 400 test samples from the first part of the experiment (step 1.) Rates of accuracy were recorded.

This tested whether the classifier which has been trained on interpolated data performed equally well on pure samples as a classifier trained on only pure data.

5.3.3 Step 3:

Next, Ten sets of 20 samples of the letter c and ten sets of 20 samples of the letter e were generated using the IDMGEN program. The ideal prototype of each of the ten sets was as follows; a machine print type form of the letter e and a machine print type form of the letter c in Knuth's CMR (Computer Modern Roman) typeface for the first set, a machine

Figure 5.3: Interpolated Samples



print type form of the letter e and a machine print type form of the letter c in Knuth's CMSS (Computer Modern Sans Serif) typeface for the second set, a machine print type form consisting of an interpolation which is 90 percent CMR and 10 percent CMSS for the third set, and so on until the last set which consists of an interpolation which is 10 percent CMR and 90 percent CMSS. The final result of this process was a test set consisting of 40 CMR samples, 40 CMSS samples, and 320 interpolated samples for a total of 400 mostly interpolated test samples.

A classifier was next trained to distinguish between e's and c's by using the same samples as in step 1 (pure font types), and tested on the interpolated samples described above. Rates of accuracy were recorded.

This tested how well the classifier which has been trained on only pure fonts performs when it is tested on interpolated fonts.

5.3.4 Step 4:

A classifier was next trained on the same samples as in step 2 (interpolated data), and tested on the interpolated test set from step 3. Rates of accuracy were recorded.

This tested how well a classifier performs when it has been trained on interpolated data and tested on interpolated data.

5.4 Summary of the Design:

- A = pure data (CMR and CMSS fonts)
- B = interpolated data (interpolated fonts)

A classifier was next trained on the same samples as in step 2 (interpolated data), and tested on the interpolated test set from step 3. Rates of accuracy were recorded.

Two hypotheses were proposed and tested using the χ^2 statistic. Although McNemar's test (Dieterrich, T.G.) has more power, we decided that the χ^2 was sufficient for our purposes, and simpler to implement.

5.5 Hypothesis 1:

AB is trained on only pure data and tested on mostly interpolated data. BB is trained and tested on interpolated data. We expect that BB will perform significantly better than AB. Therefore our null hypothesis is that AB will perform at least as well as BB. This part of the experiment speaks to the performance or strength of our algorithm.

Figure 5.4: Test Matrix

Train On \ Test On	A	B
A	A/A	A/B
B	B/A	B/B

The areas of interest in this part of our experiment are the differences between AB and BB.

5.6 Statistic 1:

We choose a χ^2 test for our experiment. Recall that in AB the classifier is trained on pure data, while BB is trained on interpolated data. Both AB and BB are tested on previously interpolated samples. In this test, the null hypothesis is that AB will perform at least as well as BB. If the null hypothesis is rejected then our classifier trained on mostly interpolated data has performed better than the classifier trained on only pure data and we can say that our interpolated classifier is better at classifying interpolated data.

To apply the χ^2 test, we have constructed two training sets S_A , and S_B and one test set *T*. We train our classifier in AB on sample S_A and our classifier in BB on sample S_B. yielding classifiers $\hat{f}_A$ and $\hat{f}_B$. We then test these classifiers on the test set *T*. For each example $x \in T$, we record how it was classified and construct the following contingency

5.6. STATISTIC 1:

table:

number of samples correct	number of samples misclassified
correctly classified by $\hat{f}_A$	misclassified by $\hat{f}_A$
correctly classified by $\hat{f}_B$	misclassified by $\hat{f}_B$

We will use the notation

n_{00}	n_{01}
n_{10}	n_{11}

where $n = n_{00} + n_{01}$ and $n = n_{10} + n_{11}$ are the total number of examples in the test set *T and* n_{ii} is the observed count for each type of test. [DI98]

Under the null hypothesis, if AB performs as well as BB then the error rate for BB is greater than or equal to the error rate for AB, $n_{11} >= n_{01}$. The test is based on a χ^2 test for goodness-of-fit that compares the distribution of counts expected under the null hypothesis to the observed counts. The expected counts under the null hypothesis are well known and they are as follows.

$(n_{00} + n_{01}) * (n_{00} + n_{10}) / \sum n_{ij}$	$(n_{00} + n_{01}) * (n_{01} + n_{11}) / \sum n_{ij}$
$(n_{10} + n_{11}) * (n_{00} + n_{10}) / \sum n_{ij}$	$(n_{10} + n_{11}) * (n_{01} + n_{11}) / \sum n_{ij}$

The following statistic is used. Yate's correction for continuity is used if the number observed in a cell is less than 5.

$$\sum \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

If the null hypothesis is correct, then the probability that this quantity is greater than $\chi^2_{1,0.95} = 3.84$ is less than 0.05. So we would reject the null hypothesis in favor of the alternative hypothesis that BB performs better than AB when tested on the particular training set *C*.

5.7 Hypothesis 2:

AA is trained and tested on pure data. BA is trained on mostly interpolated data and tested on pure data. We would hope that BA would not perform significantly worse than AA. If this is the case, we have shown that in our experiment, training on interpolated data does not "hurt" the classifier in the identification of pure data. In this way we are testing if our algorithm is "safe". Our null hypothesis is that BA and AA have the same accuracy and we would hope that the null hypothesis holds.

The areas of interest in this part of the experiment is the difference between AA and BA.

5.8 Statistic 2:

We choose a χ^2 test for our experiment. Recall that in AA the classifier is trained on pure data, while BA is trained on mostly interpolated data. Both AA and BA are tested only on pure samples. For this test, the null hypothesis is that AA and BA perform equally. If our null hypothesis is not rejected then we can say that training on interpolated data does not appear to "hurt" the classifier in the identification of pure data.

To apply the χ^2 test, we have constructed two training sets S_A , and S_B and one test set *T*. We train our classifier in AA on sample S_A and our classifier in BA on sample S_B. yielding classifiers $\hat{f}_A$ and $\hat{f}_B$. We then test these classifiers on the test set *T*. For each example $x \in T$, we record how it was classified and construct the following contingency table:

number of samples correct	number of samples misclassified
correctly classified by $\hat{f}_A$	misclassified by $\hat{f}_A$
correctly classified by $\hat{f}_B$	misclassified by $\hat{f}_B$

We will use the notation

5.8. STATISTIC 2:

n_{00}	n_{01}
n_{10}	n_{11}

where $n = n_{00} + n_{01}$ and $n = n_{10} + n_{11}$ are the total number of examples in the test set *T and* n_{ii} is the observed count for each type of test. [DI98]

Under the null hypothesis, if the two algorithms have the same error rate, $n_{00} = n_{01}$. The test is based on a χ^2 test for goodness-of-fit that compares the distribution of counts expected under the null hypothesis to the observed counts. The expected counts under the null hypothesis are well known and they are as follows.

$(n_{00} + n_{01}) * (n_{00} + n_{10}) / \sum n_{ij}$	$(n_{00} + n_{01}) * (n_{01} + n_{11}) / \sum n_{ij}$
$(n_{10} + n_{11}) * (n_{00} + n_{10}) / \sum n_{ij}$	$(n_{10} + n_{11}) * (n_{01} + n_{11}) / \sum n_{ij}$

The following statistic is used. Yate's correction for continuity is used if the number observed in a cell is less than 5.

$$\sum \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

If the null hypothesis is not rejected, then the probability that this quantity is greater than $\chi^2_{1,0.95} = 3.84$ is less than 0.05. If this quantity is less than 3.84 we would not reject the null hypothesis which states that the two algorithms have the same performance when trained on the particular training set *C*.

Chapter 6

Experiments

6.1 CMR and CMSS C and E Experiments

Our first set of experiments was performed using a serifed and a sans-serif font as the basis of our interpolations.

6.1.1 Experimental Description

For this experiment, the default parameters were used. The images for both test and training data were only slightly blurred as were the interpolated samples. The test interpolated data was taken from the entire range between the real CMR and CMSS fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: default parameters

Pure Seed: no seed

Number Samples: 1600

Interpolated Parameters: default parameters

Interpolated Seed: numb (0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

6.1. CMR AND CMSS C AND E EXPERIMENTS

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure parameters: default parameters

Pure Seed: no seed

Number Samples: 400

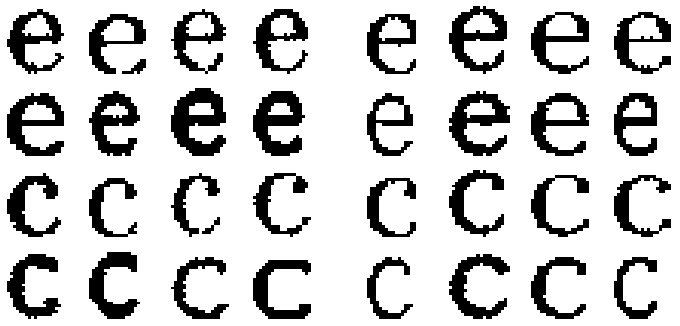
Interpolated Parameters: default parameters

Interpolated Seed: numb (0-10)

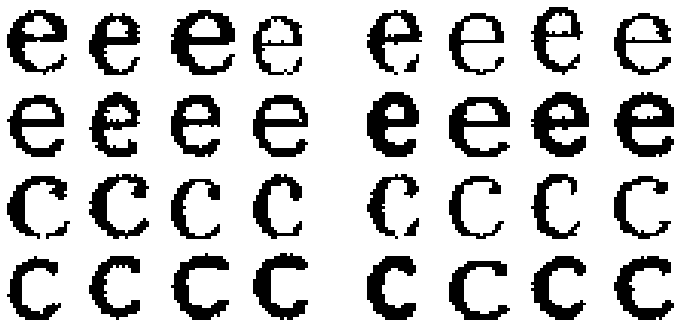
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples

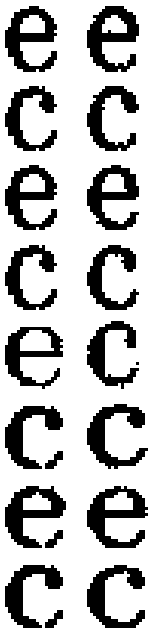


Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 0 c: 0	0	e: 1 c: 0	1
	Itrp(B)	e: 0 c: 0	0	e: 0 c: 0	0

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 399	o_2 1
		e_1 399	e_2 1
	Itrp(B)	o_3 400	o_4 0
		e_3 399	e_4 1

As explained earlier, the result of our test is $\chi^2 = 1.0$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 400	e_2 0
	Itrp(B)	o_3 400	o_4 0
		e_3 400	e_4 0

Clearly there is no difference between the performances of the classifiers so we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally well when tested on the interpolated data as well as the real data. In fact, both were practically perfect! The test was not interesting, as the e's and c's were too easily distinguished.

6.1.2 Experimental Description

For this experiment, the generated samples were slighlty more blurred and had a slight amount of noise added. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMSS fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

6.1. CMR AND CMSS C AND E EXPERIMENTS

Parameters

Training data:

Pure Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

Pure Seed: no seed

Number Samples: 1600

Interpolated Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

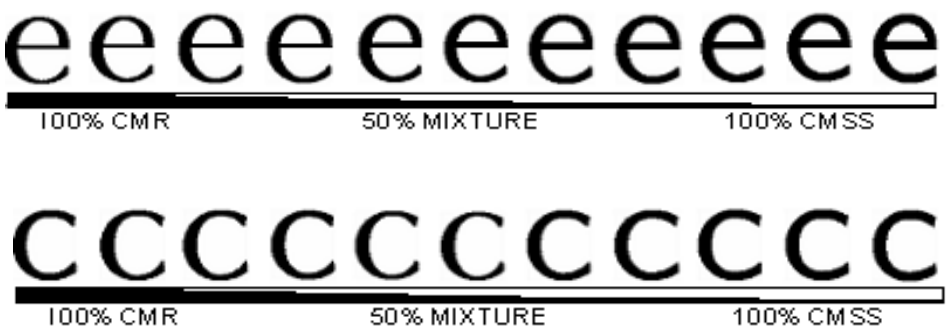
Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

Pure Test Seed: no seed

Number Samples: 400

Interpolated Test data: -e1.0,1.1 -t.15,.125 -s.130,.125

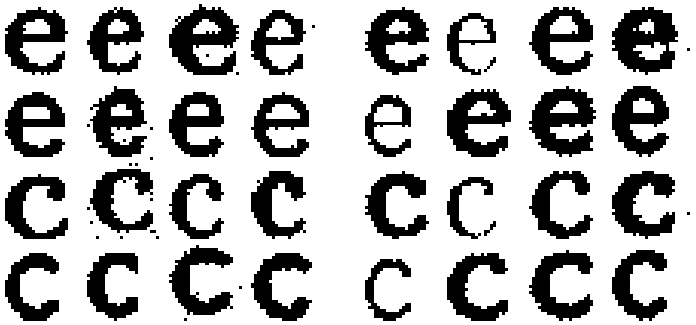
Interpolated Test Seed: numb(0-10)

Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples

Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 0 c: 0	0	e: 0 c: 0	0
	Itrp(B)	e: 0 c: 1	1	e: 0 c: 0	0

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data.

The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 400	e_2 0
	Itrp(B)	o_3 400	o_4 0
		e_3 400	e_4 0

Clearly there is no difference between the performances of the classifiers so we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 399	e_2 1
	Itrp(B)	o_3 399	o_4 1
		e_3 399	e_4 1

As explained earlier, the result of our test is $\chi^2 = 1.0$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, once again, both classifiers did well on all data so that the results were not interesting. There was only one error. It was decided that more distortion and blurring should be added. All classifiers did equally well on all data.

6.1.3 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. A seed of 2 was added to the real data so that it was randomized differently from the interpolated data. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMSS fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Pure Seed: -S2

Number Samples: 1600

Interpolated Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Pure Seed: -S2

Number Samples: 400

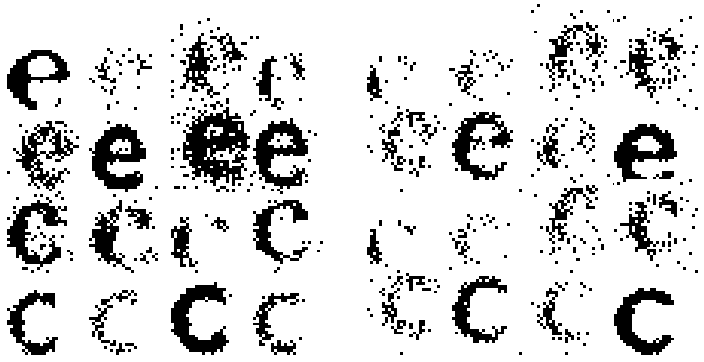
Interpolated Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Interpolated Seed: numb(0-10)

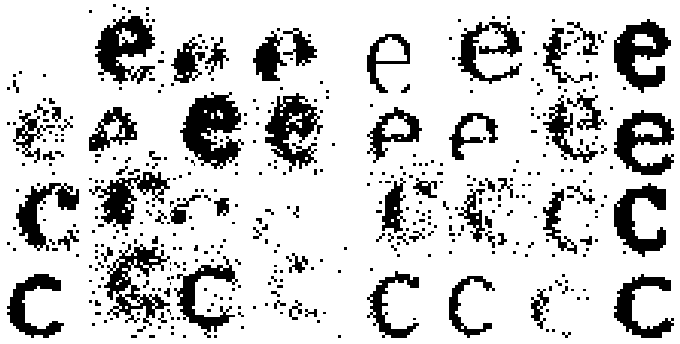
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 33 c: 41	74	e: 29 c: 33	62
	Itrp(B)	e: 23 c: 46	69	e: 31 c: 41	72

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 338	o_2 62
		e_1 333	e_2 67
	Itrp(B)	o_3 328	o_4 72
		e_3 333	e_4 67

As explained earlier, the result of our test is $\chi^2 = 0.88$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 326	o_2 74
		e_1 328	e_2 72
	Itrp(B)	o_3 331	o_4 69
		e_3 328	e_4 72

As explained earlier, the result of our test is $\chi^2 = 0.20$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, the results were more interesting. Both classifiers did badly when tested on the interpolated data as well as the real data. The data was so distorted that letters were difficult to recognize by either classifier. However, the classifier trained on interpolated data did not do any worse when tested on the real data. Interestingly, the classifier trained on the interpolated data did slightly better on the real and slightly worse on the interpolated data, however not significantly so.

6.1.4 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, the variance was limited to half of what it was in the previous experiment. Real training data had a seed of 5, while the real test data had a seed of 11. The interpolated training data had a variable seed, however the interpolated test data had a different variable seed. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMSS fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.

6.1. CMR AND CMSS C AND E EXPERIMENTS

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Pure Seed: -S5

Number Samples: 1600

Interpolated Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Interpolated Seed: Snumb (0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Pure Seed: -S11

Number Samples: 400

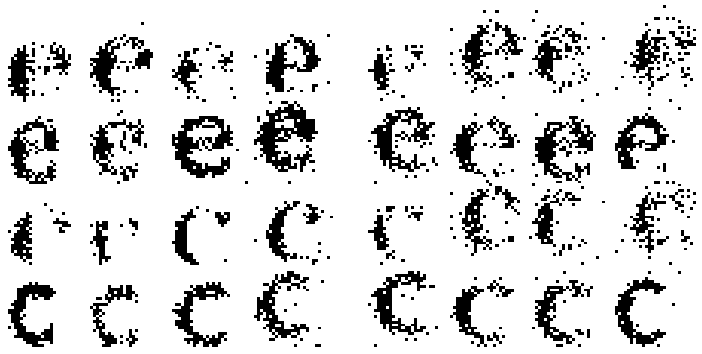
Interpolated Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Interpolated Test Seed: numb * 3 (0, 3, 6....30)

Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples

Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 30 c: 14	44	e: 47 c: 13	60
	Itrp(B)	e: 27 c: 16	43	e: 32 c: 10	42

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 340	o_2 60
		e_1 349	e_2 51
	Itrp(B)	o_3 358	o_4 42
		e_3 349	e_4 51

As explained earlier, the result of our test is $\chi^2 = 3.62$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 356	o_2 44
		e_1 356	e_2 44
	Itrp(B)	o_3 357	o_4 43
		e_3 356	e_4 44

As explained earlier, the result of our test is $\chi^2 = 0.2$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally when tested on the real data. They each misclassified about 10 per cent of the samples. The classifier trained on the interpolated data did somewhat better than the one trained on the real data when tested on the interpolated data, however not significantly so. It is possible that with the choice of a stronger statistic that these results might have been significant.

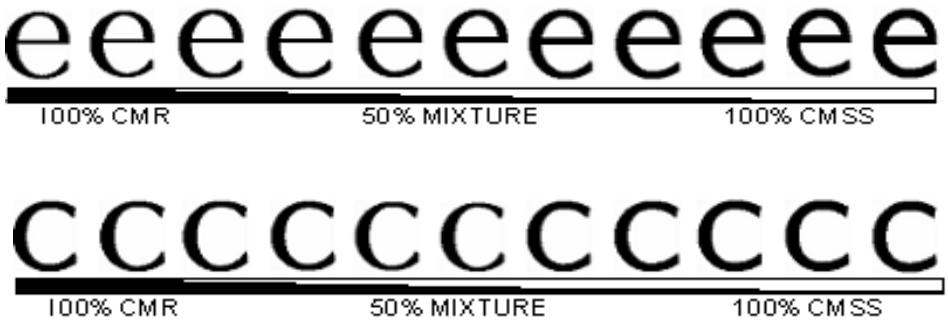
6.1.5 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, variance was decreased to slightly above the default levels. Seeds were kept the same as in the last test. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMSS fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Pure Seed: -S5

Number Samples: 1600

Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

Pure	50	**	**	**	**	**	**	**	**	**	50

Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Real Test Seed: -S11

Number Samples: 400

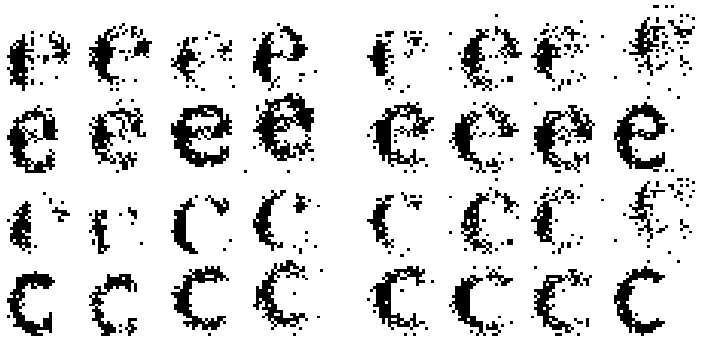
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Interpolated Seed: numb * 2 (0,2,4...20)

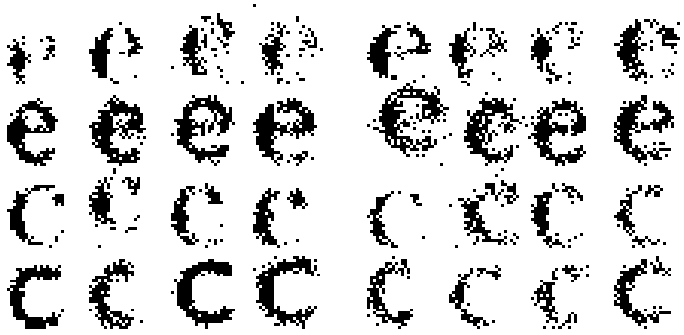
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 35 c: 6	41	e: 60 c: 4	64
	Itrp(B)	e: 42 c: 3	45	e: 32 c: 7	39

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 336	o_2 64
		e_1 348	e_2 52
	Itrp(B)	o_3 361	o_4 39
		e_3 348	e_4 52

As explained earlier, the result of our test is $\chi^2 = 6.90$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 359	o_2 41
		e_1 357	e_2 43
	Itrp(B)	o_3 355	o_4 45
		e_3 357	e_4 43

As explained earlier, the result of our test is $\chi^2 = 0.20$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers performed equally when tested on the real data. When tested on the interpolated data, the classifier trained on real data did significantly worse than the one trained on the interpolated data.

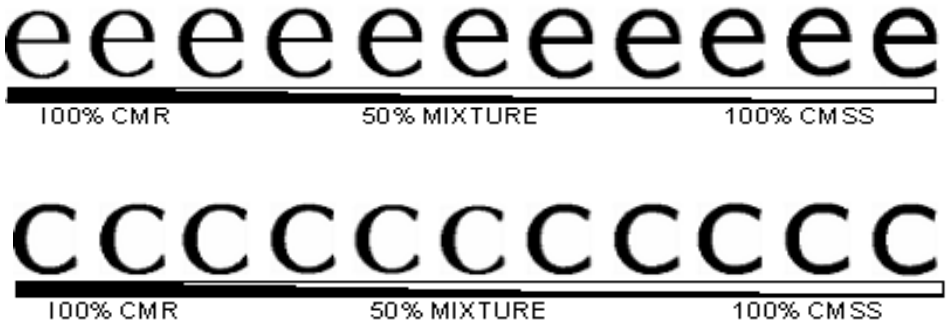
6.1.6 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, variance was slight. This differed from the last experiment in that the interpolated training samples were all taken from the midpoint between CMR and CMSS.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

6.1. CMR AND CMSS C AND E EXPERIMENTS

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Pure Seed: -S12

Number Samples: 1600

Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

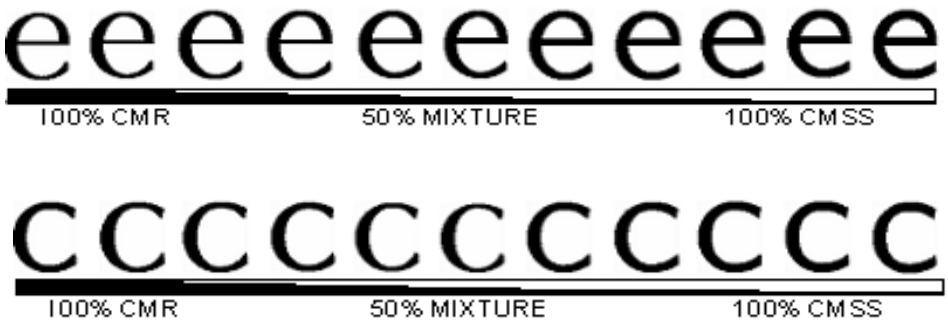
Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	**	**	**	**	*	100	**	**	**	**	**

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Real Test Seed: -S5

Number Samples: 400

Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

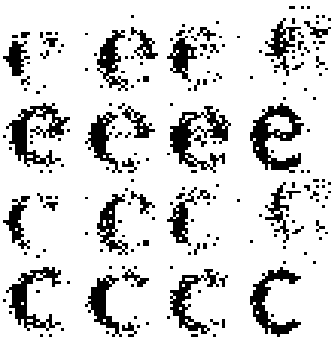
Interpolated Seed: -S5

Number Samples: 400

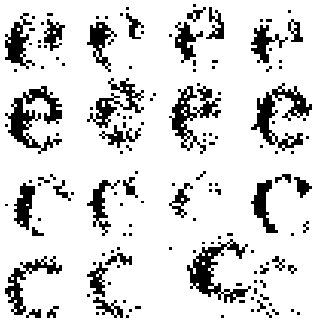
Samples chosen to illustrate data used

Pure Training Samples

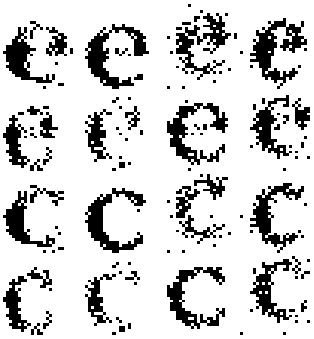
Interpolated Training Samples



Pure Test Samples



Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 50 c: 2	52	e: 76 c: 3	79
	Itrp(B)	e: 35 c: 5	40	e: 70 c: 1	71

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data.

The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 321	o_2 79
		e_1 325	e_2 75
	Itrp(B)	o_3 329	o_4 71
		e_3 325	e_4 75

As explained earlier, the result of our test is $\chi^2 = 0.50$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 348	o_2 52
		e_1 354	e_2 46
	Itrp(B)	o_3 360	o_4 40
		e_3 354	e_4 46

As explained earlier, the result of our test is $\chi^2 = 1.76$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, the interpolated classifier did slightly better on the pure data, as well as the interpolated data, however not significantly so.

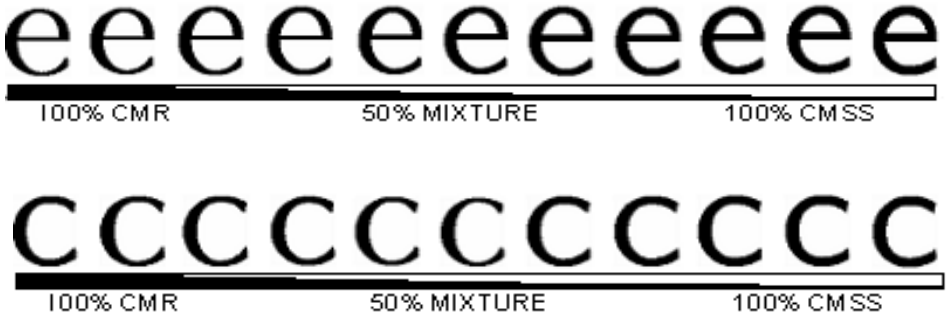
6.1.7 Experimental Description

For this experiment, the generated samples were less blurred, and had a moderate amount of noise added. The varience was less than the last experiment. Training was performed as before and the experiment took the test interpolated data entirely from the midpoint between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e1.8,1.0 -t.3,.110 -s.3,.11

Pure Seed: -S12

Number Samples: 1600

Interpolated Parameters: -e1.8,1.0 -t.3,.11 -s.3,.11

Interpolated Seed: numb (0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	**	**	**	**	*	100	**	**	**	**	**

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Test data: -e1.8,1.0 -t.3,.110 -s.3,.11

Real Seed: -S5

Number Samples: 400

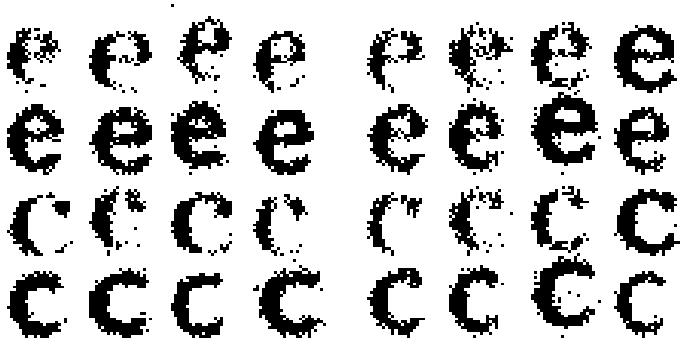
Interpolated Test data: -e1.8,1.0 -t.3,.11 -s.3,.11

Interpolated Test Seed: -S5

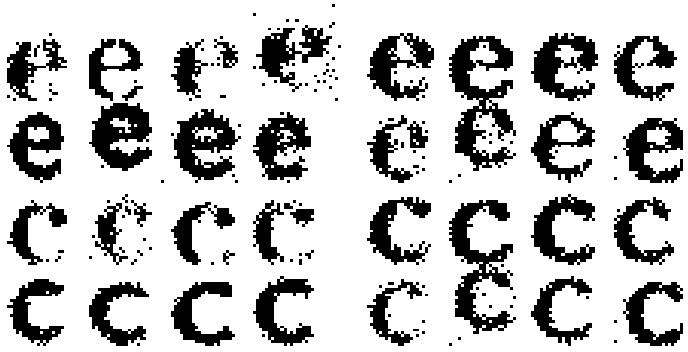
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples

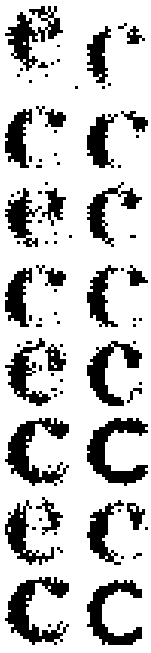


Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		Itrp(B)	
TRAIN ON	Pure(A)	e: 6 c: 0	6	e: 23 c: 0	23
	Itrp(B)	e: 7 c: 0	7	e: 10 c: 0	10

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 377	o_2 23
		e_1 383	e_2 17
	Itrp(B)	o_3 390	o_4 10
		e_3 383	e_4 17

As explained earlier, the result of our test is $\chi^2 = 5.20$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 394	o_2 6
		e_1 393	e_2 7
	Itrp(B)	o_3 393	o_4 7
		e_3 393	e_4 7

As explained earlier, the result of our test is $\chi^2 = 0.14$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, the classifier trained on interpolated data did significantly better than the one trained on real data only. Both classifiers did equally well when tested on the real samples.

6.1.8 CMR-CMSS Test Group Results

Following is a graphical illustration of our results. In this particular series of tests, interpolated data was safe in every instance, as shown by the results of hypothesis 2. The classifier trained on interpolated data was better in two instances, test 5 (greatly blurred, little variance, full range) and test 7 (slightly blurred, little variance, midpoint).

In looking at the chart, we see that both classsifiers did so well on the first two tests that there was no difference. By making the characters more distorted, with high variance, both classifiers had a much more difficult time classifying the characters. In many cases, the greatly blurred characters would be difficult for a human to distinguish as well. Test 7 is a more realistic test. In this case, the characters are only slightly blurred. All the characters, however, are taken from the midpoint of the interpolations between the two fonts.

Based on the results of these tests we decided to experiment on interpolations between two more varied fonts.

Figure 6.1: CMR-CMSS e and c Experimental Results

		Hypothesis 1				Hypothesis 2			
		Errors		Statistic		Errors		Statistic	
IMAGE QUALITY	RANGE	AB	BB	χ^2	Rej	AA	BA	χ^2	Rej
normal	full	0	1	1.00	no	0	0	0.00	no
slightly blurred	full	0	0	0.00	no	0	1	1.00	no
greatly blurred, high variance	full	62	72	.88	no	74	69	.20	no
greatly blurred, some variance	full	60	42	3.62	no	44	43	.20	no
greatly blurred, little variance	full	64	39	6.90	yes	41	45	.20	no
greatly blurred, little variance	mid	79	71	.50	no	52	40	1.76	no
slightly blurred, little variance	mid	23	10	5.20	yes	6	7	1.40	no

6.2 CMR and CMFF C and E Experiments

We next thought it would be interesting to pick two more dis-similar fonts to use as the basis for our interpolations.

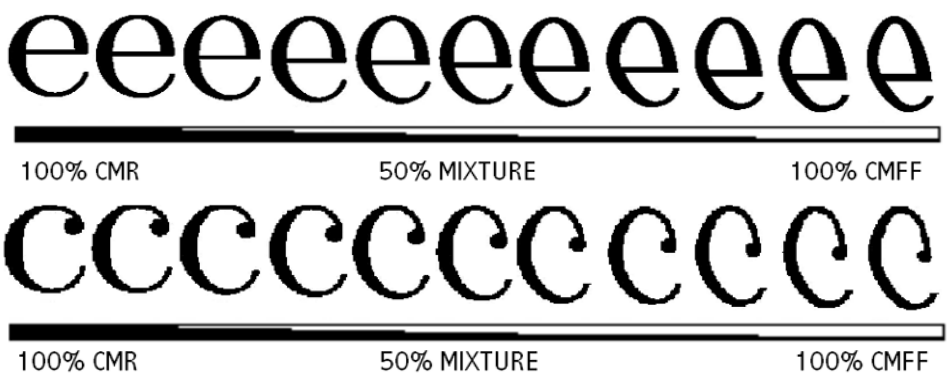
6.2.1 Experimental Description

For this experiment, the default parameters were used. The images for both test and training data were only slightly blurred as were the interpolated samples. The test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: default parameters

Pure Seed: no seed

Number Samples: 1600

Interpolated Parameters: default parameters

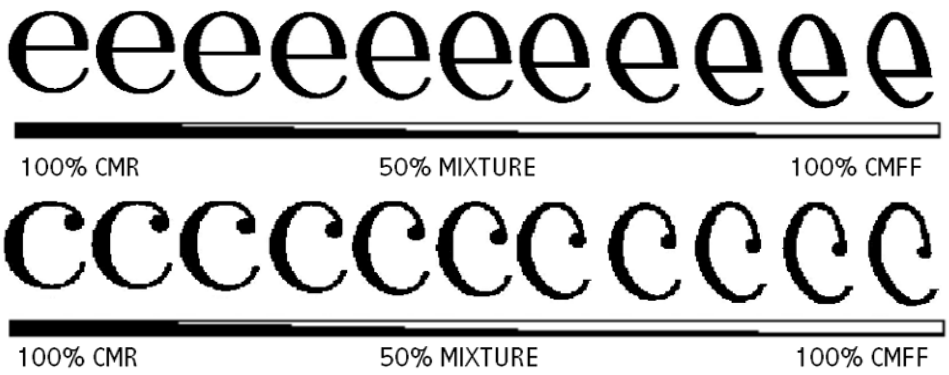
Interpolated Seed: numb (0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure parameters: default parameters

Pure Seed: no seed

Number Samples: 400

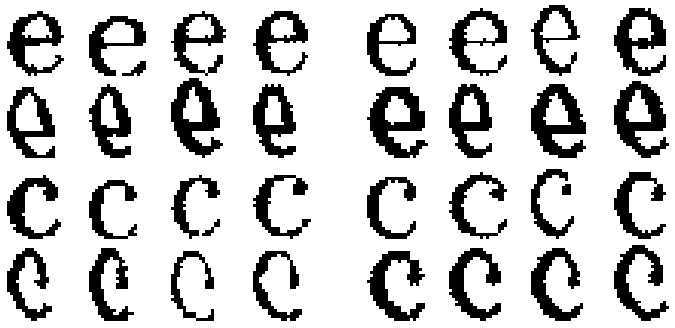
Interpolated Parameters: default parameters

Interpolated Seed: numb (0-10)

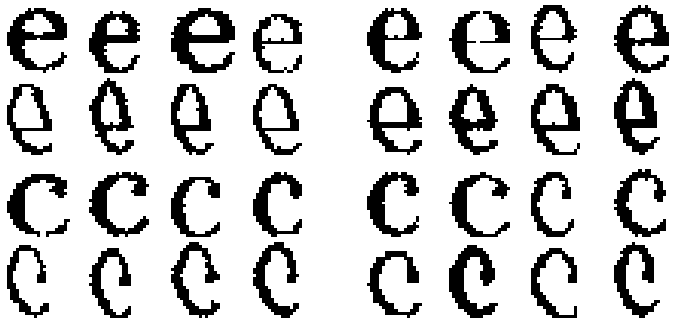
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 0 c: 0	0	e: 23 c: 0	23
	Itrp(B)	e: 0 c: 0	0	e: 4 c: 0	4

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 377	o_2 23
		e_1 386	e_2 14
	Itrp(B)	o_3 396	o_4 4
		e_3 386	e_4 14

As explained earlier, the result of our test is $\chi^2 = 10.71$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 400	e_2 0
	Itrp(B)	o_3 400	o_4 0
		e_3 400	e_4 0

Clearly there is no difference between the performances of the classifiers so we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally well when tested on the pure data. In fact, they each performed perfectly. When tested on the interpolated data, however, the classifier trained on the pure data did much worse than the one trained on the interpolated data. In fact, it misclassified 23 of the e's and none of the c's. The classifier trained on the interpolated data performed significantly better when tested on the interpolated samples.

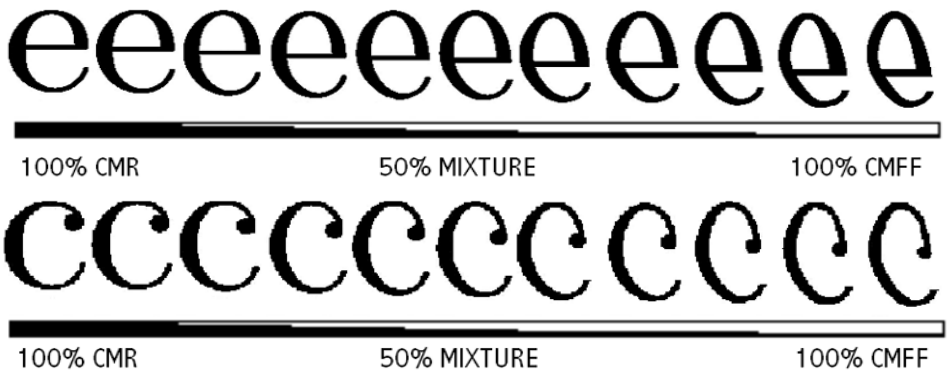
6.2.2 Experimental Description

For this experiment, the generated samples were slighlty more blurred and had a slight amount of noise added. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure 50 ** ** ** ** ** ** ** ** ** 50

Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

Pure Seed: no seed

Number Samples: 1600

Interpolated Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

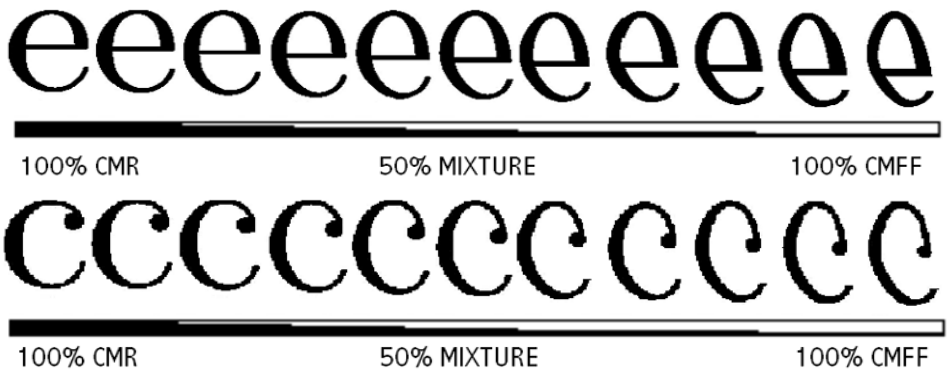
Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

Pure Test Seed: no seed

Number Samples: 400

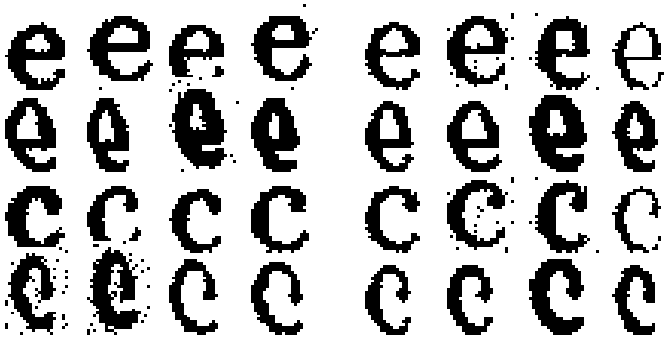
Interpolated Test data: -e1.0,1.1 -t.15,.125 -s.130,.125

Interpolated Test Seed: numb(0-10)

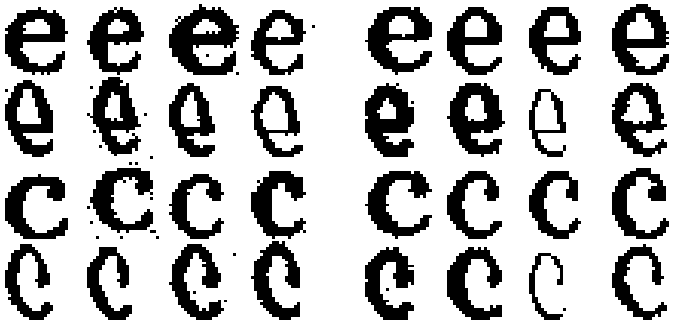
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples

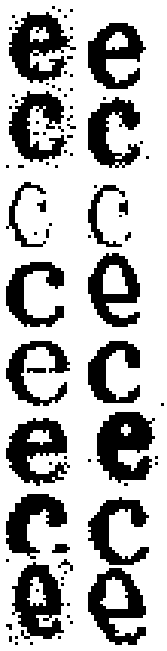


Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON			
		Pure(A)		Itrp(B)	
TRAIN ON	Pure(A)	e: 0 c: 0	0	e: 15 c: 3	18
	Itrp(B)	e: 1 c: 1	2	e: 2 c: 1	3

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 382	o_2 18
		e_1 389	e_2 11
	Itrp(B)	o_3 397	o_4 3
		e_3 389	e_4 11

As explained earlier, the result of our test is $\chi^2 = 7.93$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 399	e_2 1
	Itrp(B)	o_3 398	o_4 2
		e_3 399	e_4 1

As explained earlier, the result of our test is $\chi^2 = 0.00$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, once again, both classifiers did well on the pure data with no significant difference. The classifier trained on interpolated data did significantly better when tested on the interpolated data than the one trained only on the pure data.

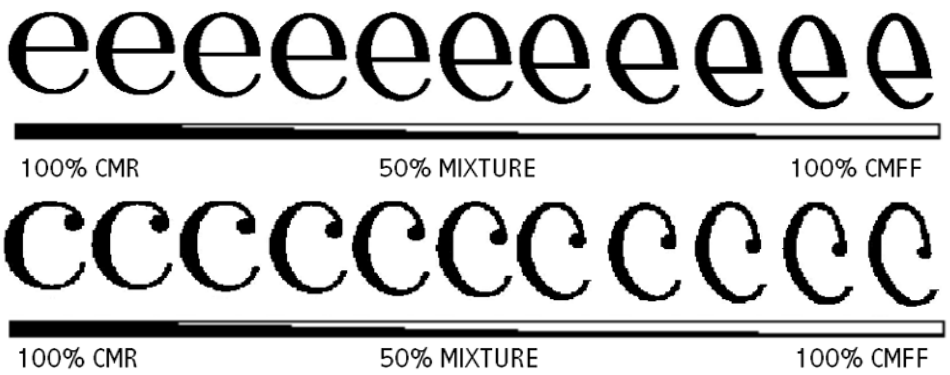
6.2.3 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. A seed of 2 was added to the real data so that it was randomized differently from the interpolated data. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Pure Seed: -S2

Number Samples: 1600

Interpolated Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

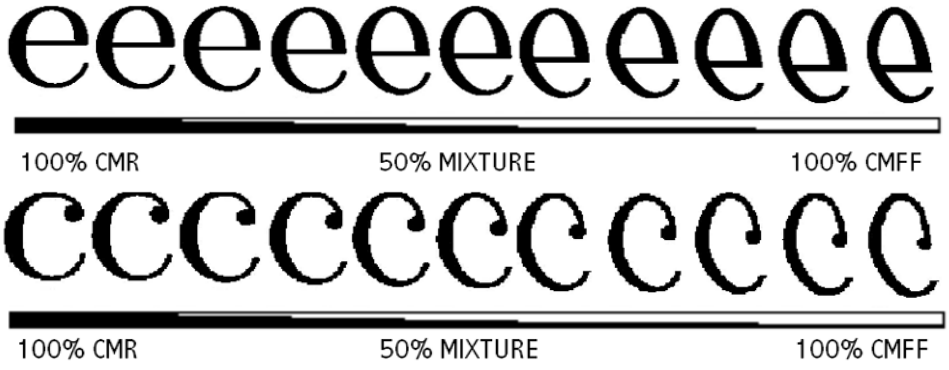
Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Pure Seed: -S2

Number Samples: 400

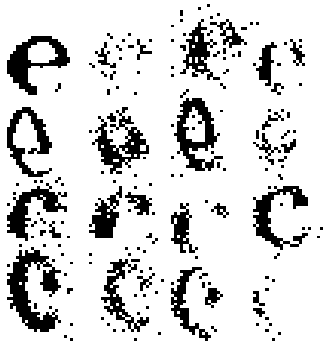
Interpolated Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Interpolated Seed: numb(0-10)

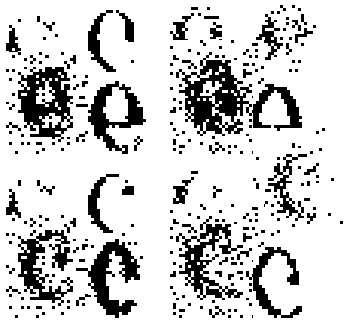
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples

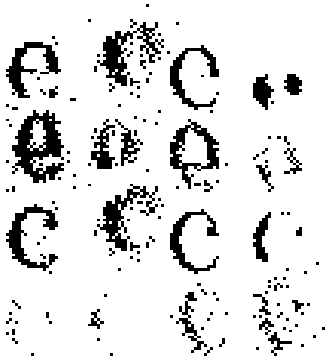


Interpolated Training Samples



Pure Test Samples

Interpolated Test Samples



Samples of Data with closest match

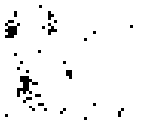
The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.


pure/pure e's


pure/pure c's

interp/pure e's

interp/pure c's


pure/interp e's


pure/interp c's

interp/interp e's


interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 33 c: 58	91	e: 42 c: 60	102
	Itrp(B)	e: 22 c: 61	83	e: 38 c: 68	106

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data.

The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 298	o_2 102
		e_1 296	e_2 104
	Itrp(B)	o_3 294	o_4 106
		e_3 296	e_4 104

As explained earlier, the result of our test is $\chi^2 = 0.8$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 309	o_2 91
		e_1 313	e_2 87
	Itrp(B)	o_3 317	o_4 83
		e_3 313	e_4 87

As explained earlier, the result of our test is $\chi^2 = 0.46$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

Both classifiers did badly when tested on the interpolated data as well as the pure data. The data was so distorted that letters were difficult to recognize by either classifier. Neither classifier did significantly different on either of the types of test data, possibly because the letters were so distorted.

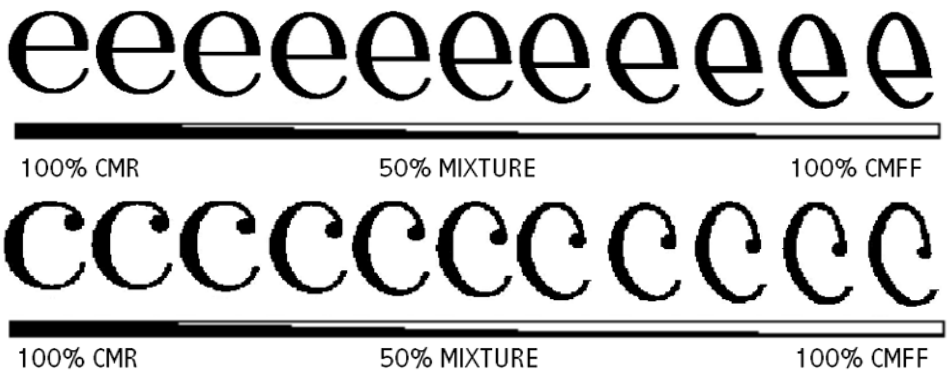
6.2.4 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, the variance was limited to half of what it was in the previous experiment. Real training data had a seed of 5, while the real test data had a seed of 11. The interpolated training data had a variable seed, while the interpolated test data had a different seed. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Pure Seed: -S5

Number Samples: 1600

Interpolated Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Interpolated Seed: Snumb (0-10)

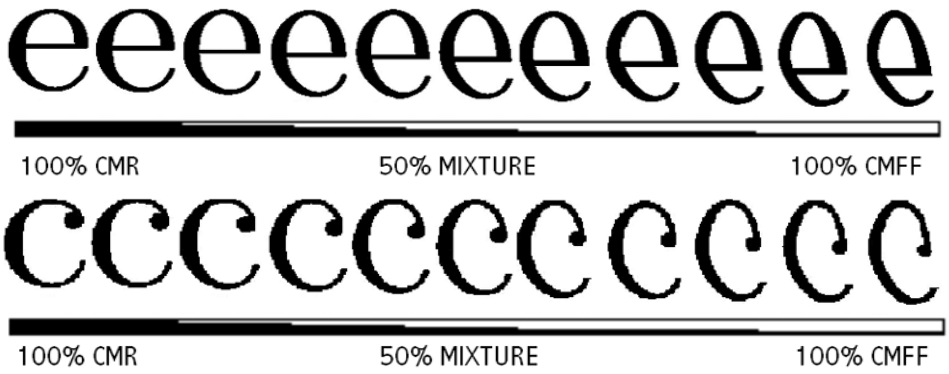
Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

6.2. CMR AND CMFF C AND E EXPERIMENTS



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Pure Seed: -S11

Number Samples: 400

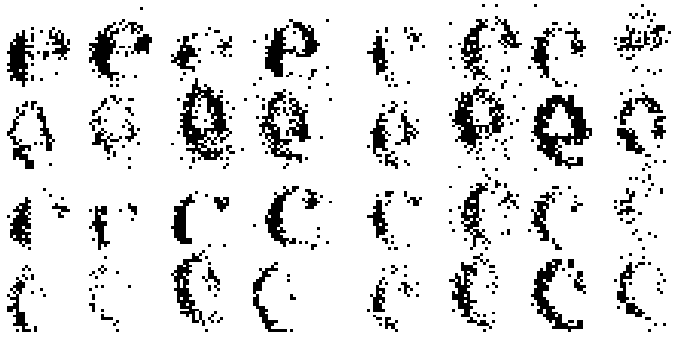
Interpolated Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Interpolated Test Seed: numb * 3 (0, 3, 6....30)

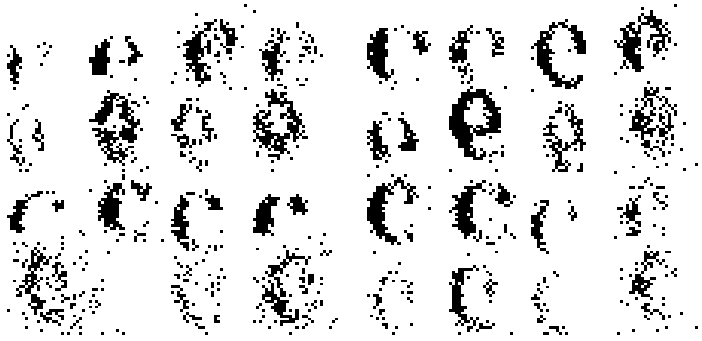
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 47 c: 18	65	e: 68 c: 16	84
	Itrp(B)	e: 54 c: 19	73	e: 56 c: 27	83

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 316	o_2 84
		e_1 316	e_2 84
	Itrp(B)	o_3 317	o_4 83
		e_3 316	e_4 84

As explained earlier, the result of our test is $\chi^2 = 0.1$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 335	o_2 65
		e_1 331	e_2 69
	Itrp(B)	o_3 327	o_4 73
		e_3 331	e_4 69

As explained earlier, the result of our test is $\chi^2 = 0.54$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally when tested on the pure data. They each misclassified more than 15 per cent of the samples. Once again, the letters were so distorted that they were difficult for either classifer to recognize. The classifier trained on the interpolated data did somewhat better than the one trained on the pure data when tested on the interpolated data, however not significantly so.

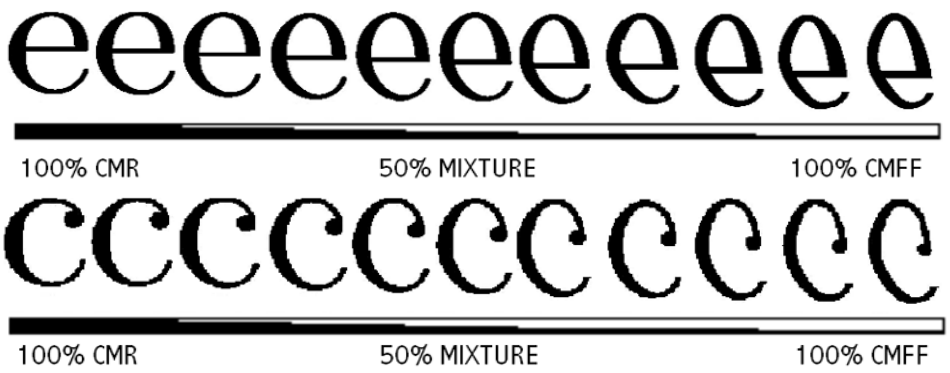
6.2.5 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, variance was decreased to slightly above the default levels. Seeds were kept the same as in the last test. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Pure Seed: -S5

Number Samples: 1600

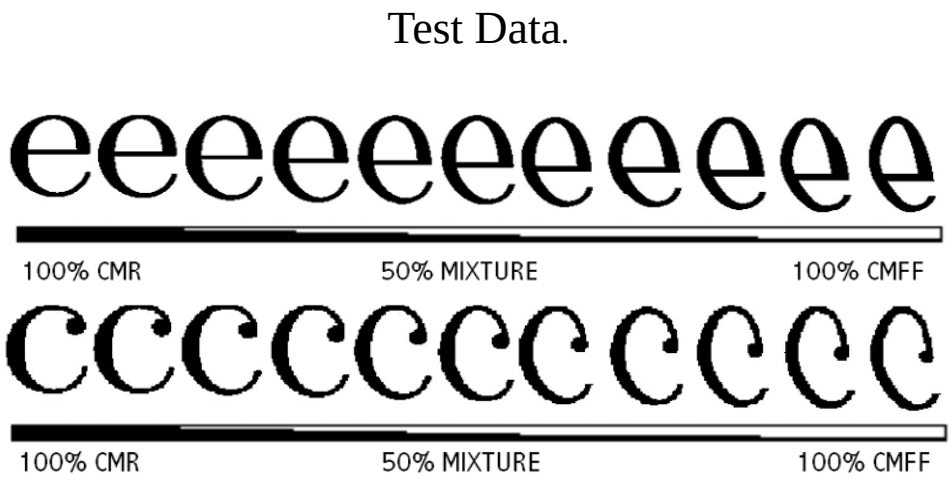
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.



Pure	50	**	**	**	**	**	**	**	**	**	50

Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

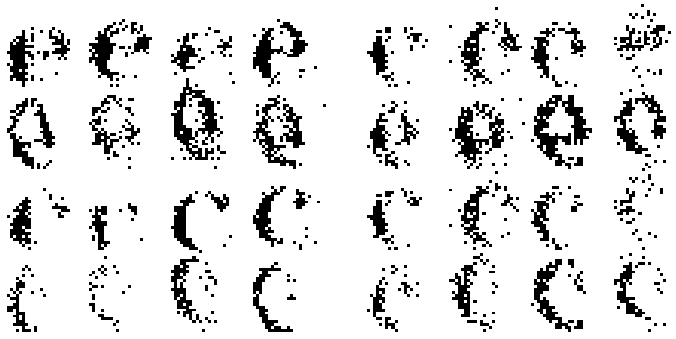
Parameters

Test data:

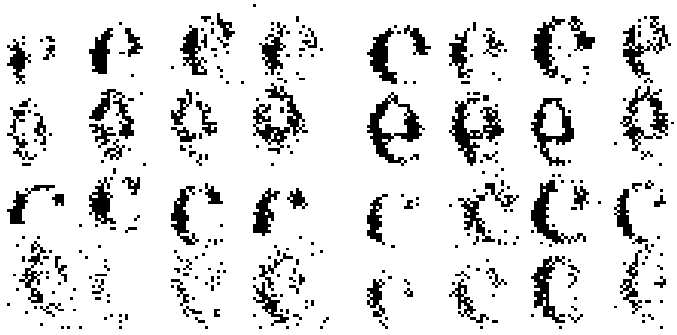
Real Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Real Test Seed: -S11
Number Samples: 400
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Interpolated Seed: numb * 2 (0,2,4...20)
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 53 c: 5	58	e: 68 c: 13	81
	Itrp(B)	e: 50 c: 14	64	e: 51 c: 6	57

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 319	o_2 81
		e_1 331	e_2 69
	Itrp(B)	o_3 343	o_4 57
		e_3 331	e_4 69

As explained earlier, the result of our test is $\chi^2 = 5.2$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 342	o_2 58
		e_1 339	e_2 61
	Itrp(B)	o_3 336	o_4 64
		e_3 339	e_4 61

As explained earlier, the result of our test is $\chi^2 = 0.32$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers performed equally when tested on the pure data. When tested on the interpolated data, the classifier trained on pure data did significantly worse than the one trained on the interpolated data. This is in line with the results from the same experiment performed on the CMR-CMSS interpolations.

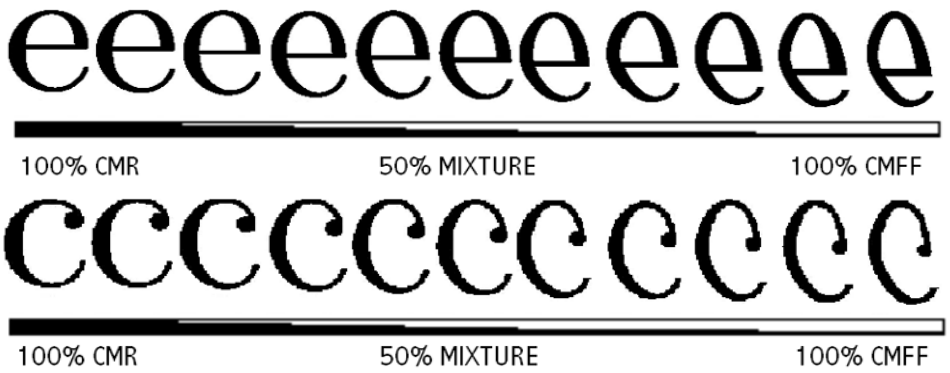
6.2.6 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, variance was slight. This differed from the last experiment in that the interpolated samples were all taken from the midpoint between CMR and CMFF.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

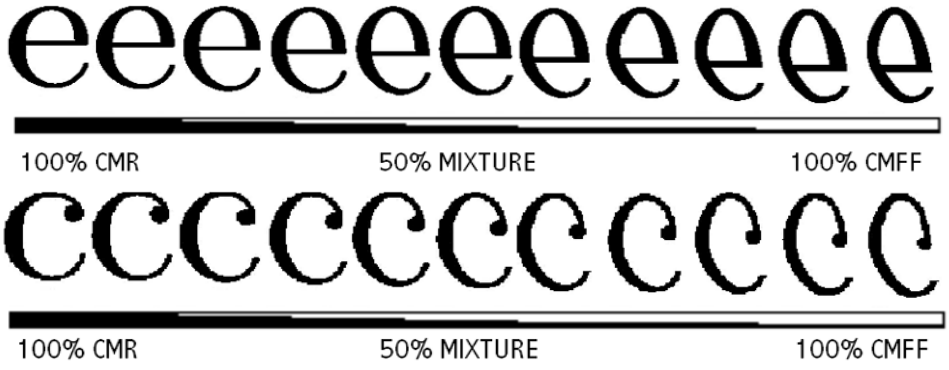
Training data:

Pure Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Pure Seed: -S12
Number Samples: 1600
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Interpolated Seed: numb(0-10)
Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	**	**	**	**	*	100	**	**	**	**	**

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Real Test Seed: -S5

Number Samples: 400

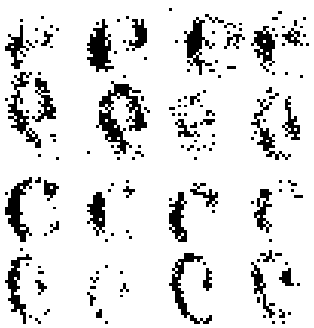
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Interpolated Seed: -S5

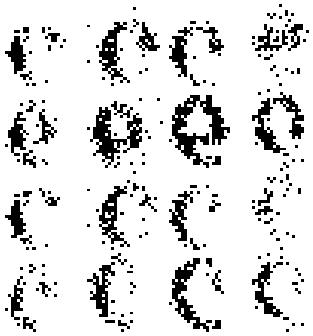
Number Samples: 400

Samples chosen to illustrate data used

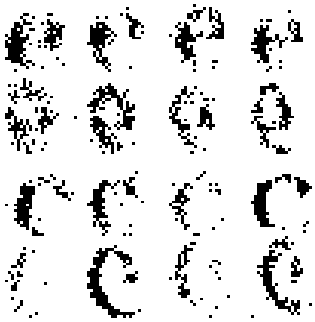
Pure Training Samples



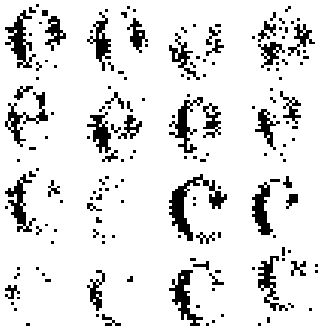
Interpolated Training Samples



Pure Test Samples



Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.


pure/pure e's


pure/pure c's


interp/pure e's


interp/pure c's


pure/interp e's


pure/interp c's


interp/interp e's


interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 67 c: 3	70	e: 66 c: 8	74
	Itrp(B)	e: 39 c: 14	53	e: 40 c: 16	56

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data.

The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 326	o_2 74
		e_1 335	e_2 65
	Itrp(B)	o_3 344	o_4 56
		e_3 335	e_4 65

As explained earlier, the result of our test is $\chi^2 = 2.96$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 330	o_2 70
		e_1 338	e_2 62
	Itrp(B)	o_3 347	o_4 53
		e_3 338	e_4 62

As explained earlier, the result of our test is $\chi^2 = 2.74$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally badly when tested on the real data. They also did equally badly on the interpolated data. Most likely the samples are so distorted that neither classifier could recognize them. Since the interpolated samples were all taken from the midpoint between CMR and CMFF they were very different from what the classifiers had been trained on.

6.2.7 Experimental Description

For this experiment, the generated samples were less blurred, and had a moderate amount of noise added. The varience was less than the last experiment. Training was performed as before and the experiment took the test interpolated data entirely from the midpoint between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e1.8,1.0 -t.3,.110 -s.3,.11

Pure Seed: -S12

Number Samples: 1600

Interpolated Parameters: -e1.8,1.0 -t.3,.11 -s.3,.11

Interpolated Seed: numb (0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	**	**	**	**	*	100	**	**	**	**	**

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Test data: -e1.8,1.0 -t.3,.110 -s.3,.11

Real Seed: -S5

Number Samples: 400

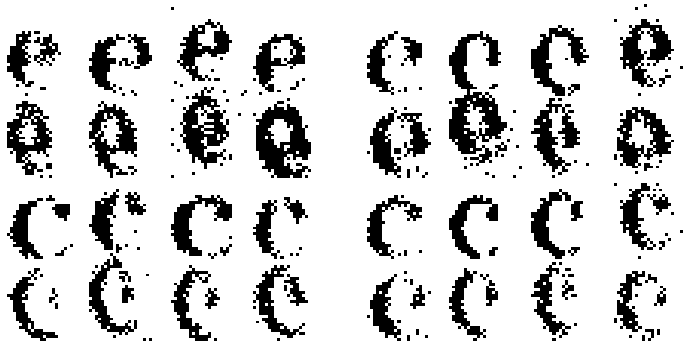
Interpolated Test data: -e1.8,1.0 -t.3,.11 -s.3,.11

Interpolated Test Seed: -S5

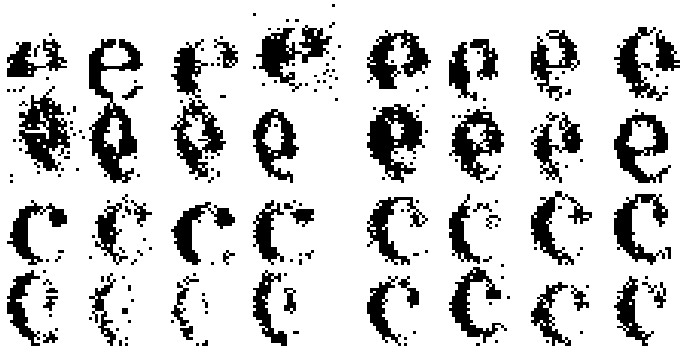
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples

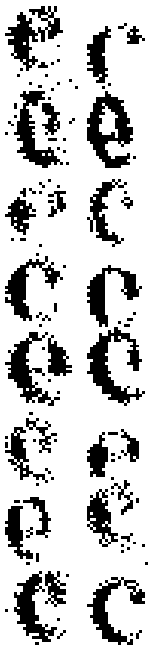


Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON: Pure(A)		TEST ON: Itrp(B)	
TRAIN ON	Pure(A)	e: 7 c: 1	8	e: 48 c: 1	49
	Itrp(B)	e: 9 c: 1	10	e: 5 c: 11	16

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 351	o_2 49
		e_1 367	e_2 33
	Itrp(B)	o_3 384	o_4 16
		e_3 367	e_4 33

As explained earlier, the result of our test is $\chi^2 = 17.97$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 392	o_2 8
		e_1 391	e_2 9
	Itrp(B)	o_3 390	o_4 10
		e_3 391	e_4 9

As explained earlier, the result of our test is $\chi^2 = 0.22$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did much better when tested on both the pure and the interpolated data. The classifier trained on interpolated data did significantly better than the one trained on pure data only when tested on the interpolated data. Both classifiers did equally well when tested on the pure samples.

6.2.8 CMR-CMFF Test Group Results

In this set of experiments we can see that the classifier trained on the mixed (pure and interpolated) samples did much better when tested on the interpolated samples. In cases where the samples were normal and slightly blurred this classfier greatly outperformed the one trained on pure samples only. As the data became more distorted the performance of both classifiers fell off and neither did well. When the classifiers were tested on the pure samples, neither outperformed the other to any significant degree.

Figure 6.2: CMR-CMFF e and c Experimental Results

		Hypothesis 1				Hypothesis 2			
		Errors		Statistic		Errors		Statistic	
Image Quality	Range	AB	BB	χ^2	Rej	AA	BA	χ^2	Rej
normal	full	23	4	10.71	yes	0	0	0.00	no
slightly blurred	full	18	3	7.93	yes	0	2	0.00	no
greatly blurred, high variance	full	102	106	.80	no	91	83	.46	no
greatly blurred, some variance	full	84	83	.10	no	65	73	.54	no
greatly blurred, little variance	full	81	57	5.20	yes	58	64	.32	no
greatly blurred, little variance	mid	74	56	2.96	no	70	53	2.74	no
slightly blurred, little variance	mid	49	16	17.97	yes	8	10	.22	no

6.3 CMR and CMFF I and J Experiments

We next decided to extend our experiments by examining the letter pair i/j. We wanted to see if our findings for the e/c pairs carried over to this letter pair.

6.3.1 Experimental Description

For this experiment, the default parameters were used. The images for both test and training data were only slightly blurred as were the interpolated samples. The test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

6.3. CMR AND CMFF I AND J EXPERIMENTS

Parameters

Training data:

Pure Parameters: default parameters

Pure Seed: no seed

Number Samples: 1600

Interpolated Parameters: default parameters

Interpolated Seed: numb (0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data: Pure parameters: default parameters

Pure Seed: no seed

Number Samples: 400

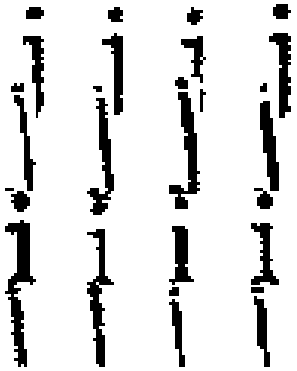
Interpolated Parameters: default parameters

Interpolated Seed: numb (0-10)

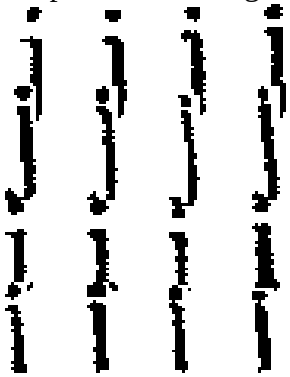
Number Samples: 400

Samples chosen to illustrate data used

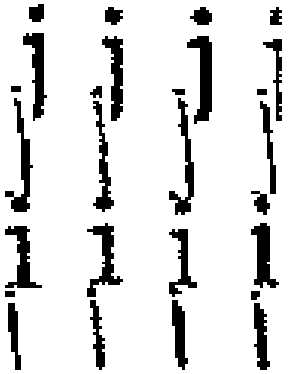
Pure Training Samples



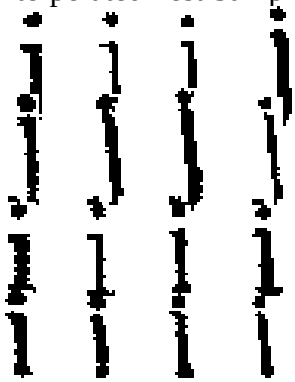
Interpolated Training Samples



Pure Test Samples

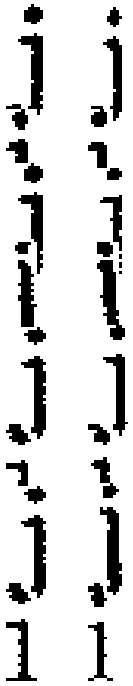


Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



pure/pure j's

pure/pure i's

interp/pure j's

interp/pure i's

pure/interp j's

pure/interp i's

interp/interp j's

interp/interp i's

Error Rates (j's / i's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	j: 0 i: 0	0	j: 0 i: 0	0
	Itrp(B)	j: 0 i: 1	1	j: 0 i: 0	0

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 400	e_2 0
	Itrp(B)	o_3 400	o_4 0
		e_3 400	e_4 0

Clearly there is no difference between the performances of the classifiers so we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 399	e_2 1
	Itrp(B)	o_3 399	o_4 1
		e_3 399	e_4 1

As explained earlier, the result of our test is $\chi^2 = 1.0$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally well when tested on the pure data as well as the interpolated data. There were no significant differences in the performances of the classifiers.

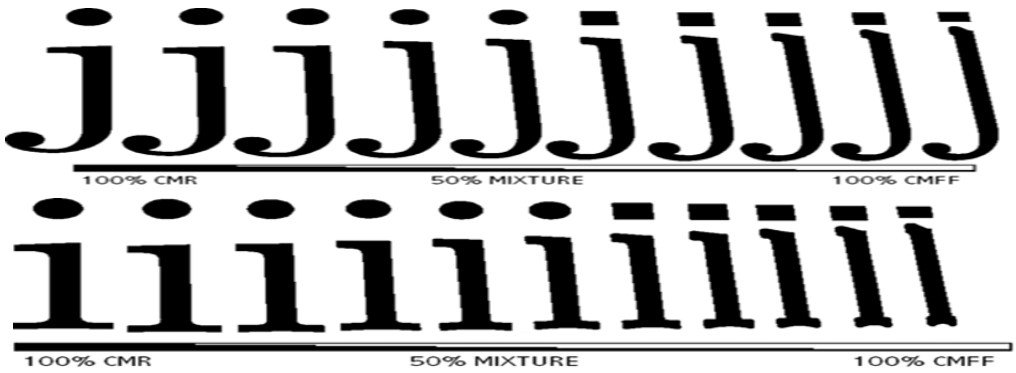
6.3.2 Experimental Description

For this experiment, the generated samples were slighlty more blurred and had a slight amount of noise added. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

Pure Seed: no seed

Number Samples: 1600

Interpolated Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

Interpolated Seed: numb(0-10)

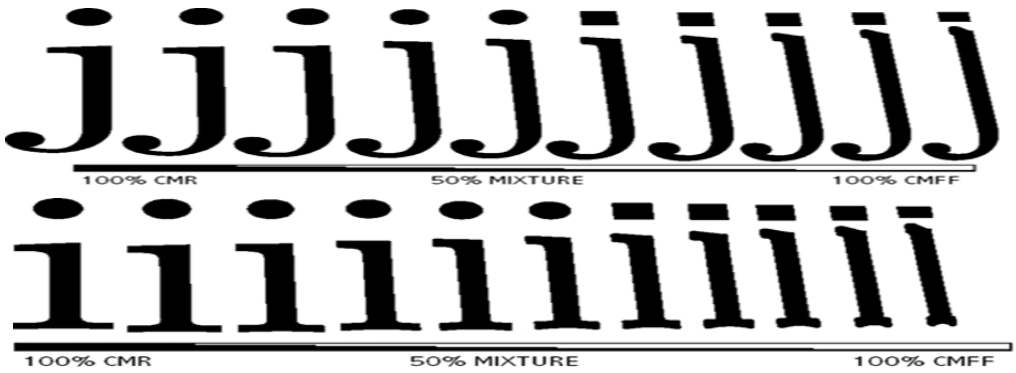
Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

6.3. CMR AND CMFF I AND J EXPERIMENTS



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

Pure Test Seed: no seed

Number Samples: 400

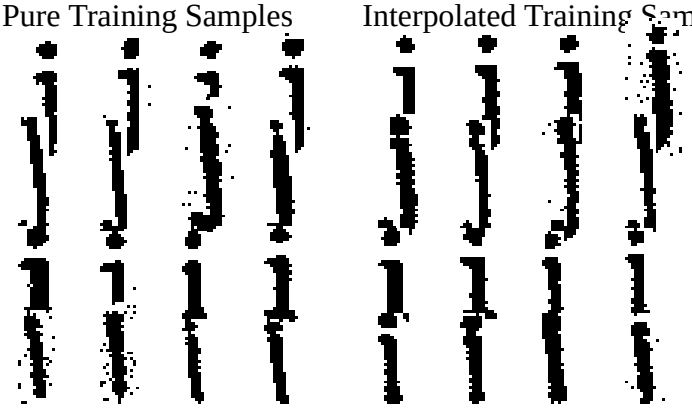
Interpolated Test data: -e1.0,1.1 -t.15,.125 -s.130,.125

Interpolated Test Seed: numb(0-10)

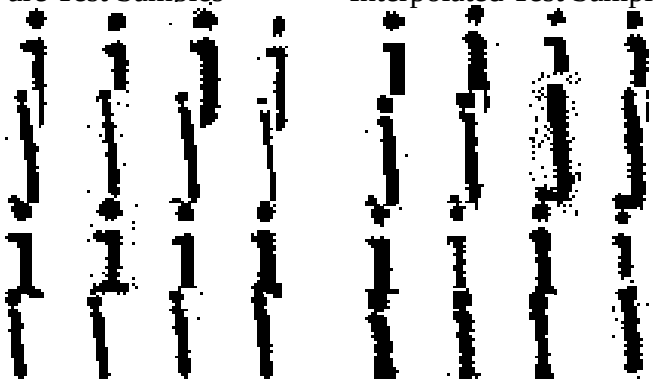
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure j's

pure/pure i's

interp/pure j's

interp/pure i's

pure/interp j's

pure/interp i's

interp/interp j's

interp/interp i's

Error Rates (j's / i's = tot misclassified out of 200)

		TEST ON: Pure(A)		TEST ON: Itrp(B)	
TRAIN ON	Pure(A)	j: 0 i: 2	2	j: 0 i: 1	1
	Itrp(B)	j: 0 i: 3	3	j: 0 i: 1	1

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 399	o_2 1
		e_1 399	e_2 1
	Itrp(B)	o_3 399	o_4 1
		e_3 399	e_4 1

As explained earlier, the result of our test is $\chi^2 = 2.0$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 398	o_2 2
		e_1 397	e_2 3
	Itrp(B)	o_3 397	o_4 3
		e_3 397	e_4 3

As explained earlier, the result of our test is $\chi^2 = 0.33$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, once again, both classifiers did well on the pure data with no significant difference. There was no siginificant difference in their performance on the interpolated data.

6.3.3 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. A seed of 2 was added to the real data. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

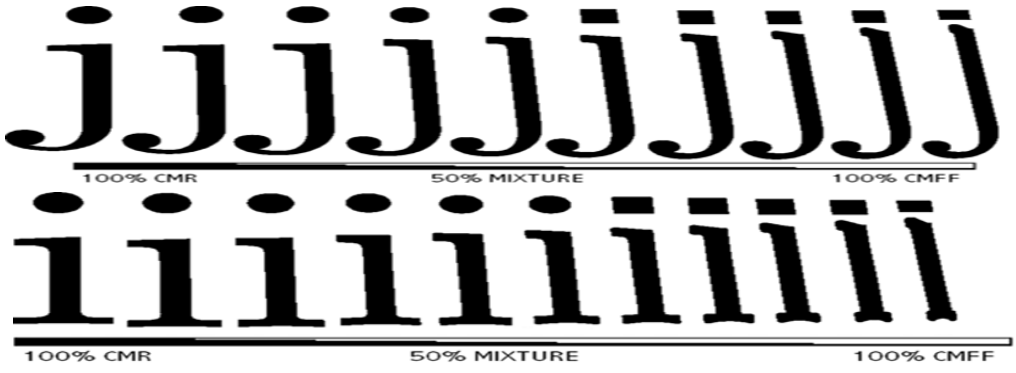
Training data:

Pure Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130
Pure Seed: -S2
Number Samples: 1600
Interpolated Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130
Interpolated Seed: numb(0-10)
Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Pure Seed: -S2

Number Samples: 400

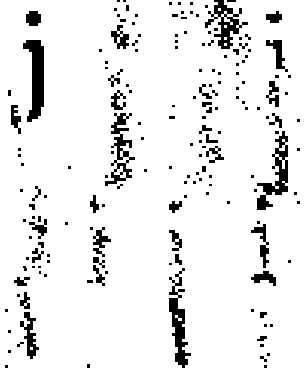
Interpolated Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Interpolated Seed: numb(0-10)

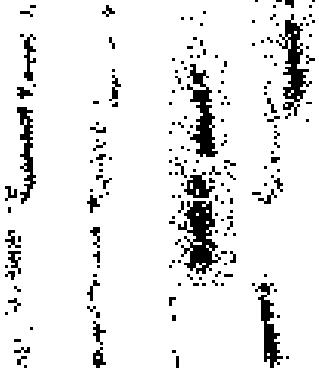
Number Samples: 400

Samples chosen to illustrate data used

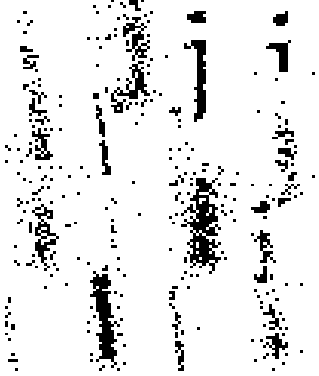
Pure Training Samples



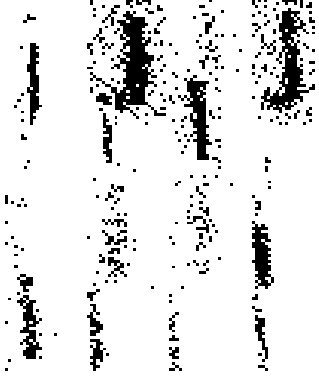
Interpolated Training Samples



Pure Test Samples



Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure j's

pure/pure i's

interp/pure j's

interp/pure i's

pure/interp j's

pure/interp i's

interp/interp j's

interp/interp i's

Error Rates (j's / i's = tot misclassified out of 200)

		TEST ON: Pure(A)		TEST ON: Itrp(B)	
TRAIN ON	Pure(A)	j: 15 i: 68	83	j: 16 i: 55	71
	Itrp(B)	j: 20 i: 76	96	j: 13 i: 52	65

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 329	o_2 71
		e_1 332	e_2 68
	Itrp(B)	o_3 335	o_4 65
		e_3 332	e_4 68

As explained earlier, the result of our test is $\chi^2 = 0.30$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 317	o_2 83
		e_1 310	e_2 90
	Itrp(B)	o_3 304	o_4 96
		e_3 310	e_4 90

As explained earlier, the result of our test is $\chi^2 = 1.20$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

Both classifiers did badly when tested on the interpolated data as well as the pure data. The data was so distorted that letters were difficult to recognize by either classifier. Neither classifier did significantly different on either of the types of test data, possibly because the letters were so distorted.

6.3.4 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, the variance was limited to half of what it was in the previous experiment. Real training data had a seed of 5, while the real test data had a seed of 11. The interpolated training data had a variable seed, and the interpolated training data had a different variable seed. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Pure Seed: -S5

Number Samples: 1600

Interpolated Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

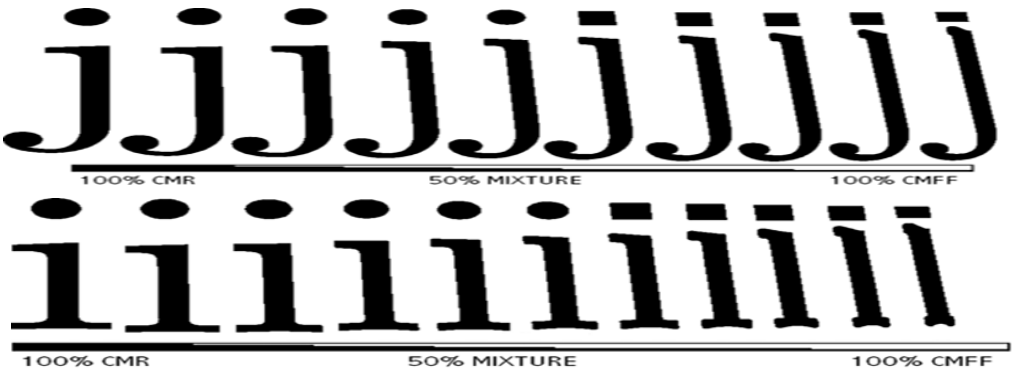
Interpolated Seed: Snumb (0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Pure Seed: -S11

Number Samples: 400

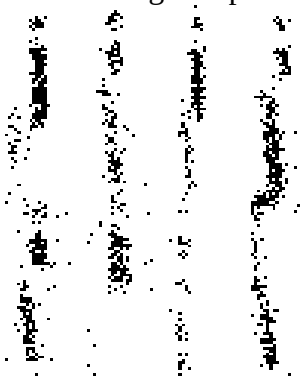
Interpolated Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Interpolated Test Seed: numb * 3 (0, 3, 6....30)

Number Samples: 400

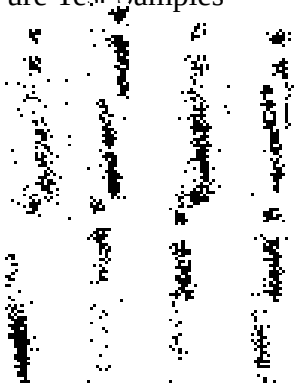
Samples chosen to illustrate data used

Pure Training Samples



Interpolated Training Samples

Pure Test Samples



Interpolated Test Samples

Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure j's

pure/pure i's

interp/pure j's

interp/pure i's

pure/interp j's

pure/interp i's

interp/interp j's

interp/interp i's

Error Rates (j's / i's = tot misclassified out of 200)

		TEST ON: Pure(A)		TEST ON: Itrp(B)	
TRAIN ON	Pure(A)	j: 10 i: 84	94	j: 12 i: 60	72
	Itrp(B)	j: 8 i: 85	93	j: 10 i: 54	64

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 328	o_2 72
		e_1 332	e_2 68
	Itrp(B)	o_3 336	o_4 64
		e_3 332	e_4 68

As explained earlier, the result of our test is $\chi^2 = 0.54$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 306	o_2 94
		e_1 306	e_2 94
	Itrp(B)	o_3 307	o_4 93
		e_3 306	e_4 94

As explained earlier, the result of our test is $\chi^2 = 0.1$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally when tested on the pure data. They each misclassified about 25 per cent of the samples. The classifier trained on the interpolated data did somewhat better than the one trained on the real data when tested on the interpolated data, however not significantly so.

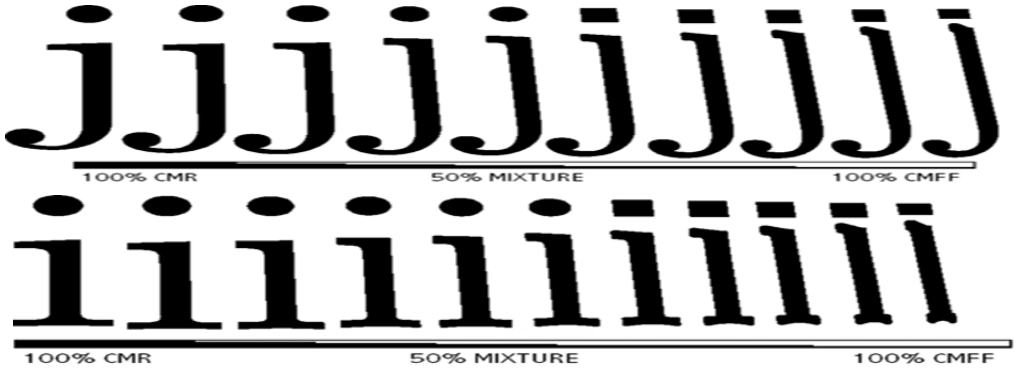
6.3.5 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, variance was decreased to slightly above the default levels. Seeds were kept the same as in the last test. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Pure Seed: -S5

Number Samples: 1600

Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

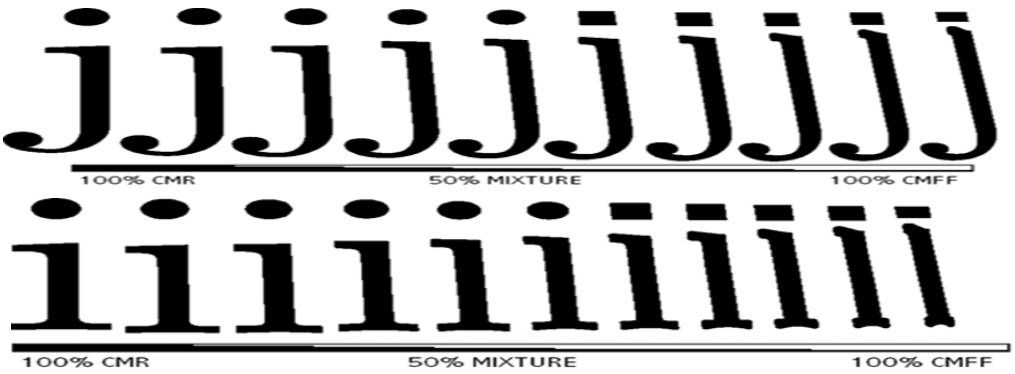
Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50

Interp	10	10	10	10	10	**	10	10	10	10	10

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

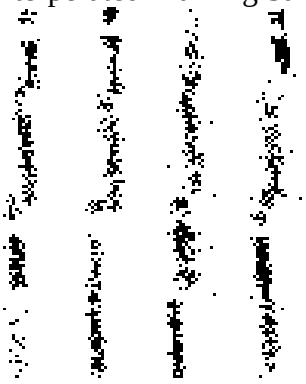
Test data:

Real Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Real Test Seed: -S11
Number Samples: 400
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Interpolated Seed: numb * 2 (0,2,4...20)
Number Samples: 400

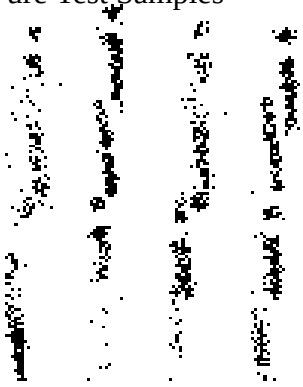
Samples chosen to illustrate data used

Pure Training Samples

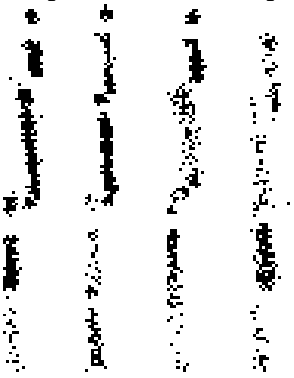
Interpolated Training Samples



Pure Test Samples



Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure j's

pure/pure i's

interp/pure j's

interp/pure i's

pure/interp j's

pure/interp i's

interp/interp j's

interp/interp i's

Error Rates (j's / i's = tot misclassified out of 200)

		TEST ON: Pure(A)		TEST ON: Itrp(B)	
TRAIN ON	Pure(A)	j: 5 i: 80	85	j: 4 i: 54	58
	Itrp(B)	j: 2 i: 85	87	j: 4 i: 49	53

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 342	o_2 58
		e_1 344	e_2 56
	Itrp(B)	o_3 347	o_4 53
		e_3 344	e_4 56

As explained earlier, the result of our test is $\chi^2 = 0.26$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 315	o_2 85
		e_1 314	e_2 86
	Itrp(B)	o_3 313	o_4 87
		e_3 314	e_4 86

As explained earlier, the result of our test is $\chi^2 = 0.2$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers performed equally when tested on the pure data as well as when tested on the interpolated data.

6.3.6 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, variance was slight. This differed from the last experiment in that the interpolated samples were all taken from the midpoint between CMR and CMFF.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

6.3. CMR AND CMFF I AND J EXPERIMENTS

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Pure Seed: -S12

Number Samples: 1600

Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

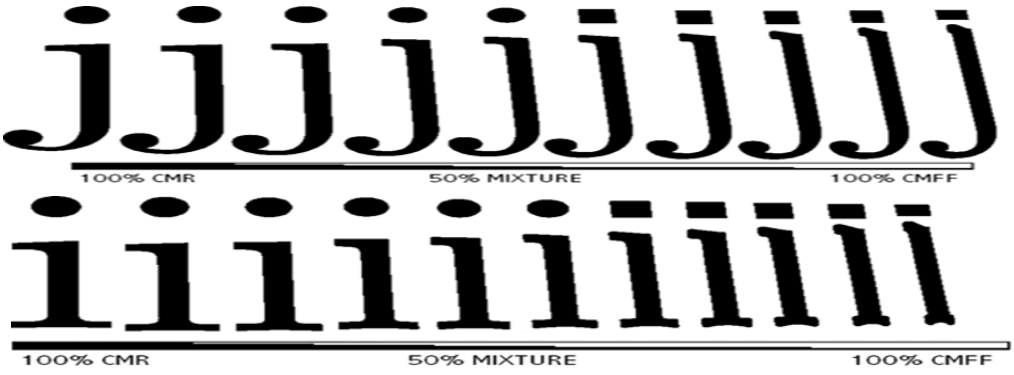
Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	**	**	**	**	*	100	**	**	**	**	**

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Real Test Seed: -S5

Number Samples: 400

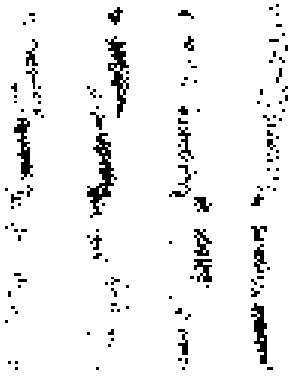
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Interpolated Seed: -S5

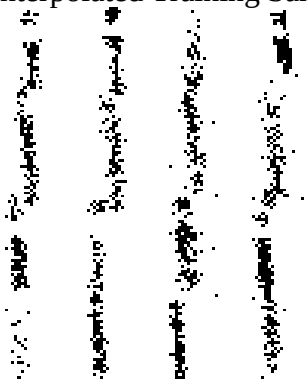
Number Samples: 400

Samples chosen to illustrate data used

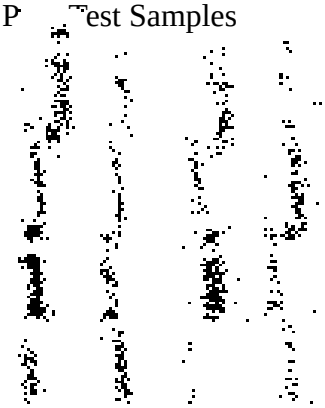
Pure Training Samples



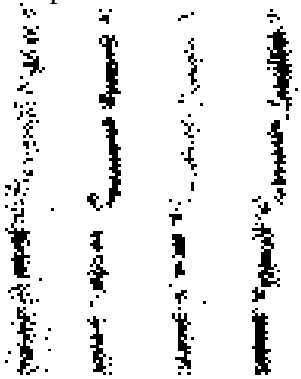
Interpolated Training Samples



Pure Test Samples

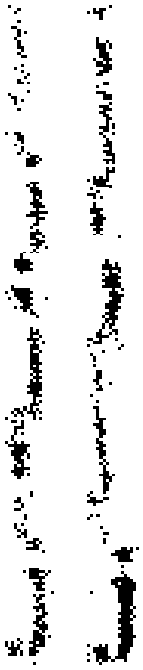


Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



pure/pure j's

pure/pure i's

interp/pure j's

interp/pure i's

pure/interp j's

pure/interp i's

interp/interp j's

interp/interp i's

Error Rates (j's / i's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	j: 3 i: 83	86	j: 8 i: 51	59
	Itrp(B)	j: 5 i: 92	97	j: 2 i: 42	44

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 341	o_2 59
		e_1 348	e_2 52
	Itrp(B)	o_3 356	o_4 44
		e_3 348	e_4 52

As explained earlier, the result of our test is $\chi^2 = 2.49$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 314	o_2 86
		e_1 308	e_2 92
	Itrp(B)	o_3 303	o_4 97
		e_3 308	e_4 92

As explained earlier, the result of our test is $\chi^2 = 0.85$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally badly when tested on the pure data. They also did equally badly on the interpolated data. The classifier trained on interpolated data did slightly better than the one trained on pure data, but not significantly so.

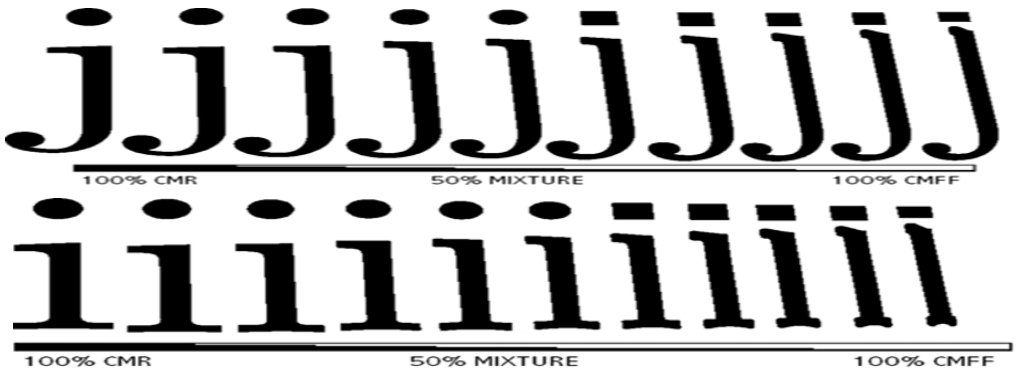
6.3.7 Experimental Description

For this experiment, the generated samples were less blurred, and had a moderate amount of noise added. The varience was less than the last experiment. Training was performed as before and the experiment took the test interpolated data entirely from the midpoint between the real CMR and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures ranging from 10-90% of each font.

Training Data.



Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	10	10	10	10	10	00	10	10	10	10	10

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e1.8,1.0 -t.3,.110 -s.3,.11

Pure Seed: -S12

Number Samples: 1600

Interpolated Parameters: -e1.8,1.0 -t.3,.11 -s.3,.11

Interpolated Seed: numb (0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.

6.3. CMR AND CMFF I AND J EXPERIMENTS

Pure	50	**	**	**	**	**	**	**	**	**	50
Interp	**	**	**	**	*	100	**	**	**	**	**

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Test data: -e1.8,1.0 -t.3,.110 -s.3,.11

Real Seed: -S5

Number Samples: 400

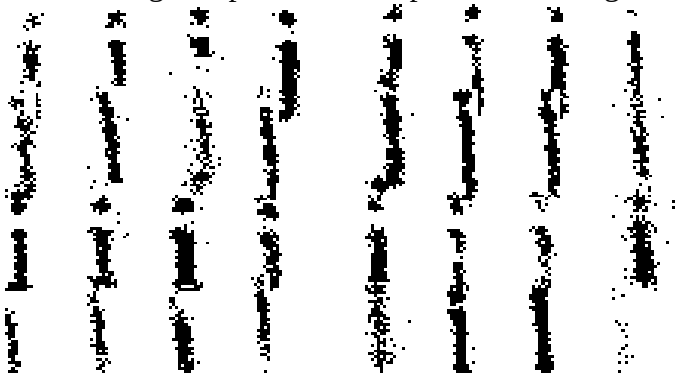
Interpolated Test data: -e1.8,1.0 -t.3,.11 -s.3,.11

Interpolated Test Seed: -S5

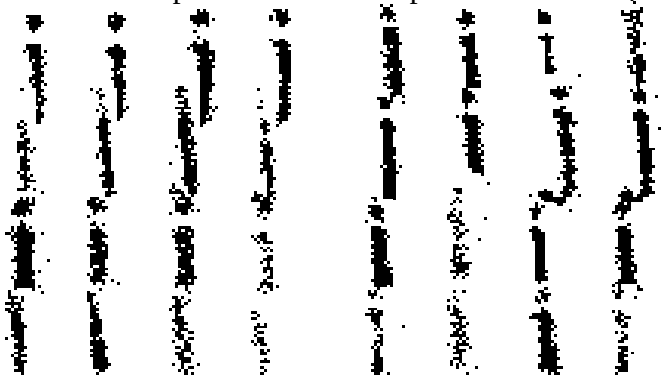
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure j's

pure/pure i's

interp/pure j's

interp/pure i's

pure/interp j's

pure/interp i's

interp/interp j's

interp/interp i's

Error Rates (j's / i's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON Pure(A)		j: 0 i: 29	29	j: 0 i: 10	10
TRAIN ON Itrp(B)		j: 4 i: 33	37	j: 0 i: 10	10

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual observed counts are below.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 390	o_2 10
		e_1 390	e_2 10
	Itrp(B)	o_3 390	o_4 10
		e_3 390	e_4 10

Clearly there is no difference between the performances of the classifiers so we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 371	o_2 29
		e_1 367	e_2 33
	Itrp(B)	o_3 363	o_4 37
		e_3 367	e_4 33

As explained earlier, the result of our test is $\chi^2 = 1.4$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did much better when tested on both the pure and the interpolated data, however neither classifier showed a significant difference.

6.3.8 CMR-CMFF (i and j)

When we look at the following chart, we can see that there appears to be no statistical difference between the classifiers when tested on either the pure only samples, or the mixed samples. This supports our hypothesis that training on interpolated data does not hurt the clasifier.

Figure 6.3: CMR-CMFF i and j Experimental Results

		Hypothesis 1				Hypothesis 2			
			Errors	Statistic		Errors		Statistic	
Image Quality	Range	AB	BB	χ^2	Rej	AA	BA	χ^2	Rej
normal	full	0	0	0.00	no	0	1	1.00	no
slightly blurred	full	2	3	2.00	no	1	1	.33	no
greatly blurred, high variance	full	71	65	.3	no	83	96	1.20	no
greatly blurred, some variance	full	72	64	.54	no	94	93	.10	no
greatly blurred, little variance	full	58	53	.26	no	85	87	.20	no
greatly blurred, little variance	mid	59	44	2.49	no	86	97	.85	no
slightly blurred, little variance	mid	10	10	0.0	no	29	37	1.40	no

6.4 Three Way CMR/CMFF/CMSS Experiments

We next thought it would be interesting to construct a 3-way interpolation between 3 dissimilar fonts, thus extending our interpolation space into 3 dimensions from 2 dimensions.

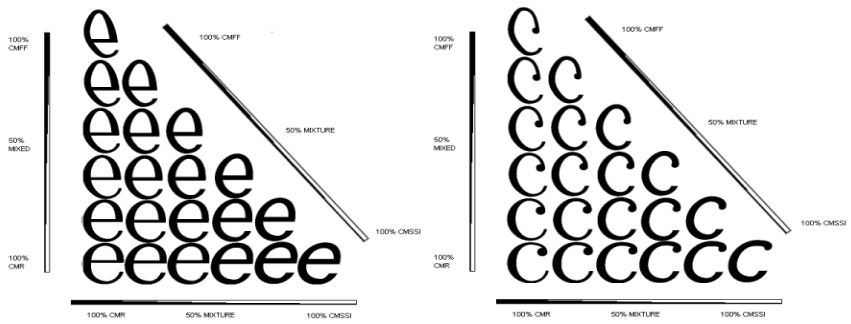
6.4.1 Experimental Description

For this experiment, the generated samples were not blurred and had little noise. A variable seed was used for the interpolated data while the real data had no seed. One third of the pure training samples were taken from each of the pure fonts (CMR, CMFF and CMSSI). The test interpolated data was taken from the entire range between the real CMR, CMFF, and CMSSI fonts, with approximately 6 percent from each interpolation.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures of the three fonts.

Training Data.



Pur	33					
	**	**				
	**	**	**			
	**	**	**	**		
	**	**	**	**	**	
	33	**	**	**	**	33
Int	**					
	06	06				
	06	06	06			
	06	**	**	06		
	06	06	06	06	06	
	**	06	06	06	06	**

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

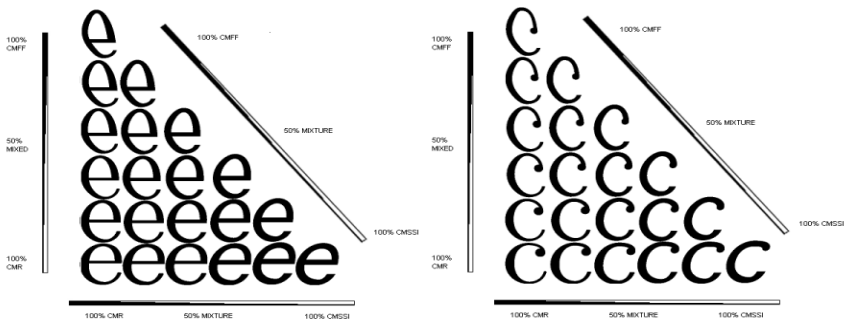
Training data:

Pure Parameters: default parameters
Pure Seed: no seed
Number Samples: 1800
Interpolated Parameters: default parameters
Interpolated Seed: numb (0-10)
Number Samples: 480

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pur 33

** **

** ** **

** ** ** **

** ** ** ** **

33 ** ** ** ** 33

Int **

06 06

06 06 06

06 ** ** 06

06 06 06 06 06

** 06 06 06 06 **

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

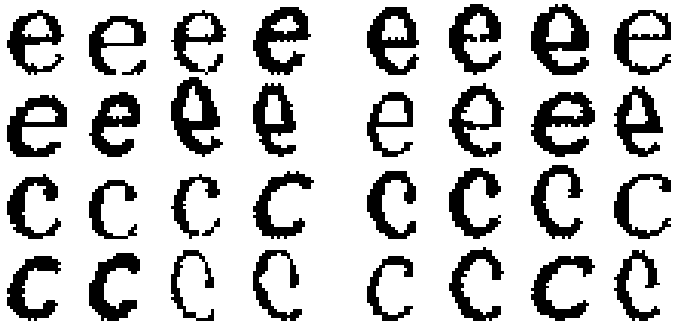
Parameters

Test data:

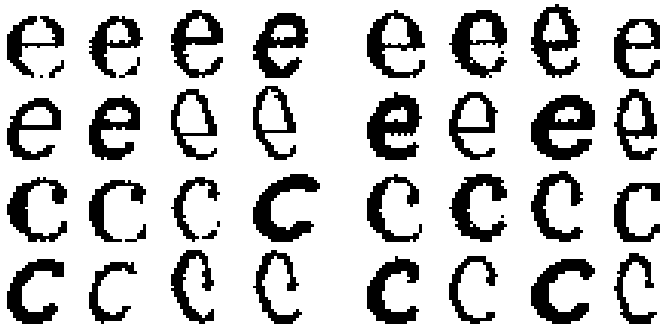
Pure parameters: default parameters
Pure Seed: no seed
Number Samples: 1800
Interpolated Parameters: default parameters
Interpolated Seed: numb (0-10)
Number Samples: 480

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 0 c: 0	0	e: 40 c: 2	42
	Itrp(B)	e: 0 c: 0	0	e: 3 c: 0	3

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual counts observed are listed below

		Right	Wrong
TRAIN ON	Pure(A)	o_1 358	o_2 42
		e_1 377	e_2 23
	Itrp(B)	o_3 397	o_4 3
		e_3 377	e_4 23

As explained earlier, the result of our test is $\chi^2 = 31.57$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 400	e_2 0
	Itrp(B)	o_3 400	o_4 0
		e_3 400	e_4 0

Clearly there is no difference between the performances of the classifiers so we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, unexpectedly, the classifier trained on the interpolated data did significantly better than the one trained on only pure data when tested on the interpolated data. However, the classifier trained on the interpolated data performed equally with the one trained on the pure data when tested on the pure data.

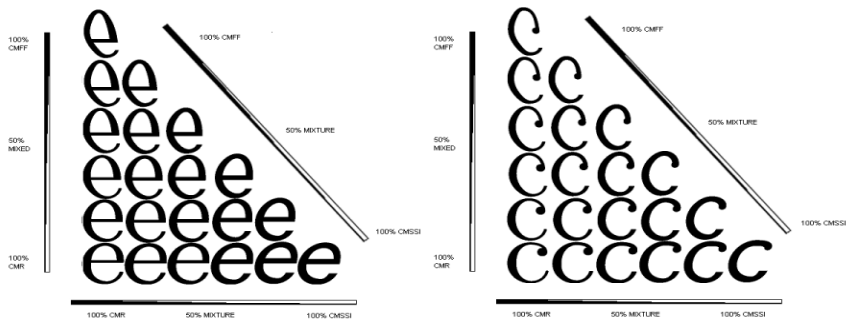
6.4.2 Experimental Description

For this experiment, the generated samples were slightly more blurred, and had a small amount of noise added. No seed was used for the pure data howver a variable seed was used for the interpolated data. The test interpolated data was taken from the entire range among the real CMR,CMFF and CMSSI fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures of the three fonts.

Training Data.



Pur	33					
	**	**				
	**	**	**			
	**	**	**	**		
	**	**	**	**	**	
	33	**	**	**	**	33
Int	**					
	06	06				
	06	06	06			
	06	**	**	06		
	06	06	06	06	06	
	**	06	06	06	06	**

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

Pure Seed: no seed

Number Samples: 1800

Interpolated Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125

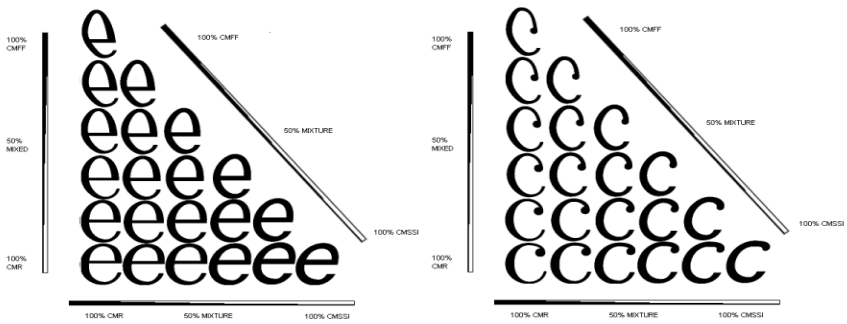
Interpolated Seed: numb(0-10)

Number Samples: 1800

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pur 33

** **

** ** **

** ** ** **

** ** ** ** **

33 ** ** ** ** 33

Int **

06 06

06 06 06

06 ** ** 06

06 06 06 06 06

** 06 06 06 06 **

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

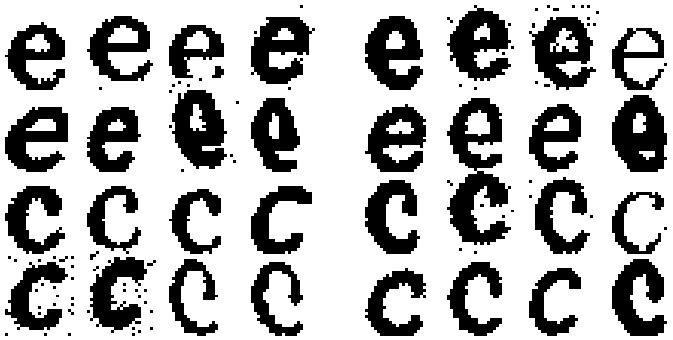
Parameters

Test data:

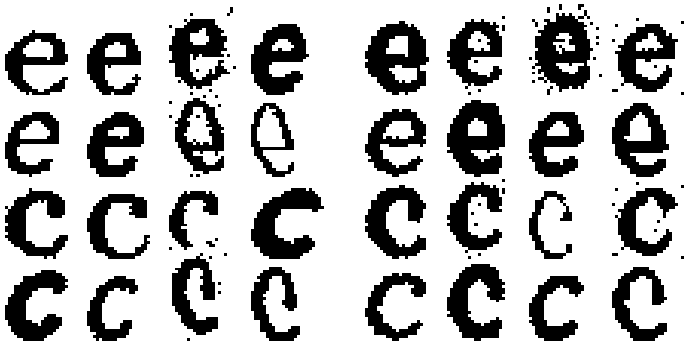
Pure Parameters: -e1.0,1.1 -t.15,.125 -s.130,.125
Pure Test Seed: no seed
Number Samples: 480
Interpolated Test data: -e1.0,1.1 -t.15,.125 -s.130,.125
Interpolated Test Seed: numb(0-10)
Number Samples: 480

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 0 c: 0	0	e: 17 c: 1	18
	Itrp(B)	e: 0 c: 0	0	e: 0 c: 0	0

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual counts observed are listed below

		Right	Wrong
TRAIN ON	Pure(A)	o_1 382	o_2 18
		e_1 391	e_2 9
	Itrp(B)	o_3 400	o_4 0
		e_3 391	e_4 9

As explained earlier, the result of our test is $\chi^2 = 14.54$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 400	o_2 0
		e_1 400	e_2 0
	Itrp(B)	o_3 400	o_4 0
		e_3 400	e_4 0

Clearly there is no difference between the performances of the classifiers so we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, unexpectedly, the classifier trained on the interpolated data did significantly better than the one trained on only pure data when tested on the interpolated data. The classifiers performed equally when tested on the pure data.

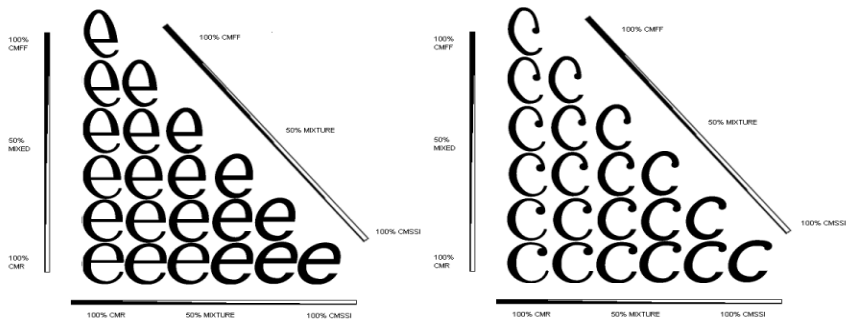
6.4.3 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. A seed fo 2 was used for the real data, while the interpolated data had a variable seed. Training was performed as before while the test interpolated data was taken from the entire range between the real CMR, CMFF and CMSSI fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures of the three fonts.

Training Data.



Pur	33					
	**	**				
	**	**	**			
	**	**	**	**		
	**	**	**	**	**	
	33	**	**	**	**	33
Int	**					
	06	06				
	06	06	06			
	06	**	**	06		
	06	06	06	06	06	
	**	06	06	06	06	**

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

Pure Seed: -S2

Number Samples: 1600

Interpolated Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130

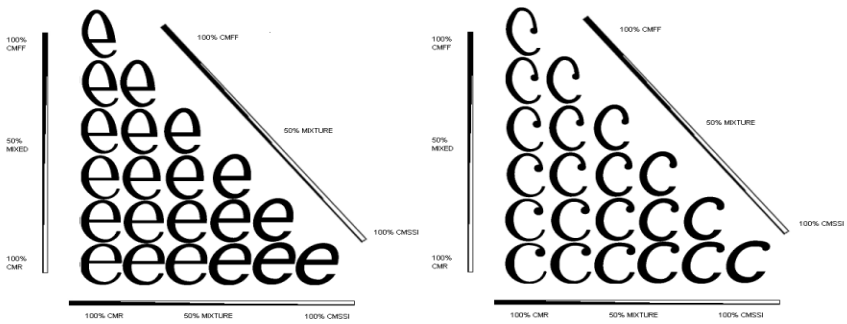
Interpolated Seed: numb(0-10)

Number Samples: 1600

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pur 33

** **

** ** **

** ** ** **

** ** ** ** **

33 ** ** ** ** 33

Int **

06 06

06 06 06

06 ** ** 06

06 06 06 06 06

** 06 06 06 06 **

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

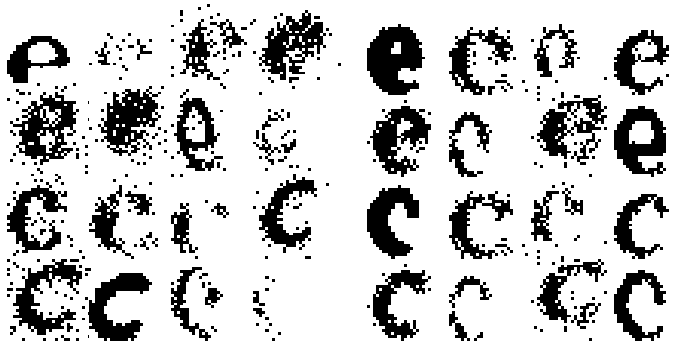
Parameters

Test data:

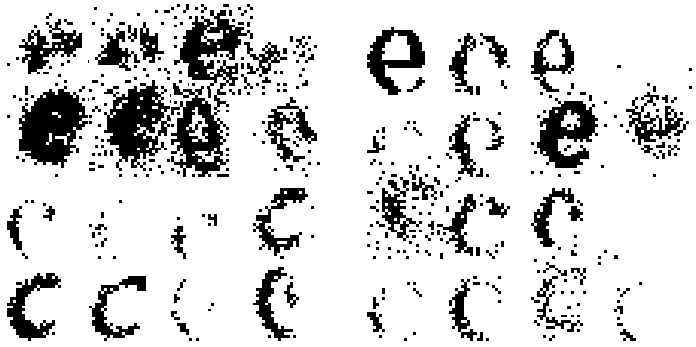
Pure Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130
Pure Seed: -S2
Number Samples: 400
Interpolated Parameters: -e2.0,2.1 -t.4,.4 -s.4,.4130
Interpolated Seed: numb(0-10)
Number Samples: 400

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples

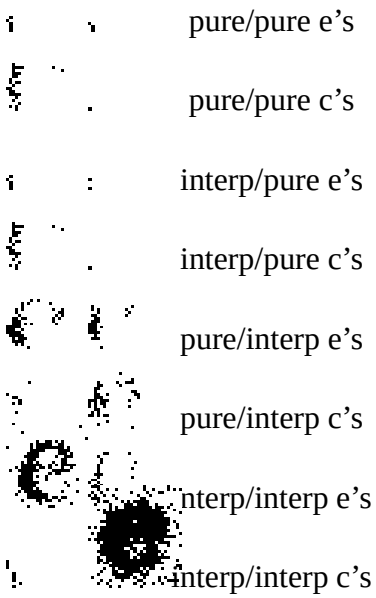


Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 36 c: 55	91	e: 62 c: 49	111
	Itrp(B)	e: 24 c: 68	92	e: 48 c: 63	111

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual counts observed are listed below

		Right	Wrong
TRAIN ON	Pure(A)	o_1 289	o_2 111
		e_1 289	e_2 111
	Itrp(B)	o_3 289	o_4 111
		e_3 289	e_4 111

Clearly there is no difference between the performances of the classifiers so we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 309	o_2 91
		e_1 308	e_2 92
	Itrp(B)	o_3 308	o_4 92
		e_3 308	e_4 92

As explained earlier, the result of our test is $\chi^2 = 0.1$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers performed the same when tested on both interpolated and pure data. Both performed badly, perhaps because the characters were so badly distorted.

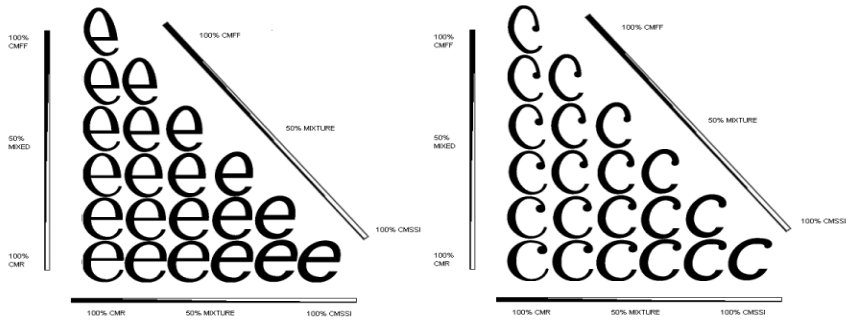
6.4.4 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, the variance was limited to half of what it was in the previous experiment. Real training data had a seed of 5, while the real test data had a seed of 11. The interpolated training data had a variable seed, and the interpolated test data had a different variable seed. Training was performed as before while the test interpolated data was taken from almost the entire range between the real CMR, CMFF and CMSSI fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures of the three fonts.

Training Data.



Pur 33

**	**				
**	**	**			
**	**	**	**		
**	**	**	**	**	
33	**	**	**	**	33

Int **

06	06				
06	06	06			
06	**	**	06		
06	06	06	06	06	
**	06	06	06	06	**

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

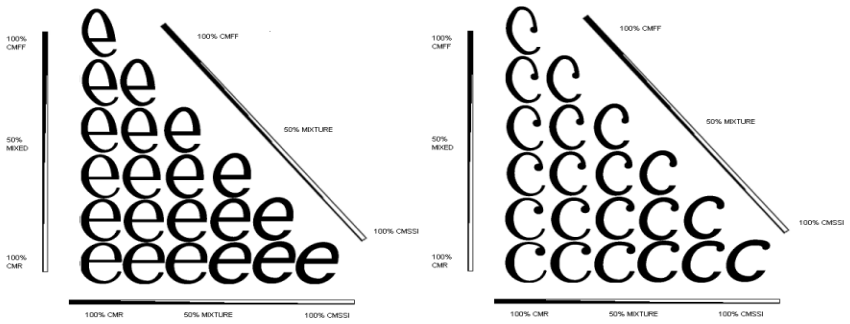
6.4. THREE WAY CMR/CMFF/CMSS EXPERIMENTS

Pure Seed: -S5
Number Samples: 1800
Interpolated Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2
Interpolated Seed: Snumb (0-10)
Number Samples: 1800

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pur 33

** **

** ** **

** ** ** **

** ** ** ** **

33 ** ** ** ** 33

Int **

06 06

06 06 06

06 ** ** 06

06 06 06 06 06

** 06 06 06 06 **

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Pure Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Pure Seed: -S11

Number Samples: 480

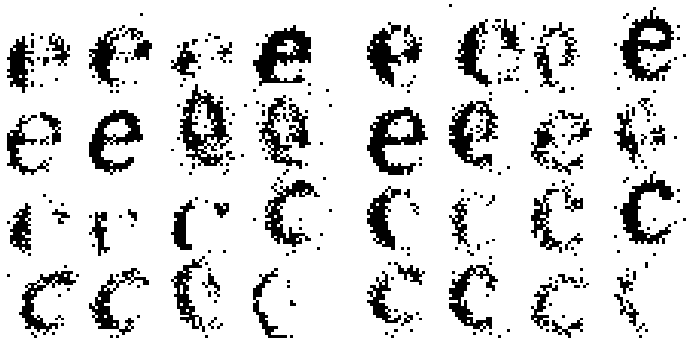
Interpolated Parameters: -e2.0,1.1 -t.4,.2 -s.4,.2

Interpolated Test Seed: numb * 3 (0, 3, 6....30)

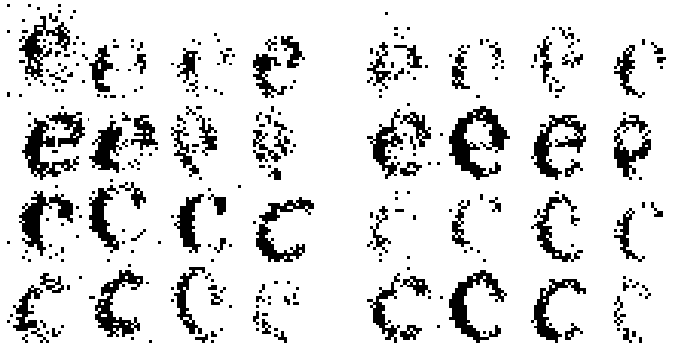
Number Samples: 480

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		Itrp(B)	
TRAIN ON	Pure(A)	e: 44 c: 20	64	e: 89 c: 20	109
	Itrp(B)	e: 46 c: 26	72	e: 30 c: 18	48

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual counts observed are listed below

		Right	Wrong
TRAIN ON	Pure(A)	o_1 291	o_2 109
		e_1 321	e_2 79
	Itrp(B)	o_3 352	o_4 48
		e_3 321	e_4 79

As explained earlier, the result of our test is $\chi^2 = 29.34$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 336	o_2 64
		e_1 332	e_2 68
	Itrp(B)	o_3 328	o_4 72
		e_3 332	e_4 68

As explained earlier, the result of our test is $\chi^2 = 0.54$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally badly when tested on the pure data. They each misclassified about 30 per cent of the samples. The classifier trained on the interpolated data did significantly better than the one trained on the pure data when tested on the interpolated data.

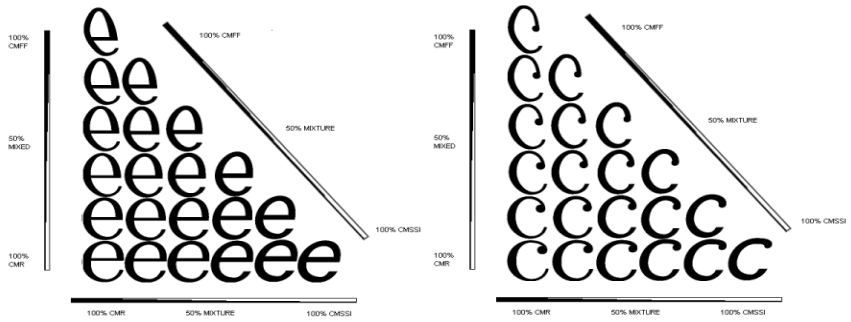
6.4.5 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, variance was decreased to slightly above the default levels. Seeds were random for the interpolated training data, and a different random seed was used for the interpolated test data. Training was performed as before while the test interpolated data was taken from the entire range among the real CMR, CMSSI and CMFF fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures of the three fonts.

Training Data.



Pur 33

** **

** ** **

** ** ** **

** ** ** ** **

33 ** ** ** ** 33

Int **

06 06

06 06 06

06 ** ** 06

06 06 06 06 06

** 06 06 06 06 **

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

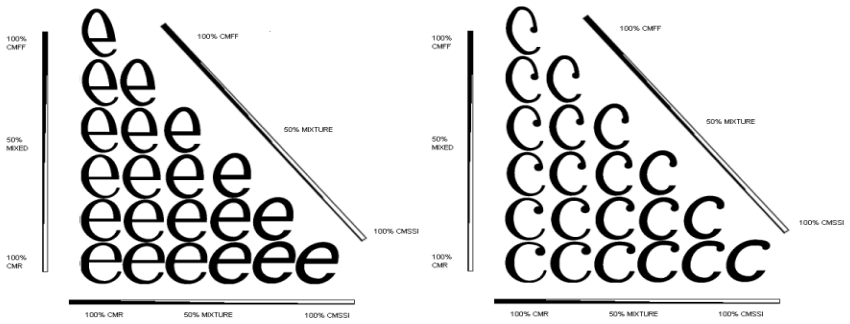
6.4. THREE WAY CMR/CMFF/CMSS EXPERIMENTS

Pure Seed: -S5
Number Samples: 1800
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Interpolated Seed: numb(0-10)
Number Samples: 1800

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pur 33

** **

** ** **

** ** ** **

** ** ** ** **

33 ** ** ** ** 33

Int **

06 06

06 06 06

06 ** ** 06

06 06 06 06 06

** 06 06 06 06 **

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Real Test Seed: -S11

Number Samples: 480

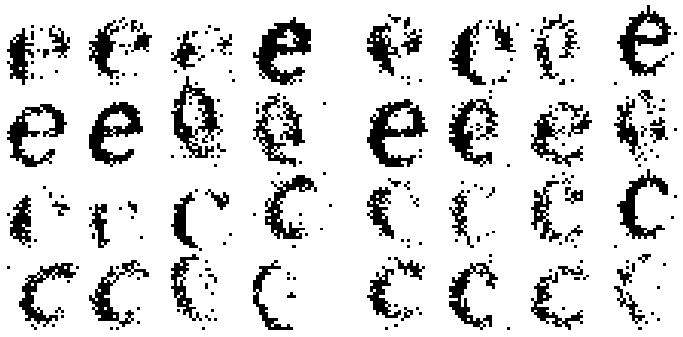
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Interpolated Seed: numb * 2 (0,2,4...20)

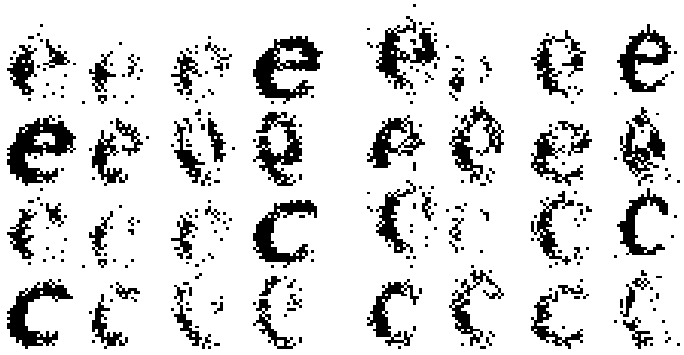
Number Samples: 480

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples

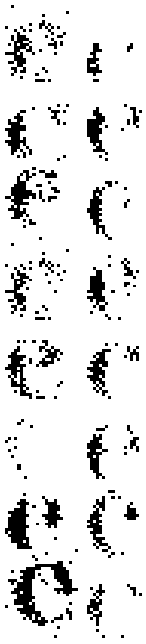


Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 49 c: 5	54	e: 80 c: 6	86
	Itrp(B)	e: 59 c: 11	70	e: 39 c: 16	55

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual counts observed are listed below

		Right	Wrong
TRAIN ON	Pure(A)	o_1 314	o_2 86
		e_1 329	e_2 71
	Itrp(B)	o_3 345	o_4 55
		e_3 329	e_4 71

As explained earlier, the result of our test is $\chi^2 = 8.21$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 346	o_2 54
		e_1 338	e_2 62
	Itrp(B)	o_3 330	o_4 70
		e_3 338	e_4 62

As explained earlier, the result of our test is $\chi^2 = 2.42$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers performed equally when tested on the pure data. When tested on the interpolated data, the classifier trained on pure data did significantly worse than the one trained on the interpolated data.

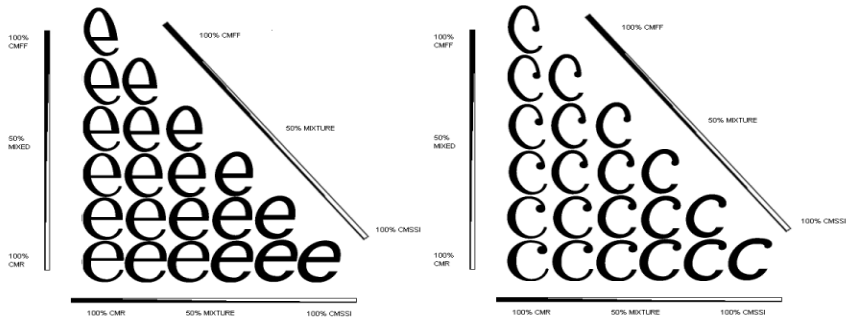
6.4.6 Experimental Description

For this experiment, the generated samples were greatly blurred, and had a large amount of noise added. However, variance was slight. This differed from the last experiment in that the interpolated test samples were all taken from the midpoint between CMR, CMFF and CMSSI.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures of the three fonts.

Training Data.



Pur	33					
	**	**				
	**	**	**			
	**	**	**	**		
	**	**	**	**	**	
	33	**	**	**	**	33
Int	**					
	06	06				
	06	06	06			
	06	**	**	06		
	06	06	06	06	06	
	**	06	06	06	06	**

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

Pure Seed: -S12

Number Samples: 1800

Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125

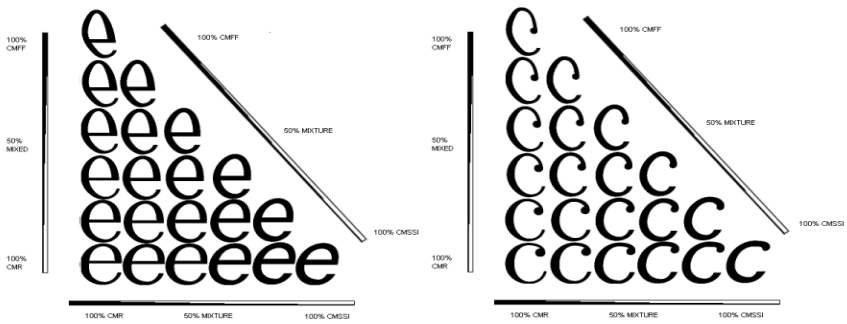
Interpolated Seed: numb2 + numb (numb2 = 0,2,4...10), (numb = 0,2,4...10)

Number Samples: 1800

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pur 33

** **

** ** **

** ** ** **

** ** ** ** **

33 ** ** ** ** 33

Int **

** **

** ** **

** 50 50 **

** ** ** ** **

** ** ** ** ** **

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

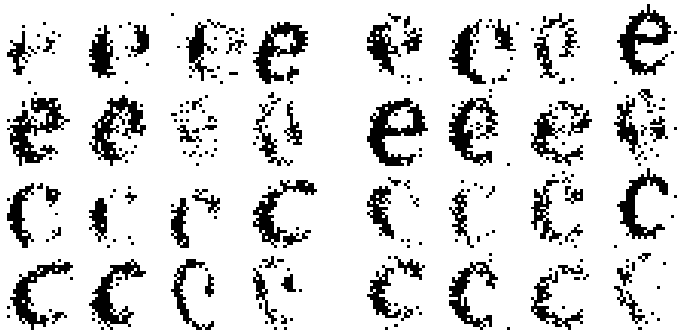
Parameters

Test data:

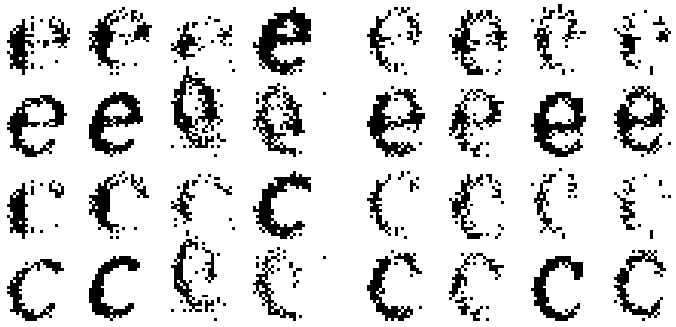
Real Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Real Test Seed: -S5
Number Samples: 480
Interpolated Parameters: -e2.0,1.1 -t.4,.125 -s.4,.125
Interpolated Seed: (6, 10)
Number Samples: 480

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples

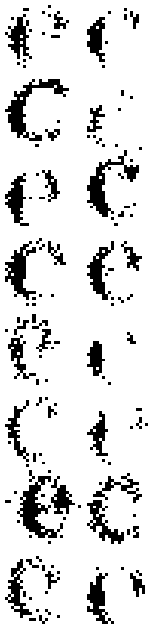


Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.



pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 66 c: 1	67	e: 49 c: 8	57
	Itrp(B)	e: 40 c: 12	52	e: 17 c: 5	22

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual counts observed are listed below

		Right	Wrong
TRAIN ON	Pure(A)	o_1 343	o_2 57
		e_1 360	e_2 40
	Itrp(B)	o_3 378	o_4 22
		e_3 360	e_4 40

As explained earlier, the result of our test is $\chi^2 = 17.2$. Since this result is greater than or equal to 3.84 we can reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 333	o_2 67
		e_1 340	e_2 60
	Itrp(B)	o_3 348	o_4 52
		e_3 340	e_4 60

As explained earlier, the result of our test is $\chi^2 = 2.19$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did equally badly when tested on the pure data. The classifier trained on pure only data did badly when tested on the interpolated data, and significantly worse than the one trained on interpolated data. Since the interpolated samples were all taken from the midpoint between CMR, CMSS and CMSSI, they were very different from what the pure classifier had been trained on.

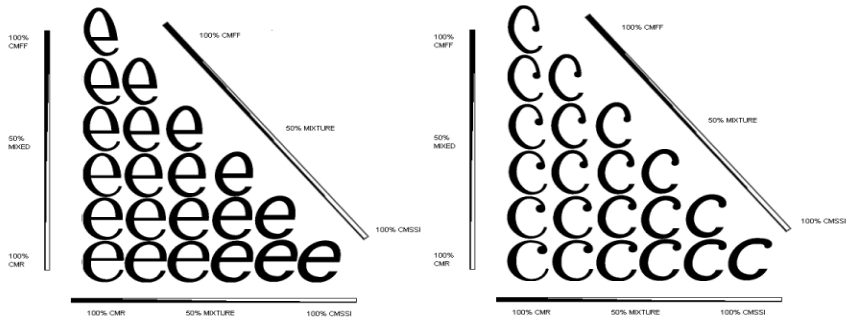
6.4.7 Experimental Description

For this experiment, the generated samples were less blurred, and had a moderate amount of noise added. The varience was less than the last experiment. Training was performed as before and the experiment took the test interpolated data entirely from the midpoint between the real CMR, CMFF and CMSSI fonts.

Percentage Typeface Mix of Training characters:

Following is a graphical representation of the types of data for this particular experiment. Recall that the end characters use real fonts while the in-between characters are mixtures of the three fonts.

Training Data.



Pur 50

** **

** ** **

** ** ** **

** ** ** ** **

50 ** ** ** ** 50

Int **

10 10

10 10 10

10 ** ** 10

10 10 10 10 10

** 10 10 10 10 **

The generating parameters and number of samples for the training data from this typeface model and this image quality model are below.

Parameters

Training data:

Pure Parameters: -e1.8,1.0 -t.3,.110 -s.3,.11

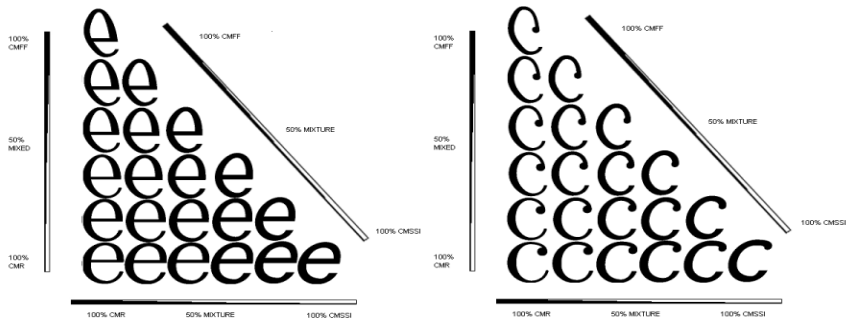
6.4. THREE WAY CMR/CMFF/CMSS EXPERIMENTS

Pure Seed: -S12
Number Samples: 1800
Interpolated Parameters: -e1.8,1.0 -t.3,.11 -s.3,.11
Interpolated Seed: numb2 + numb (numb2 = 0,2,4...10), (numb = 0,2,4...10)
Number Samples: 1800

Percentage Typeface Mix of Test characters:

Following is a graphical representation of the types of test data for this particular experiment.

Test Data.



Pur 50

** **
** ** **
** ** ** **
** ** ** ** **
50 ** ** ** ** 50

Int **
** **
** ** **
** 50 50 **
** ** ** ** **
** ** ** ** ** **

The generating parameters and number of test samples for the test data for this typeface model from this image quality model are below.

Parameters

Test data:

Real Test data: -e1.8,1.0 -t.3,.110 -s.3,.11

Real Seed: -S5

Number Samples: 480

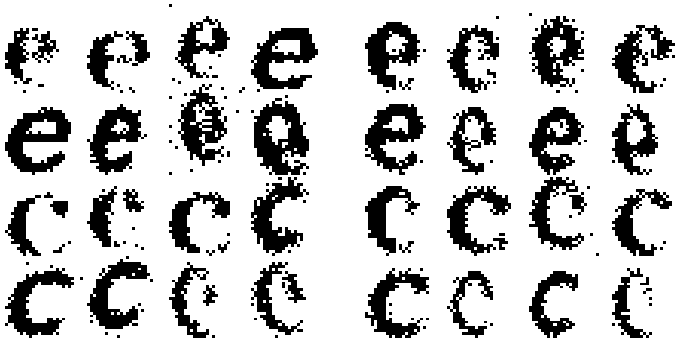
Interpolated Test data: -e1.8,1.0 -t.3,.11 -s.3,.11

Interpolated Test Seed: (6, 10)

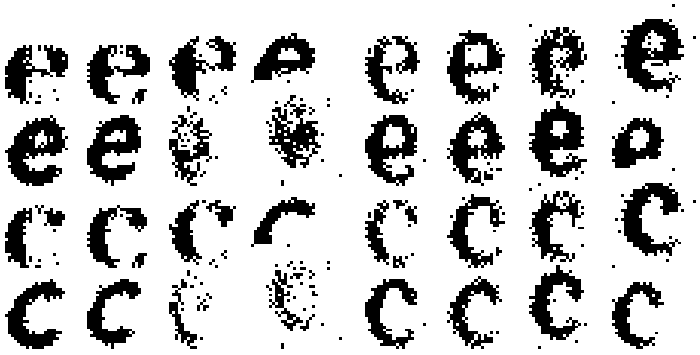
Number Samples: 480

Samples chosen to illustrate data used

Pure Training Samples Interpolated Training Samples



Pure Test Samples Interpolated Test Samples



Samples of Data with closest match

The following samples are of misclassified letters where possible. The first character is the test sample, while the second is its nearest neighbor.

pure/pure e's

pure/pure c's

interp/pure e's

interp/pure c's

pure/interp e's

pure/interp c's

interp/interp e's

interp/interp c's

Error Rates (e's / c's = tot misclassified out of 200)

		TEST ON Pure(A)		TEST ON Itrp(B)	
TRAIN ON	Pure(A)	e: 7 c: 0	7	e: 9 c: 0	9
	Itrp(B)	e: 9 c: 6	15	e: 2 c: 2	4

Hypothesis 1:

We claim that BB, the classifier trained on a mixture of pure and interpolated data will do better than AB, the one trained on the pure only data when tested on Interpolated data. The null hypothesis is therefore that AB will perform at least as well as BB, that is to say that the error rate for AB will be less than or equal to the error rate for BB. This means that the observed error rates $o_2 <= o_4$. The actual counts observed are listed below

		Right	Wrong
TRAIN ON	Pure(A)	o_1 391	o_2 9
		e_1 393	e_2 7
	Itrp(B)	o_3 396	o_4 4
		e_3 393	e_4 7

As explained earlier, the result of our test is $\chi^2 = 0.72$. Since this result is not greater than or equal to 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier BB performs better than classifer AB when tested on interpolated data.

Hypothesis 2:

We claim that BA, the classifier trained on the mixture of pure and interpolated training data will do at least as well as AA, the classifier trained on only pure data when tested on pure test data. The null hypothesis is that AA and BA will perform equally, thus the two classifiers should have the same error rates. This means that the observed error rates are equal, that is, $o_2 = o_4$. If we do not reject the null hypothesis then we have some support for our claim that they perform equally.

		Right	Wrong
TRAIN ON	Pure(A)	o_1 393	o_2 7
		e_1 389	e_2 11
	Itrp(B)	o_3 385	o_4 15
		e_3 389	e_4 11

As explained earlier, the result of our test is $\chi^2 = 2.98$. Since this result is not greater than 3.84 we can not reject the null hypothesis in favor of the hypothesis that the classifier has a different performance when trained on the training sets *A and B.*

Conclusions:

For this experiment, both classifiers did much better when tested on the interpolated data. The classifier trained on interpolated data did not do significantly better than the one trained on pure data only when tested on the interpolated data. Both classifiers also did equally well when tested on the pure samples. These were not the results we expected based on the last experiment.

6.4.8 Three Way CMR/CMFF/CMSS Test Group Results

In the following chart, we can graphically see that that classifier trained on the mixed data did significantly better than the one trained on the pure only data. This classifier appears to have recognized normal and slightly blurred samples better than the pure classifier if those samples wee taken from the full range of the data. However, it is interesting to see that the classifiers perform so similarly when the slightly blurred images are taken from midpoint of the interpolations. Once again, we see that the classifiers had similar performance when tested on the pure only data.

Figure 6.4: CMR-CMSSI-CMFF e and c Experimental Results

		Hypothesis 1				Hypothesis 2			
		Errors		Statistic		Errors		Statistic	
Image Quality	Range	AB	BB	χ^2	Rej	AA	BA	χ^2	Rej
normal	full	42	3	31.57	yes	0	0	0.00	no
slightly blurred	full	18	0	14.54	yes	0	0	0.00	no
greatly blurred, high variance	full	111	111	0.0	no	91	92	.10	no
greatly blurred, some variance	full	109	48	29.34	yes	64	72	0.54	no
greatly blurred, little variance	full	86	55	8.21	yes	54	70	2.42	no
greatly blurred, little variance	mid	57	22	17.20	yes	67	52	2.19	no
slightly blurred, little variance	mid	9	4	.72	no	7	15	2.96	no

Chapter 7

Conclusions

Our experiments have shown that classifiers trained on synthetic data interpolated in parameter space is safe, that is to say in our tests, a classifier trained on this data never worsened from tests on pure samples. Furthermore, the classifier trained on interpolated data often but not always improved accuracy in tests on interpolated samples. Accuracy was improved about one third of the time in the four sets of experiments we conducted as described below.

7.1 First Set

Our first set of experiments was performed on interpolations between the Computer Modern Roman (CMR), a serifed font, and the Computer Modern Sans Serif (CMSS), a sans-serif font, both from the Computer Modern (CM) family of fonts. These fonts were fairly similar to one other and results of testing on a set of pure test samples proved that the classifier trained on interpolated training samples performed as well as the one trained on pure samples. The classifier trained on the interpolated samples performed better than the one trained on pure samples in the two instances when test samples were taken from the full range of interpolated samples, were greatly blurred, and had little variance from the midpoint of the blurring parameter. The interpolated classifier also performed better when the test samples were all taken from the midpoint between CMR and CMSS and

Figure 7.1: CMR-CMSS e and c Results

		Hypothesis 1				Hypothesis 2			
		Errors		Statistic		Errors		Statistic	
IMAGE QUALITY	TEST SET	AB	BB	χ^2	Rej	AA	BA	χ^2	Rej
normal	full	0	1	1.00	no	0	0	0.00	no
slightly blurred	full	0	0	0.00	no	0	1	1.00	no
greatly blurred, high variance	full	62	72	.88	no	74	69	.20	no
greatly blurred, some variance	full	60	42	3.62	no	44	43	.20	no
greatly blurred, little variance	full	64	39	6.90	yes	41	45	.20	no
greatly blurred, little variance	mid	79	71	.50	no	52	40	1.76	no
slightly blurred, little variance	mid	23	10	5.20	yes	6	7	1.40	no

were only slightly blurred with little variance.

In the first two tests, the image quality was either normal or only slightly blurred. Both classifiers were able to recognize a large fraction of the characters and both performed equally well on the interpolated samples, which were taken from the full range of interpolated images. As blurring and variance increased in tests three through five, both classifiers started to perform badly. Only when the variance was decreased did the interpolated classifier pull ahead.

For the last two tests the interpolated samples were taken entirely from the midpoint between the two fonts, and thus were equally dissimilar from each font. When the test samples were greatly blurred, neither classifier was able to recognize many of the images, however when the images became less blurred, the interpolated classifier did better.

Figure 7.2: CMR-CMFF e and c Results

		Hypothesis 1				Hypothesis 2			
		Errors		Statistic		Errors		Statistic	
Image Quality	Range	AB	BB	χ^2	Rej	AA	BA	χ^2	Rej
normal	full	23	4	10.71	yes	0	0	0.00	no
slightly blurred	full	18	3	7.93	yes	0	2	0.00	no
greatly blurred, high variance	full	102	106	.80	no	91	83	.46	no
greatly blurred, some variance	full	84	83	.10	no	65	73	.54	no
greatly blurred, little variance	full	81	57	5.20	yes	58	64	.32	no
greatly blurred, little variance	mid	74	56	2.96	no	70	53	2.74	no
slightly blurred, little variance	mid	49	16	17.97	yes	8	10	.22	no

7.2 Second Set

We next thought it would be interesting to test our classifier on two fonts which were less similar. While they were once again taken from the Computer Modern family, the fonts chosen were much different from each other. We chose Computer Modern Roman (CMR) and Computer Modern Funny Font (CMFF) for this set of experiments. The same seven experiments were performed as before.

Once again, both classifiers performed equally when tested on the pure (CMR and CMFF) test sets. However, this time, the classifier trained on the interpolated data performed better when tested on the slightly blurred test sets (both full range and midpoint), as well as the greatly blurred-little variance test set (both ranges).

Figure 7.3: CMR-CMFF i and j Results

		Hypothesis 1				Hypothesis 2			
		Errors		Statistic		Errors		Statistic	
Image Quality	Range	AB	BB	χ^2	Rej	AA	BA	χ^2	Rej
normal	full	0	0	0.00	no	0	1	1.00	no
slightly blurred	full	2	3	2.00	no	1	1	.33	no
greatly blurred, high variance	full	71	65	.3	no	83	96	1.20	no
greatly blurred, some variance	full	72	64	.54	no	94	93	.10	no
greatly blurred, little variance	full	58	53	.26	no	85	87	.20	no
greatly blurred, little variance	mid	59	44	2.49	no	86	97	.85	no
slightly blurred, little variance	mid	10	10	0.0	no	29	37	1.40	no

7.3 Third Set

The next set of tests was performed on the two fonts, CMR and CMFF with the letters i and j. Once again, both classifiers performed equally when tested on the pure test sets. In every case, the interpolated classifier was at least as good as the pure classifier. However, when tested on the interpolated test images, there was also no difference between the recognition of the images. We think that there was either a great enough difference between the i's and j's that the classifiers were able to correctly identify them, or perhaps the i's and j's were similar enough in the fonts chosen that the interpolated samples did not provide enough variety.

7.4 Fourth Set

The last set of test was even more interesting and challenging. In these experiments we interpolated among three fonts, Computer Modern Roman (CMR), Computer Modern Funny Font (CMFF) and a completely different one, Computer Modern Sans Serif Italics (CMSSI). While these fonts are all from the Computer Modern family, they vary as to

Figure 7.4: CMR-CMFF-CMSSI e and c Results

		Hypothesis 1				Hypothesis 2			
		Errors		Statistic		Errors		Statistic	
Image Quality	Range	AB	BB	χ^2	Rej	AA	BA	χ^2	Rej
normal	full	42	3	31.57	yes	0	0	0.00	no
slightly blurred	full	18	0	14.54	yes	0	0	0.00	no
greatly blurred, high variance	full	111	111	0.0	no	91	92	.10	no
greatly blurred, some variance	full	109	48	29.34	yes	64	72	0.54	no
greatly blurred, little variance	full	86	55	8.21	yes	54	70	2.42	no
greatly blurred, little variance	mid	57	22	17.20	yes	67	52	2.19	no
slightly blurred, little variance	mid	9	4	.72	no	7	15	2.96	no

serifs, slant, thickness and many other characteristics.

We performed the same set of seven tests on the three-way interpolations, and once again, found that both classifiers performed equally when tested on the pure samples. That is to say, there was no loss of accuracy when the classifier trained on interpolated samples was tested on the pure test set.

Interestingly enough, the classifier trained on the interpolated samples performed better when tested on the interpolated samples in five out of the seven cases. It performed better when tested on the full range of samples in every case except the one in which the images were greatly blurred with high variance. Both classifiers did badly on this one. It also performed better when the images were greatly blurred and taken from the midpoint range only. Both classifiers, however, performed equally well when tested on the slightly blurred images taken from the midpoint.

It is noteworthy that this last result differs from that of the CMR-CMFF test on the slightly blurred midpoint test samples. In that test, the classifier trained on the pure images only

could not recognize the midpoint images as well as the one trained on the interpolated images. Why was the CMR-CMFF-CMSSI classifier able to recognize the interpolated midpoint images better than the CMR-CMFF one? Could it be that the addition of the third font, even though it is a pure font, makes the classifier better able to recognize a previously unseen font? And if that is so, would the addition of many more fonts, both pure and interpolated make it even better?

7.5 Concerns

It might be argued that it is obvious that the classifier trained on interpolated samples would do better when tested on interpolated samples, but closer examination leads one to realize that this might not be the case. While the parameters are contained within a convex hull created by the starting parameters, it is not at all certain that the features of samples created with these parameters are also contained within a convex hull of the features.

It is easy to think of an example in which this is not so. For example, let us define two parameters which are used to generate a box, length and width. Suppose we start with 5x1 and end with 2x4. Interpolating between the these two parameters, we derive 4x2 and 3x3 as new dimensions. These are clearly between the starting values. Now suppose we choose to use the feature of box area in some fashion to classify our boxes. Our beginning parameters will yield *pure* boxes of area 5 and 8 respectively, while the interpolations yield boxes of area 5 and 9. The area of the second box does not fall within the convex hull of the area of the pure boxes. One could argue that a better feature should have been chosen, but choosing features is somewhat of an art, and it is not always straightforward to pick the best from among many.

This leads to the question of whether or not our chosen metric, the Hamming Metric, is convex in the domain of typeface images of characters. Although it might well be, we have not shown that to be so. One might reasonably ask that if a very large *c* in CMR is closest to a specific very large *e* in the training set would a very large interpolated *c* (say

halfway between CMR and CMFF) be closer to a small interpolated *c* in the same interpolated font, or closer to the large *e* as well. This may have been the situation in some of our tests, particularly those performed on the *i*s and *j*s. In these cases, the classifier trained on the interpolated data performed no better than the classifier trained on the pure data

Another concern is that the use of the interpolated training data in the design of our experiments might have led to improvements merely by the addition of more data and that adding more pure samples would have had the same effect. We point out that the interpolated tests did not add more data, as the interpolated samples replaced existing pure samples. The training sets were the same sizes in each of the tests. The interpolated training samples are more varied and it is easy to imagine that by adding these, a neighbor might be introduced which is equidistant with some other neighboring sample but of a different class. In such a case, it would be a toss-up as to which sample was picked as the nearest, and the classifier might perform worse.

One potential problem with the addition of new training samples is the problem of overfitting which leads to bad generalization. Sometimes one might get good accuracy but bad generalization, or bad accuracy but good generalization. In other words, the classifier might get a tight fit around the training data, however this may lead to poor classification on the test data.

We bring up these issues in an effort to address questions which might arise regarding the design of our experiments, and to show that under some circumstances the addition of interpolated training samples might lead to poorer performance both with the interpolated test samples as well as the pure test samples.

Chapter 8

Additional Experiment

One of the existing tests might be biased towards the interpolated training data set; it was suggested that it might be better to employ pure test data only in a separate, parallel set of experiments.

8.1 Design of New Experiments

We briefly describe the design of new experiments as follows:

Compare the error rate of classifiers trained on data with interpolated data, to the error rate of a classifier trained on pure data only, running a sequence of tests over a range of the number of training samples. This may provide stronger evidence for our broad hypothesis that the addition of interpolated samples drives down the error rate of a classifier on both pure and interpolated samples, without harming the accuracy of the classifier on pure samples. If so, it will strengthen our argument that training on interpolated data does not hurt when testing on pure data, and might also show that the addition of interpolated training data can help the classifier when testing on pure data.

To implement this experiment we performed the following steps.

1. We created a training set consisting of 1000 pure samples each of e and c:

   500 CMR es, 500 CMFF es, 500 CMR cs and 500 CMFF cs

   The image quality parameters for this pure training set were:

   -e.8,1.0 -t.3,.110 -s.3,.11, Seed: -S12.

   This provided a moderate amount of blur, threshold and sensitivity with little variance.

2. We created a pure test set of 1000 of the samples of e and c consisting of

   500 CMR es, 500 CMFF es, 500 CMR cs and 500 CMFF cs

   The image quality parameters were:

   -e.8,1.0 -t.3,.110 -s.3,.11 Seed: -S5.

   The seed was different to insure that generally distinct samples were created.

3. A series of tests on pure training sets was performed as follows:

   (a) For the first test, the classifier was trained on 10 samples each of e and c:

   5 CMR es, 5 CMFF es, 5 CMR cs and 5 CMFF cs

   i. The classifier was tested on the 1000 test samples.

   ii. Error rates were recorded

   (b) For the second test, the classifier was trained on 20 samples each of e and c:

   10 CMR es, 10 CMFF es, 10 CMR cs and 10 CMFF cs).

   The training set included the original samples from test a. above.

   i. The classifier was tested on the 1000 test samples.

   ii. Error rates were recorded

   (c) This series was extended, at each step adding 10 more training samples each of e and c , for a total of 100 tests.

   i. Each classifier was tested on the 1000 test samples.

ii. Error rates were recorded for each classifier,

4. The results from 3. above were plotted: number of errors as a function of the number of samples.

Figure 8.1: Results with pure training data

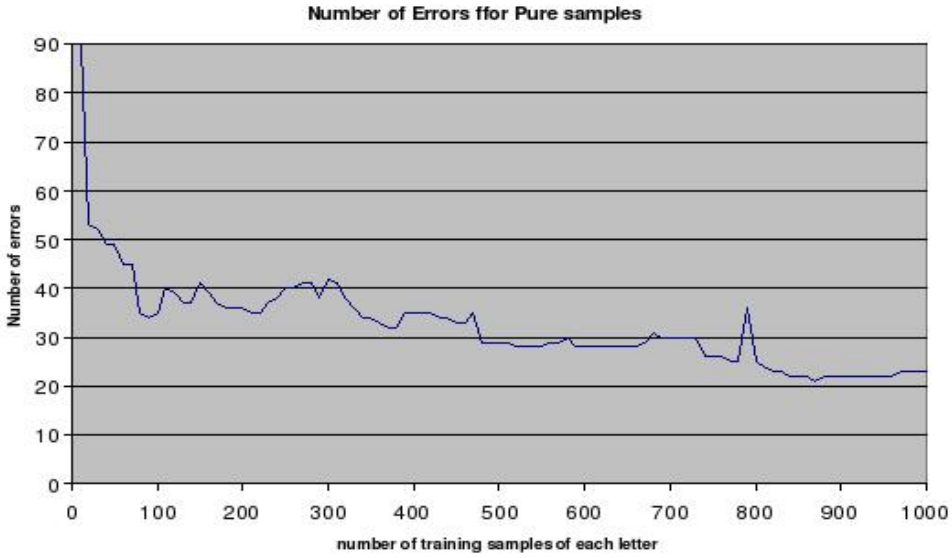


Figure 8.1 shows a graph of our results for the tests using pure test samples and only pure training data.

The graph clearly shows a large drop in the error rate between 10 and 80 samples. In fact, the error rate goes from 90 rapidly down to around 35 errors. From 80 to 800 samples the graph trends downwards to a minimum of approximately 23 errors and then flattens out. Althrough there are several peaks and valleys in the individual error rates, these can be attributed to the fact that among the training samples for that particular set, there might have been one sample which was in fact closer to samples of the other class. When we smooth out the curve, we can clearly see the downward trend.

5. Using this graph it was possible to identify our area of interest as the interval between 10 and 500 samples for further experimentation with interpolated training data. This domain was tested further using interpolated test data as follows.

6. A training set with 10 times the number of samples at the maximum of the area of interest was created. (In our test, that would be thus 5000 test samples). These training samples were interpolated samples ranging between CMR and CMFF.

   The image quality parameters for the interpolated training set were

   -e.8,1.0 -t.3,.110 -s.3,.11 Seed: variable

7. For each of our test points between 10 and 500 (the area of interest), a test as above was performed using as training data the original pure samples enriched by 10x the amount of interpolated training samples. (In our example, the first test had 10 pure samples +100 interpolated samples added to the training data, while the last test had 500 pure samples +5000 interpolated samples added to the training data.)

   The results of these tests as well as the results of the tests on the pure training data only were plotted together: number of errors as a function of the number of pure samples (from 10 to 500).

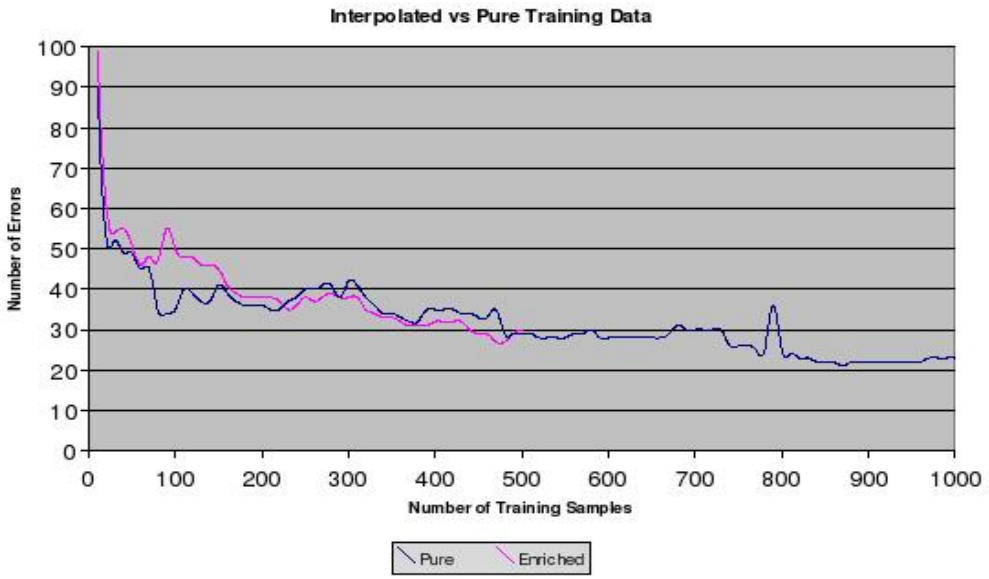
8.2 Results of New Experiment

Based on these results we were able to decide whether or not the error rate on the training data enriched with interpolated samples is the same or different than that of the pure training set, and in what way. Figure 8.2 shows our results.

The graphs clearly show similar error rates for the tests conducted with pure traing data to the tests conducted with the enriched training data. The two lines on the graph start with a similar number of errors and descend at roughly the same rate, leveling off at about the same number of training samples. In general, this supports our theory that interpolated

training data does not hurt the classifier when tested on pure samples. However, there is an interesting divergence at roughly 40 samples. In this area the line for the pure training samples descends abruptly then rises to around 40 errors while the line for the interpolated training samples rises to around 55 before descending to 40 errors. We chose this area to examine in more detail in an effort to determine if the difference was significant.

Figure 8.2: Results with pure vs. enriched training data

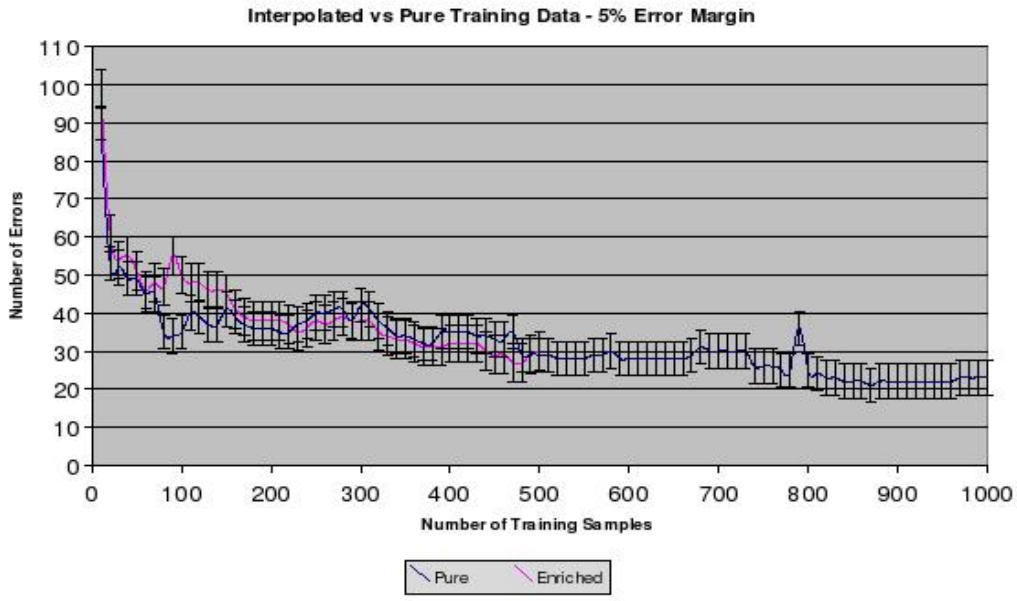


Our first step was to determine if the differences fell within a 5 percent margin of error. If they did so, we could conclude that in fact there was no difference between the test results.

Figure 8.3 shows graphical display of the test results with bars representing the margin of error for each sample point. Even while accounting for the margin of error there are still a few test points which show a difference; the question then becomes whether or not that difference is significant.

Our next step was to expand the graph to more clearly show our area of interest and to identify the test points in question. In doing so, we found that the points in question were

Figure 8.3: Error bars for pure vs. enriched training data



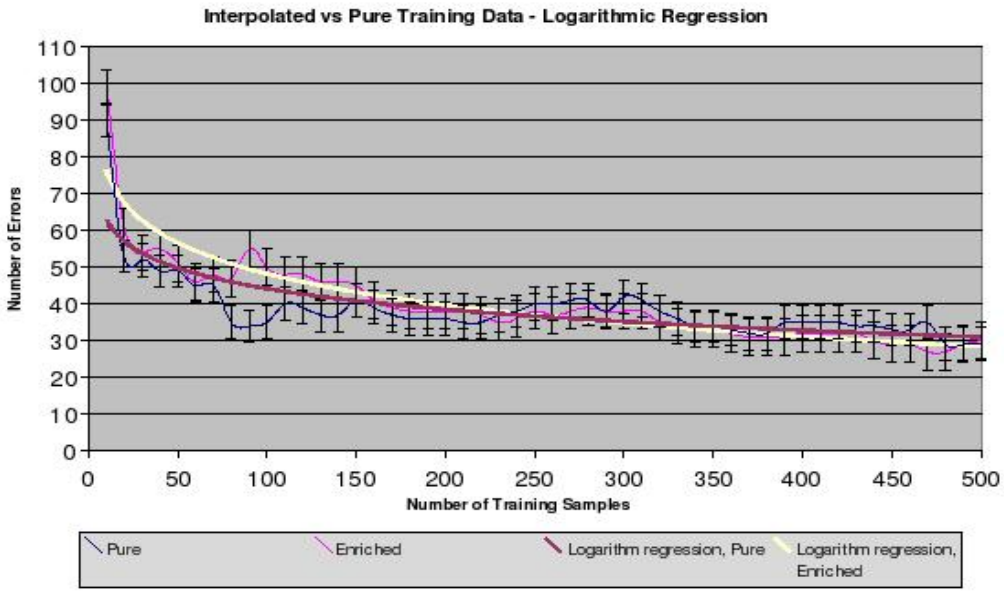
at 80, 90 and 100 samples. The pure training set at these points thus consisted of 80, 90 and 100 samples each of e and c, while the enriched training set at these particular points consisted of 80 pure + 800 interpolated, 90 pure + 900 interpolated and 100 pure + 1000 interpolated samples each of e and c.

Figure 8.4 shows our area of interest enlarged. We have also fitted logarithmic regression lines to the data which will be discussed in more detail later.

Based on the identification of the tests consisting of 80, 90 and 100 pure training samples and the corresponding enriched plus pure training samples we decided to perform χ^2 tests on the individual results at these test points.

We obtained the following results for each of the three test points. As before, our χ^2 is based on the differences between the observed and expected counts of errors for each type of classifier. The calculation of the χ^2 is straightforward and discussed earlier in the dissertation.

Figure 8.4: Expanded graph pure vs. enriched training data



- test point 80: χ^2 = 1.76
- test point 90: χ^2 = 5.10
- test point 100: χ^2 = 2.66

Since the result of the χ^2 for test point 90 is greater than 3.84 we can conclude that there is a significant difference between the accuracy of the classifier trained on pure only samples and the one trained on enriched samples for this particular test. The other two test points did not show a significant difference.

While there may be various reasons for this difference, we believe that the most likely is that the difference is due to the random variation in the generation of the test and training samples. We do not believe that a single difference in one test out of a series of 250 tests invalidates our conclusion that the use of interpolated training data harms the accuracy of a classifier trained on only pure samples when tested on pure data.

8.3 Further Tests

We further tested the area between 70 and 110 samples by conducting a series of tests using different seeds for a same sized set of test data, and the training sets of pure and enriched samples as before. This had the effect of creating different but same-sized test sets. We averaged the results of the series of tests and plotted the results on a new graph (figure 8.5) including the error bars to show a 5 percent margin of error.

Since the bars did not overlap, we continued our investigation by performing the χ^2 for each of these test points, arriving at the following:

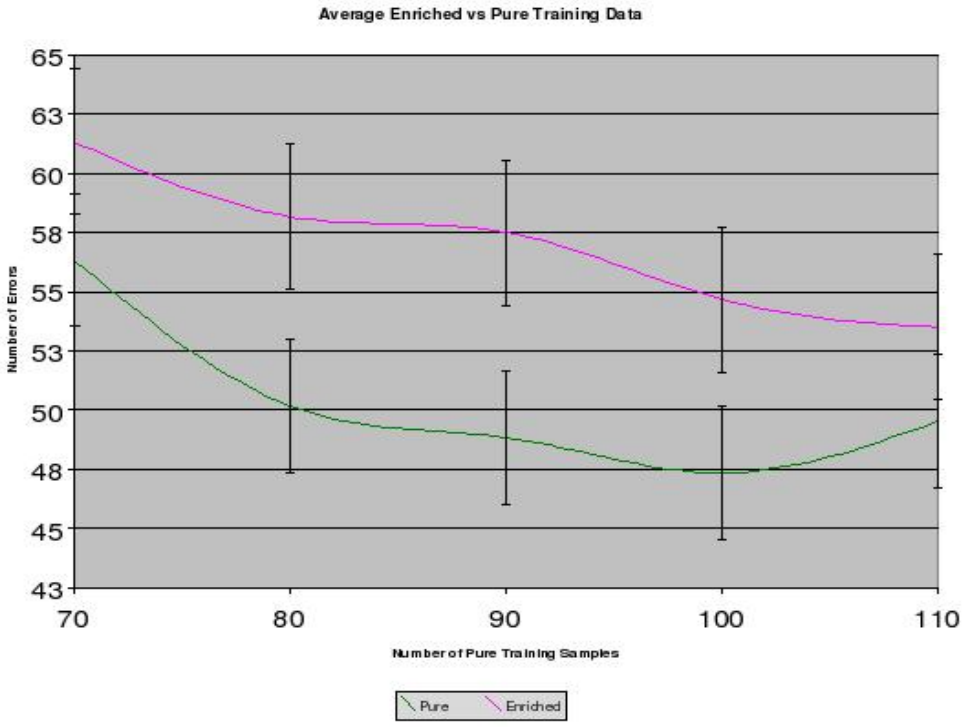
- test point 80: χ^2 = .60
- test point 90: χ^2 = .78
- test point 100: χ^2 = .64

Since none of these results are greater than 3.84, we can conclude that there is no significant difference between the performance of the enriched and the pure only classifiers at these test points.

The logarithmic regression lines were chosen to fit to the graph as they provided the best match to our data. In examining these lines one can see a smooth downward trend. With the addition of more and more training samples, and absent the addition of new information about the samples (that is to say, new features), the error rate of the classifier will approach some minimum floor beyond which the addition of more samples will not cause it to improve. Our experience and theory tells us that this minimum number of errors is predicted to be twice the Bayes error, and further impovement beyond that is not possible.

Although we had hoped that the classifier trained on the enriched training samples would perform better when tested on the pure sample set, our additional experiment did not show any evidence that this occurred. This, again, is in line with the results from our previous experiments in which the classifier trained on interpolated samples did not differ

Figure 8.5: Average of 5 tests



significantly from the one trained on the pure only samples in the classification of pure samples. (AA vs. AB)

Chapter 9

Overall Conclusions

In our research we set out to answer three questions. First, we wanted to discover if it is possible to create interpolations in parameter space. We found that it was indeed possible. By using Knuth's Metafont system and interpolating between the parameters for each font we were able to create legible and credible new fonts.

Secondly we wanted to explore if the use of these interpolations is safe, at the very least in a controlled set of experiments. And thirdly, we sought to determine if there is at least one set of circumstances in which the use of samples created with the interpolated parameters leads to better performance.

In our systematic family of tests, we found that the use of synthetic training data generated by interpolation in parameter space is safe in that it has never worsened results, and has frequently improved results. The improvement is greater when the fonts being interpolated between are more different from one another and when the images are greatly blurred with little variance. The three-way interpolation tests showed the most number of significant improvements for the interpolated training sets. Note that the results shown are for the two easily confused pairs of characters *e/c* and *i/j*.

We refer the reader to the chart in figure 9.1 for a concise graphical summary of our results.

Furthermore, an additional series of experiments we performed to test whether or not the addition of new interpolated samples harmed a classifier when tested on pure samples, showed graphically that in general, this did not hurt the classifier. This provides support for our assertion that the use of interpolated data is safe.

Our research has brought up many interesting ideas. No one in the world has yet used typeface interpolations to generate synthetic data and we are pleased to be the first and hope there is a useful application for our research results. We would offer up the following ideas as food for thought.

- Legibility is convex in parameter space, that is to say that any font interpolated between two legible fonts is still legible.
- Legibility is convex both in typographic space and in image quality space.
- Conversely, can we say that any interpolated font between two illegible fonts is illegible? We have not proven that, but it is interesting to consider.

So what exactly do we mean by legibility? One definition is that legibility is commonly thought of as the degree at which glyphs and vocabulary are understandable or readable based on appearance. [LD08] Legibility pertains to the quality of the typeface design. Sometimes it is simply a matter of type size, however often it is a matter of typeface design. In general, typefaces that are true to the basic letterforms are more legible than typefaces that have been condensed, expanded, embellished, or abstracted. Some factors influencing legibility are type size and design, for example, comparing serif vs. sans serif type, italic type vs. roman type, line length, line spacing, color and contrast. Some commonly agreed findings of legibility research include the following [TS08]:

- Text set in lower case is more legible than text set all in upper case (capitals), presumably because lower case letter structures and word shapes are more distinctive.
- Extenders (ascenders, descenders and other projecting parts) increase salience (prominence).

- Regular upright type (roman type) is found to be more legible than italics.
- Contrast, without dazzling brightness, has also been found to be important, with black on yellow/cream being most effective.
- Positive images (e.g. black on white) are easier to read than negative or reversed (e.g. white on black).

The most legible typefaces contain big features such as large, open counters, ample lowercase x-heights, and character shapes that are obvious and easy to recognize. They are restrained. They are not excessively light or bold, weight changes within character strokes are subtle, and serifs, if the face has them, do not call attention to themselves. [TA08]

Let us say that we are interpolating between two legible characters. Assume that we start with one legible character and smoothly change the size, open counters, x-heights, etc. and various other characteristics of legibility as defined above. At some interpolation, say the *ith* interpolation, we arrive at a character which is no longer legible. Now, since we are interpolating smoothly, we keep changing the characteristics in the same direction to produce more interpolations. For example, if the lower-case x-heights are smaller in interpolation *i*, they will be even smaller in interpolation *i+1*. As we continue to change the features, the character will become more and more distorted. At some point we will arrive at the other end of the interpolation. It would be impossible for this character to be legible, as in order to be so, the characteristics would have had to change back to become legible again, and they would not do so via interpolation. So, if the ending font were legible, all interpolated fonts in between would have to be legible.

While this is not a rigorous proof, it is intuitively reasonable and we suspect it could be proven for many commonly accepted individual components that make up legibility.

But is legibility also convex in image quality space? Image quality is determined by many factors of which we list just a few. [SB05]

- Sharpness - determines the amount of detail an image can convey. System sharpness is affected by the lens and sensor (pixel count and anti-aliasing filter). Lost sharpness can be restored by sharping, but oversharpening, can degrade image quality by causing ”halos” to appear near contrast boundaries.
- Noise is a random variation of image density, visible as grain in film and pixel level variations in digital images.
- Tonal Response is the relationship between light and pixel level.
- Distortion is an aberration that causes straight lines to curve near the edges of images.
- Light falloff or vignetting darkens images near the corners.
- Veiling glare is stray light in lenses and optical systems caused by reflections between lens elements and the inside barrel of the lens.

Let us pick one of the image quality factors, say noise. Our argument for the convexity of legibility in image quality space is similar to our argument for typographic space. That is to say, if our two starting images both have an acceptable amount of noise it would be impossible to interpolate between the two and obtain an in-between image with a less acceptable amount of noise. We believe this argument could be made for all of the image quality factors.

Now can we make the argument that any interpolated font between two illegible fonts is illegible? To answer this question, let us consider just one of the Metafont factors that governs the creation of a character, barheight, which determines the height of letter bar lines, such as the horizontal lines on an *E*. One can easily see that if the barheight is too small, the horizontal bars will almost disappear, leading to illegibility. Conversely, if the barheight is extremely large, the bars will merge into one big blob also leading to an illegible font. If we were to interpolate between these two extremes, we would find somewhere in between a perfect barheight which would produce a legible font.

From this example we would conclude that an interpolated font between two illegibile fonts might in fact be legible. Furthermore, given a legible font, we can always create two illegible fonts by extrapolating in either direction from the starting parameters until the ending parameters are so extreme that the fonts they produce can not be recognized.

Based on these ideas as well as our experimental results we would offer the reader this advice — engineers who wish to build classifiers to test all possible legible typefaces can use synthetically generated data. This will not harm, and occasionally will improve the classifier.

Figure 9.1: Overall Results

CHARS	FONT STYLES	IMAGE QUALITY	TEST SET	SAFE?	BETTER?
e and c	CMR-CMSS	normal	full range	yes	–
		slightly blurred	full range	yes	–
		greatly blurred, high variance	full range	yes	–
		greatly blurred, some variance	full range	yes	–
		greatly blurred, little variance	full range	yes	yes
		greatly blurred, little variance	midpoint	yes	–
		slightly blurred, little variance	midpoint	yes	yes
e and c	CMR-CMFF	normal	full range	yes	–
		slightly blurred	full range	yes	yes
		greatly blurred, high variance	full range	yes	–
		greatly blurred, some variance	full range	yes	–
		greatly blurred, little variance	full range	yes	yes
		greatly blurred, little variance	midpoint	yes	yes
		slightly blurred, little variance	midpoint	yes	yes
e and c	CMR-CMFF-CMSSI	normal	full range	yes	yes
		slightly blurred	full range	yes	yes
		greatly blurred, high variance	full range	yes	–
		greatly blurred, some variance	full range	yes	yes
		greatly blurred, little variance	full range	yes	yes
		greatly blurred, little variance	midpoint	yes	yes
		slightly blurred, little variance	midpoint	yes	–
i and j	CMRCMFF	normal	full range	yes	–
		slightly blurred	full range	yes	–
		greatly blurred, high variance	full range	yes	–
		greatly blurred, some variance	full range	yes	–
		greatly blurred, little variance	full range	yes	–
		greatly blurred, little variance	midpoint	yes	–
		slightly blurred, little variance	midpoint	yes	–

Chapter 10

Future Work

There are many issues we haven't explored. As discussed previously, there might be a bias in our set of experiments using interpolated test samples to see if the classifier trained on interpolated data performs better. A more comprehensive test might include the entire range of fonts from the Computer Modern Family (of which there are many) in the generation of our test samples. In this way one might obtain many samples of images created in fonts which have not been seen by either of our classifiers, the one trained on pure as well as the one trained on the interpolated data. An alternative to this might be to use images created with uninterpolated fonts from other than the Computer Modern family as test images in the comparison of the interpolated and pure training classifiers. There is no reason why this could not be done, and it might prove informative.

It would be good to examine the relationship among the three spaces. Does a convex region in parameter space imply a convex region in feature space and vice versa? We postulate that if the distribution is convex in feature space, then convex interpolation can't hurt and might help. If we suppose the native distribution is not convex, then convex interpolation might hurt. It would be interesting to look for a case in which the native distribution is not convex, but nevertheless convex interpolation doesnt hurt, and also to construct a case where it helps, all the while attempting to

have as little risk as possible.

Issues for future research include

- It would be helpful to answer some of the following questions for the generation and use of synthetic data in training and testing classifiers.
    1. Can we devise a method for training on synthetic data that is guaranteed never to increase confusion between any two categories?
    2. What are the conditions for the generation of synthetic data that improve classification? When is no more improvement possible and worsening likely?
    3. Can we generate exactly as many new samples as are needed to force a certain reduction in error rate?
    4. Can we consistently generate data that is *misclassified*? We might throw such data into a boosting algorithm so it attempts to accommodate the failure and thus adapt the decision boundary.
- Which methods are best suited for operating in the three spaces: parameter space, sample space, and feature space?
    1. Is it better to synthesize data in parameter space or feature space?
    2. Is there any hope of training classifiers using synthetic data to recognize Hofstadter's "Letter Spirit" typefaces?
- Here are some more theoretical questions we would also like to explore.
    1. Can we generalize convex combinations to allow non-convex combinations which are bounded and controlled, *e.g.* extrapolation? Can these also be made safe?
    2. Can we use non-convex extrapolation in a controlled way to map confusion regions in parameter space, *i.e.* regions in which synthetic data are not safe?
    3. Using the above, can we identify natural interclass boundaries by extrapolating synthetic cases until failure occurs?

4. Will any of our findings be more broadly applicable (beyond the scope of character recognition)?

Chapter 11

Lexicon of Terms

Data Space - the natural space within which real data is found. For example, images in the form of arrays of pixel values.

Feature Space - a datum is represented by a set of numerical values. Each one is a manually chosen feature, and thus each feature is a point in a multi dimensional vector space. Features are an artifact of engineering, that is to say, features are constructed or chosen to make the task of classification easier.

Parameter Space - the set of parameter values which control generation of the datum. This assumes that we know exactly how data can be generated, often pseudo-randomly.

Interpolation - in mathematics, the calculation of the values of a function between values already known.

Extrapolation - in mathematics, the calculation of the values of a function outside the range of known values.

Synthetic Data - is produced artificially, or devised, arranged, or fabricated for special situations to imitate or replace real values.

Real Data - of natural origin or occurrence.

Bibliography

[BA92] Baird, H. Document Image Defect Models. In Baird, H. S., Bunke, H. and Yamamoto, K., editors, *Structured Document Image Analysis*, Springer-Verlag, 1992.

[CH02] Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, W. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[CL02] Cano, J., Perez-Cortes J., Arlandis, J., and Llobet, R.. Training set expansion in handwritten character recognition. In *9th SSPR/4th SPR*, pages 548–556, Windsor, Ontario, 2002.

[DA91] Dasarathy, B., editors. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. ISBN 0-8186-8930-7., 1991.

[DI98] Dietterich, T. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[DU01] Duda, R., Hart, P., and Stork D. *Pattern Classification*. John Wiley and Sons Inc., 2001.

[GV04a] Guo, H., and Viktor, H. Boosting with data generation: Improving the classification of hard to learn examples. In *IEA/AIE*, volume LNAI 3029, pages 1082–1091, Heidelberg, 2004. Springer-Verlag Berlin.

[GV04b] Guo, H., and Viktor, H. Learning from imbalanced data sets with boosting and data generation: The databoost-im approach. *Sigkdd*

Explorations, 6(1):30–39, 2004.

[HA50] Hamming, R. Error Detecting and Error Correcting Codes. Bell Systems Technical Journal 26(2):147–160, 1950.

[HB97] Ho, T., and Baird, H. Large-scale simulation studies in image pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1067–1079, 1997.

[HM93] Hofstadter, D., and McGraw, G. Letter spirit: An emergent model of the perception and creation of alphabetic style. Technical Report 68, Indiana University Center for Research on Concepts and Cognition, Department of Computer Science, Feb 15, 1993 1993.

[JA00] Japkowicz, N. Learning from imbalanced data sets: a comparison of various strategies. Technical Report Tech. rep. WS-00-05, 2000.

[JZ04] Jiang, Y., and Zhou, Z. Editing trainig data for knn classifiers with neural network ensemble. In *Proceedings of the 1st International Symposium on Neural Networks (ISNN'04*, pages pp.356–361, Dalian, China, 2004.

[KN86] Knuth, D. *Computer Modern Type Faces*. Addison Wesley Publishing Company, 1986.

[LD08] Legibility is the Degree (n.d.) Retrieved Nov 17, 2008 from http://en.wikipedia.org/wiki/Legibility

[LN01] Lopresti, D. and Nagy, G. Issues in ground-truthing graphic documents. *Fourth International Workshop on Graphics Recognition Algorithms and Applications*, Selected Papers from the Fourth International Workshop on Graphics Recognition Algorithms and Applications:46–66, 2001.

[MM97] Mao, J., and Mohiuddin, K.. Improving ocr performance using character degradation models and boosting algorithm. *Pattern Recognition Letters*, 18:1415–1419, 1997.

[MO00] Mori, M., Suzuki, A., Shio, A. and Ohtsuka, S. Generating new samples from handwritten numerals based on point correspondence.

In L.R.B. Schomaker and L.G.Vuurpijl, editor, *Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 281–290, Amsterdam, 2000.

[RA89] Rabiner, L. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE,77(2)*, pages 257–286, February 1989.

[SB02] Sanchez, J., Barandela R., Marques A., Alejo, R., and Badenas, J. Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 24(7):1015–1022, 2002.

[SB05] Sheikh, H.R., and Bovik, A.C. Information Theoretic Approaches to Image Quality Assessment. In Bovik, A.C., *Handbook of Image and Video Processing*, Elsevier, 2005.

[SU04] Sun, J., Hotta, Y., and Katsuyama, Y. Low resolution character recognition by dual eigenspace and synthetic degraded patterns. In ACM, editor, *HDP '04*, pages 15–22, Washington, DC, USA, 2004. ACM.

[SV94] Smith, S., Bourgoin, M., Sims K. and Voorhees, H. Handwritten character classification using nearest neighbor in large databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):915–919, 1994.

[SV98] Simard, P., Le Cun, Y., Denker, J., and Victorri, B. Transformation invariance in pattern recognition - tangent distance and tangent propagation. In G. B. Orr Miller and K-R, editors, *Neural Networks: Tricks of the Trade*, volume Chapter 12. Springer, 1998.

[TA08] Three aspects of legibility (n.d.) Retrieved Nov 17, 2008 from http://www.fonts.com/aboutfonts/articles/typography/legibility.htm

[TS08] Text set in lower case (n.d.) Retrieved Nov 17, 2008 from http://en.wikipedia.org/wiki/Typography

[VH03] Varga, T. and Bunke, H. Effects of training set expansion in handwriting recognition using synthetic data. In *11th Conf. of the International Graphonomics Society*, pages 200–203, Scottsdale, Arizona, USA, 2003.

[VH04] Varga, T. and Bunke, H. Comparing natural and synthetic training data for off-line cursive handwriting recognition. In IEEE, editor, *9th International Workshop on Frontiers in Handwriting Recognition*, 2004.

[VG04] Viktor, H., and Guo, H. *Multiple Classifier Prediction Improvements against Imbalanced Datasets through Added Synthetic Examples*. Lecture Notes in Computer Science. 3138 edition, 2004. TY - BOOK.

[ZM97] Zhu, H., Hall, P. and May, J. Software unit test coverage and adequacy. *ACM Computing Surveys*, 29(4), 1997. Good survey of testing methods.

Vita

Jean graduated *summa cum laude* from East Stroudsburg State College in 1982 with a B.S. in Computer Science and a B.A. in French. She attained her M.S. in Computer Science from East Stroudsburg University in May, 1990. Jean has worked extensively in the Information Technology industry, and has been employed at Lehigh University in the Enterprise Systems Implementation area since 1998. In 2007, she received the Lehigh University Team Spot Bonus and the Tradition of Excellence Award. Jean has also taught at Northampton County Community College, consulted at PPL Inc., and was the Director of Data Processing for Monroe County from 1986 through 1993.

Jean has authored or presented papers at ACM VRST 2002, HCI International 2005, and SPIE/IS&T 2006 and currently has a paper accepted for SPIE/IS&T 2008. Jean has been a member of the *DICE* research group at Lehigh University under the direction of Henry S. Baird. She has participated or and presented at PABUG (Banner Users Group), Collegenet, and GMIS (Government Management Information Systems) conferences. While at Lehigh University Jean helped develop the RCEAS Pilot Engineering Summer Camp program, *Whirlwind Tour of Computer Engineering* in 2004, and has volunteered consistently with the United Way Day of Caring and Freshman MOOV. Jean was a member of the MONCARES Human Resources steering committee from 1990-1992 and has been the Administrative Board Chairperson for the Neola United Methodist Church since 2000. She is a merit badge counselor for the Boy Scouts of America in the areas of Computers, Citizenship in the Nation, and Backpacking.