Perturbation Models for Generating Synthetic Training Data in Handwriting Recognition

Tamás Varga and Horst Bunke

University of Bern
Institute of Computer Science and Applied Mathematics (IAM)
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{varga,bunke}@iam.unibe.ch

Summary. In this chapter, the use of synthetic training data for handwriting recognition is studied. After an overview of the previous works related to the field, the authors' main results regarding this research area are presented and discussed, including a perturbation model for the generation of synthetic text lines from existing cursively handwritten lines of text produced by human writers. The goal of synthetic text line generation is to improve the performance of an off-line cursive handwriting recognition system by providing it with additional training data. It can be expected that by adding synthetic training data the variability of the training set improves, which leads to a higher recognition rate. On the other hand, synthetic training data may bias a recognizer towards unnatural handwriting styles, which could lead to a deterioration of the recognition rate. The proposed perturbation model is evaluated under several experimental conditions, and it is shown that significant improvement of the recognition performance is possible even when the original training set is large and the text lines are provided by a large number of different writers.

1 Introduction

The problem of automatic recognition of scanned handwritten documents is of great significance in numerous scientific, business, industrial, and personal applications that require the reading and processing of human written texts. The ultimate goal is that computers approach, or even surpass, the text recognition performance of humans. Despite the enormous amount of research activities that already have been carried out in the past decades to study this problem, it is considered very difficult and still not satisfactorily solved [1, 2]. Today's commercial systems work in areas where strict task specific knowledge and constraints are available, such as postal address reading [3], and the processing of bank checks [4] and forms [5, 6]. On the other hand, the more challenging task of recognizing unconstrained handwriting has also many potential applications, for example, office automation, digital libraries, and personal digital

T. Varga and H. Bunke: *Perturbation Models for Generating Synthetic Training Data in Handwriting Recognition*, Studies in Computational Intelligence (SCI) **90**, 333–360 (2008)
www.springerlink.com

assisting devices. In this chapter the problem of unconstrained recognition is addressed.

Despite the existence of the numerous elaborated and mature handwriting recognition techniques [7, 8, 9, 10, 11, 12], machines' reading performance is still considerably lower than that of humans. This inspired researchers to focus not only on the development of novel recognition algorithms, but also on the improvement of other aspects of handwriting recognition systems. These efforts include multiple classifier combination [13, 14, 15], the better utilization of the available a-priori, e.g. linguistic knowledge [16, 17], as well as the collection of large, publicly available datasets of human written texts [18, 19, 20], which enables better training of the recognizers and also an objective comparison of their performances.

As an alternative, to overcome the difficulties and inherent limitations of collecting a large number of human written samples, the present chapter investigates the generation and use of synthetic training data for off-line cursive handwriting recognition. It has been shown in many works before that the size and quality of the training data has a great impact on the performance of handwriting recognition systems. A general observation is that the more texts are used for training, the better recognition performance can be achieved [21, 22, 23, 24].

In this work it is examined whether this observation holds if the training set is augmented by synthetically generated texts. The motivation is that augmenting the training set by computer generated text samples is much faster and cheaper than collecting additional human written samples. To achieve our goal, a perturbation model is presented to generate synthetic text lines from existing cursively handwritten lines of text produced by human writers. Our purpose is to add synthetic data to the natural training data, rendered by human writers, so as to enlarge the training set. The basic idea of the approach is to use continuous nonlinear functions that control a class of geometrical transformations applied on the existing handwritten texts. The functions ensure that the distortions performed are not reversed by standard preprocessing operations of handwriting recognition systems. Besides the geometrical distortions, thinning and thickening operations are also part of the model.

A closer examination reveals, however, that the use of synthetic training data does not necessarily lead to an improvement of the recognition rate, because of two adversarial effects. First, it can be expected that the variability of the training set improves, which potentially leads to a higher recognition rate. On the other hand, synthetic training data may bias a recognizer towards unnatural handwriting styles, which can lead to a deterioration of the recognition rate, particularly if natural handwriting is used for testing.

The aim in this chapter is to find configurations of our recognizer and the synthetic handwriting generation process, by which the recognition performance can be significantly improved. The parameters examined include the number of Gaussian mixture components in the recognizer used for distribution estimation, distortion strength, training set size, and the number of writers in the training set. It is shown that significant improvement of the recognition performance is possible even when the original training set is large and the text lines are provided by many different writers. But to really achieve an improvement in this case, one has also to consider the capacity of the recognition system, which needs to be appropriately adjusted when expanding the training set with synthetic text lines. Parts of this work have been published in [25, 26]. The current chapter provides a synoptic presentation and overview of the authors' previous work on synthetic text line generation for the training of handwriting recognition systems.

The chapter is organized as follows. In Section 2, an overview of the related previous works on synthetic text generation is given. Section 3 introduces our perturbation model, while in Section 4 a concise description of the off-line handwriting recognition system used for the experiments is given. Experimental results are presented in Section 5. Finally, Section 6 provides some conclusions and suggestions for future work.

2 Synthetically Generated Text

The concept of synthetic text relates to both machine printed and handwritten documents. Synthesizing text means that real-world processes that affect the final appearance of a text are simulated by a computer program. For example, in the case of machine printed documents the printing and scanning defects, while in the case of handwriting the different writing instruments or the whole writing process can be modeled and simulated by computer.

Synthetic texts can be generated in numerous ways, and they have widespread use in the field of document analysis and recognition. In the following, a brief overview is given. Approaches for both machine printed and handwritten synthetic text generation are presented, since they often have similar aims, and thus the findings and developments of one field can also affect and stimulate the other one and vice versa.

2.1 Improving and Evaluating Recognition Systems

The two main difficulties that contemporary text recognizers have to face are the degraded quality of document images as well as the great variation of the possible text styles [27, 28, 29]. The quality of document images usually degrades to various extent during printing, scanning, photocopying, and faxing. Style variation means that either different fonts might be used (machine printed text), or many individual writing styles can occur (handwritten text).

One way to alleviate the above mentioned problems is to train the recognizers using sets of text samples that are more representative to the specific recognition task under consideration. This idea is supported by two facts. First of all, every recognizer needs to be trained, i.e. it has to learn how the different characters and/or words may look like. Furthermore, in the past decade researchers in the field of image pattern recognition realized that any further improvement of recognition performance depends as much on the size and quality of the training data as on the underlying features and classification algorithms used [30]. As a rule of thumb says, the classifier that is trained on the most data wins.

A straightforward way to improve the training set is to collect more real-world text samples [18, 19, 20]. The effectiveness of this approach has been experimentally justified by numerous works in the literature, yielding higher recognition performance for increased training set sizes [21, 22, 23, 24]. Unfortunately, collecting real-world samples is a rather expensive and time consuming procedure, and truthing the collected data is error-prone [31, 32]. A possible solution to these drawbacks is to create text image databases automatically by generating synthetic data, which is cheap, fast, and far less error-prone. Furthermore, it enables the generation of much larger databases than those acquired by the conventional method. The main weakness of the synthetic approach is that the generated data may not be as representative as real-world data.

In machine printed OCR (Optical Character Recognition), especially when the possible fonts are a-priori known, the concept of representativeness of the training set can be approached from the side of document degradation. In [33, 34, 35], defects caused by the use of printing and imaging devices are explicitly modeled and applied to ideal input images (e.g. Postscript document) to generate realistic image populations. Such synthetic data can then be used to build huge and more representative training sets for document image recognition systems [36, 37, 38]. The ability of controlling the degree of degradation makes it also possible to carry out systematic design and evaluation of OCR systems [36, 39, 40, 41].

For handwriting recognition, no parameterized model of real-world image populations is available, due to the lack of mathematical models accounting for the enormous variations present in human handwriting. Nevertheless, several attempts to generate synthetic data for handwriting recognition systems are reported.

In [42], human written character tuples are used to build up synthetic text pages. Other approaches apply random perturbations on human written characters [21, 43, 44, 45, 46], or words [47, 48]. In [49], realistic off-line characters are generated from on-line patterns using different painting modes.

Generating synthetic handwriting does not necessarily require to use human written texts as a basis. In [50] and [51], characters are generated by perturbation of the structural description of character prototypes.

Those works where the application of synthetic training data yielded improved recognition performance over natural training data are mainly related to the field of isolated character recognition [21, 43, 45, 46]. The natural training set was augmented by perturbed versions of human written samples, and the larger training set enabled better training of the recognizer. However, to the knowledge of the authors, for the problem of general, cursive handwritten word and text line recognition, no similar results besides those of the authors (see e.g. [25, 26]) involving synthetically generated text images have been reported.

Finally, perturbation approaches can also be applied in the recognition phase, making the recognizer insensitive to small transformations or distortions of the image to be recognized [44, 47, 52].

2.2 Handwritten Notes and Communications

The use of handwriting has the ability to make a message or a letter look more natural and personal. One way to facilitate the input of such messages for electronic communication is to design methods that are able to generate handwriting-style texts, particularly in the style of a specific person.

Such methods have several possible applications. For example, using a word processor, editable handwritten messages could be inputted much faster directly from the keyboard. For pen-based computers, errors made by the user could be corrected automatically by substituting the erroneous part of text by its corrected version, using the same writing style.

In [53], texts showing a person's handwriting style are synthesized from a set of tuples of letters, collected previously from that person, by simply concatenating an appropriate series of static images of tuples together.

Learning-based approaches are presented in [54], [55], and [56], to generate Hangul characters, handwritten numerals, and cursive text, respectively, of a specific person's handwriting style. These methods need temporal (on-line) information to create a stochastic model of an individual style.

A method that is based on character prototypes instead of human written samples is presented in [57]. Korean characters are synthesized using templates of ideal characters, and a motor model of handwriting generation (see [58]) adapted to the characteristics of Korean script. The templates consist of strokes of predefined writing order. After the geometrical perturbation of a template, beta curvilinear velocity and pen-lifting profiles are generated for the strokes, which are overlapped in time. Finally, the character is drawn using the generated velocity and pen-lifting profiles.

One possible application of the method is to build handwriting-style fonts for word processors. On the other hand, the method can provide training data for handwriting recognizers. Although the generated characters look natural and represent various styles, they were not used for training purposes.

2.3 Reading-Based CAPTCHAs

At present, there is a clear gap between the reading abilities of humans and machines. Particularly, humans are remarkably good at reading seriously degraded (e.g. deformed, occluded, or noisy) images of text, while modern OCR systems usually fail when facing such an image [59].

This observation can be used to design so-called CAPTCHAs (Completely Automatic Public Turing test to tell Computers and Humans Apart), to distinguish humans from computers [60, 61, 62]. The main application of CAPTCHAs is to prevent computer programs from automatic registration to publicly available services offered on the Internet. For example, this way spammers can be prevented from registering automatically thousands of free e-mail accounts for their fraudulent activities.

Several reading-based CAPTCHAs were proposed in the literature. All of them synthesize a degraded text image that is used to challenge the applicant to read it. The approval for the access to the required resource is then based on the correctness of the answer the applicant types in. The challenges may contain machine printed texts [60, 59, 63, 64, 65, 66, 67], or handwriting [68]. Reading-based CAPTCHAs that are already in industrial use include [60], [66], and [67].

3 Perturbation Model

Variation in human handwriting is due to many sources, including letter shape variation, variety of writing instruments, and others. In this section, a perturbation model for the distortion of cursive handwritten text lines is presented, where these sources of variation are modeled by geometrical transformations as well as thinning and thickening operations.

3.1 Previous Work and Design Goals

In the field of handwritten character recognition, numerous methods are reported to perturb character images. Among other geometrical transformations, translation, scaling, rotation, shearing, shrinking, interpolation between character samples, and also nonlinear deformations were tried [21, 43, 45, 46]. Other types of perturbations include erosion and dilation [21], and pixel inversion noise [45].

Although they seem to be very different approaches, surprisingly almost all of the transformations mentioned in the previous paragraph have been applied successfully to generate additional training samples for character recognition systems, yielding improvements in the recognition performance.[1] Thus the character recognition experiments suggest that most of the perturbations might improve the recognition rate. Furthermore, there is no comparative study showing that one or more of these approaches are superior to the others.

With this background from character recognition research in mind, the design of our perturbation model was motivated by two important aspects: simplicity and nonlinearity. Simplicity is achieved by applying the same concept

[1] The only exception is shrinking, which deteriorated the system performance in [21]

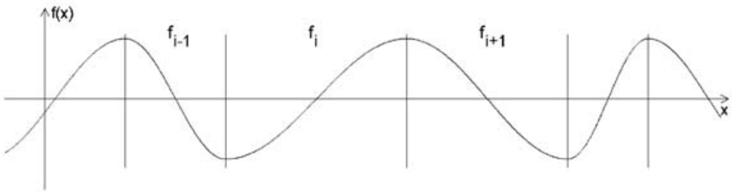


Fig. 1. Example of a CosineWave function

(underlying function, see Subsection 3.2) to each type of geometrical transformation, and considering only some basic types of distortions (shearing, scaling and shifting along one of the main axes). Nonlinearity is needed so that the distortions applied on the handwriting cannot be reversed by standard linear preprocessing operations of a state-of-the-art handwriting recognition system (see Section 4).

The perturbation model incorporates some parameters with a range of possible values, from which a random value is picked each time before distorting a text line. There is a constraint on the text lines to be distorted: they have to be skew and slant corrected, because of the nature of the applied geometrical transformations. This constraint is not severe, because skew and slant correction are very common preprocessing steps found in almost any handwriting recognition system. In the following subsections the perturbation model is described in greater detail.

3.2 Underlying Functions

Each geometrical transformation in the model is controlled by a continuous nonlinear function, which determines the strength of the considered transformation. These functions will be called *underlying functions*.

The underlying functions are synthesized from a simple function, called *CosineWave*. A CosineWave is the concatenation of n functions, $f_1, f_2, \ldots, f_n$, where $f_i : [0, l_i] \to \mathbb{R}$, $f_i(x) = (-1)^i \cdot a \cdot \cos(\frac{\pi}{l_i} \cdot x)$, $l_i > 0$. An example is shown in Fig. 1. The functions f_i (separated by vertical line segments in Fig. 1) are called *components*. The *length* of component f_i is l_i and its *amplitude* is $|a|$. The amplitude does not depend on i, i.e. it is the same for all components.

To randomly generate a CosineWave instance, three ranges of parameter values need to be defined:

- $[a_{min}, a_{max}]$ for the amplitude $|a|$,
- $[l_{min}, l_{max}]$ for the component length,
- $[x_{min}, x_{max}]$ for the interval to be covered by the concatenation of all components.

The generation of a CosineWave is based on the following steps. First the amplitude is selected by picking a value $\alpha \in [a_{min}, a_{max}]$ randomly and letting $a = \alpha$ or $a = -\alpha$ with a 50% probability each. Then l_1 is decided by randomly

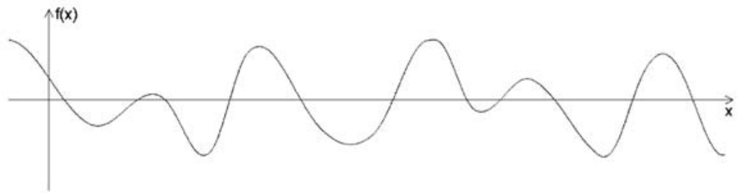


Fig. 2. Example of a sum of two CosineWave functions

picking a value from $[l_{min}, l_{max}]$. Finally the beginning of the first component (i.e. f_1) is chosen randomly from the $[x_{min} - l_1, x_{min}]$ interval. From this point on we only have to add additional components, one after the other, with randomly chosen lengths, until x_{max} is reached. For randomly picking a value from an interval, always the uniform distribution over that interval is used.

An underlying function is obtained by summing up a number, m, of such CosineWave functions. Fig. 2 depicts an example of such an underlying function with $m = 2$.

3.3 Geometrical Transformations

The underlying functions control several geometrical transformations, which are divided into two groups: the *line level* transformations applied on whole lines of text, and the *connected component level* transformations applied on the individual connected components of the considered line of text. The underlying function of each transformation is randomly generated, as described in Subsection 3.2. The parameters x_{min} and x_{max} are always defined by the actual size of the image to be distorted. In the following the geometrical transformations will be defined and illustrated by figures. Note that the figures are only for illustration purposes, and weaker instances of the distortions are actually used in the experiments described later on.

There are four classes of geometrical transformations on the line level. Their purpose is to change properties, such as slant, horizontal and vertical size, and the position of characters with respect to the baseline. The line level transformations are these:

- **Shearing:** The underlying function, denoted by $f(x)$, of this transformation defines the tangent of the shearing angle for each x coordinate. Shearing is performed with respect to the lower baseline. An example is shown in Fig. 3. In this example and the following ones, the original text line is shown at the bottom, the underlying function in the middle, and the result of the distortion on top.
- **Horizontal scaling:** Here the underlying function determines the horizontal scaling factor, $1 + f(x)$, for each x coordinate. This transformation

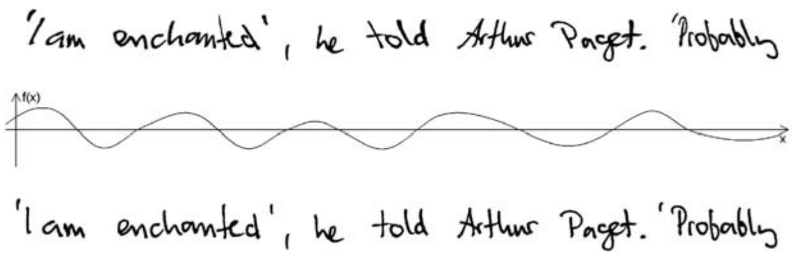


Fig. 3. Illustration of shearing. The original text line is at the bottom, the underlying function is in the middle, and result of the distortion is on top

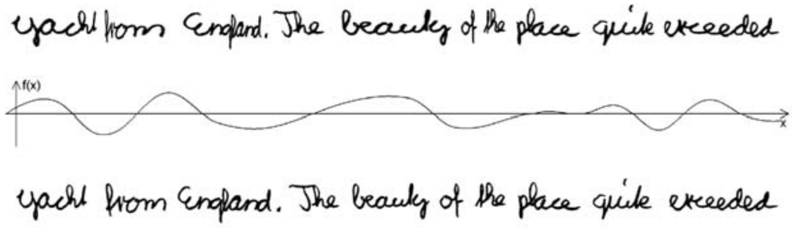


Fig. 4. Illustration of horizontal scaling

is performed through horizontal shifting of the pixel columns.[2] An example of this operation is shown in Fig. 4.

- **Vertical scaling:** The underlying function determines the vertical scaling factor, $1 + f(x)$, for each x coordinate. Scaling is performed with respect to the lower baseline. An example can be seen in Fig. 5.
- **Baseline bending:** This operation shifts the pixel columns in vertical direction, by the amount of $h \cdot f(x)$ for each x coordinate, where h is the height of the body of the text (i.e. the distance between the upper and lower baselines). An example is given in Fig. 6.[3]

The perturbation model also includes transformations, similar to the ones described above, on the level of connected components. These transformations change the structure of the writing in a local context, i.e. within each connected component. After the application of these transformations, the resulting connected components are scaled in both horizontal and vertical direction so that their bounding boxes regain their original sizes, and then they are placed in the image exactly at their original locations. For each connected component, individual underlying functions are generated. There are three classes of such transformations:

[2] The appropriate shifting value at x is given by $\int_0^x (1 + f(x))dx = x + \int_0^x f(x)dx$

[3] It can be observed that the baseline is usually not a straight line, but rather of a wavy shape

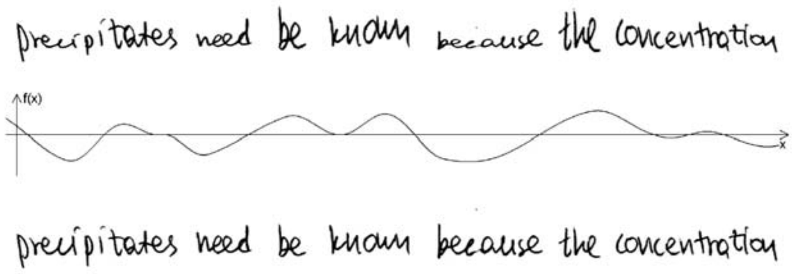


Fig. 5. Illustration of vertical scaling

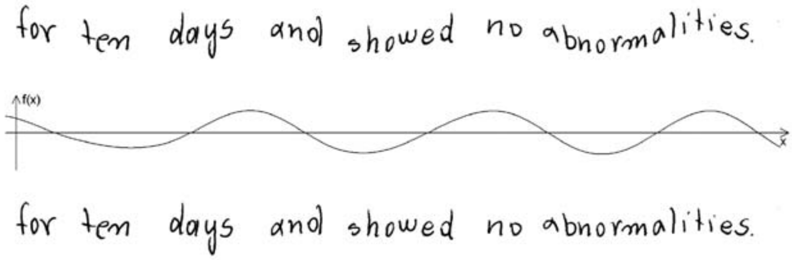


Fig. 6. Illustration of baseline bending

- **Horizontal scaling:** This transformation is identical to the line level horizontal scaling as described before, but it is applied to individual connected components rather than whole lines of text.
- **Vertical scaling 1:** This is the counterpart of horizontal scaling in the vertical direction.
- **Vertical scaling 2:** This transformation is identical to the line level vertical scaling, except that scaling is performed with respect to the horizontal middle-line of the bounding box.

The effect of all three transformations applied one after the other is shown in Fig. 7. In this figure, the lower text line is the original one, and above its distorted version is displayed. One can observe that in spite of the distortions the connected components underwent, their bounding boxes have remained the same.

3.4 Thinning and Thickening Operations

The appearance of a text line can also be changed by varying the thickness of its strokes. In the present perturbation model this is done by applying thinning or thickening steps iteratively. The method is based on a grayscale variant of the MB2 thinning algorithm [69]. (A general way to get the grayscale version

Fig. 7. Illustration of connected component level distortions. The original text line is below, and the result of the distortions is above

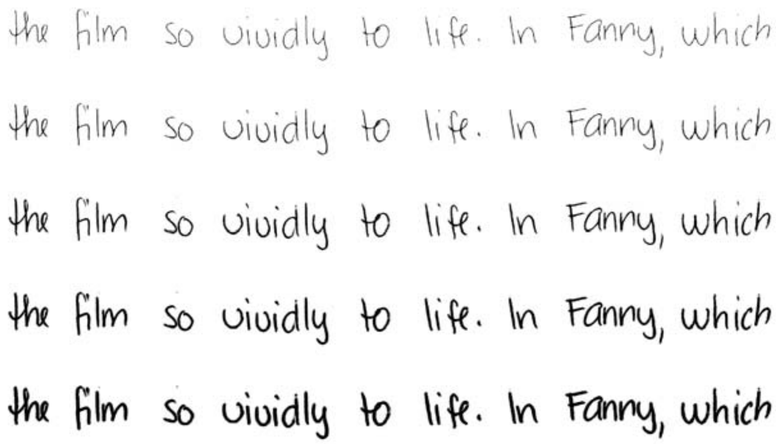


Fig. 8. Illustration of thinning (above) and thickening (below) operations. The original text line is in the middle

of a specific type of thinning algorithm operating on binary images can be found in [70]). Thinning and thickening could also be performed using the morphological erosion and dilation operators, respectively, but this would not be safe when applied iteratively, because part of the original writing might be lost after too many steps of erosion. An illustration is given in Fig. 8, where the original text line is located in the middle, and above (below) it the results of two successive thinning (thickening) steps can be seen. The choice whether thinning or thickening is applied, as well as the number of steps (including zero) is randomly made.

3.5 Distorted Text Line Generation

Now that the main constituents of the perturbation model have been introduced, a simple scheme for the distortion of whole text lines can be designed. The steps of the perturbation method for distorting a given skew and slant corrected text line are the following:

1. Apply each of the line level transformations to the text line, one after the other, in the order given in Subsection 3.3.

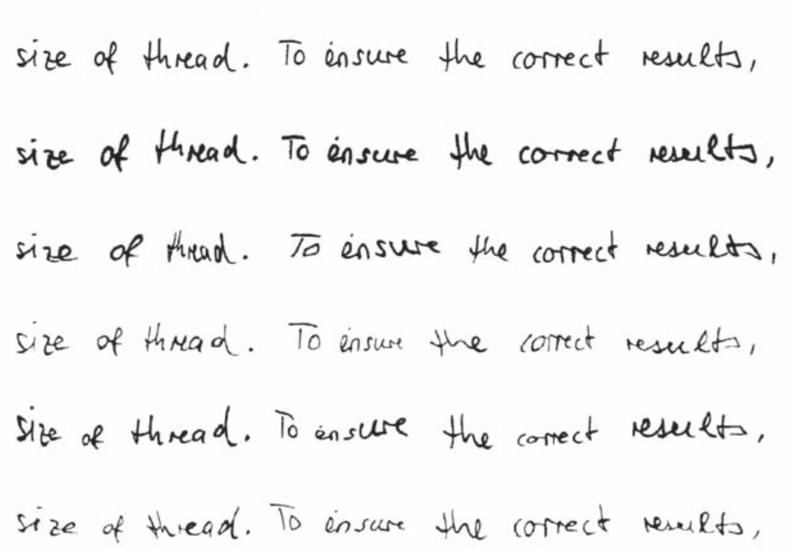


Fig. 9. Demonstration of the perturbation method. The original human written text line is on top, and below it five distorted versions can be seen

2. For each individual connected component, apply the connected component level transformations, and make sure that the bounding boxes remain the same with respect to both size and location.
3. Apply thinning or thickening operations.

Of course, these steps are not required to be always rigorously followed. In particular, one can omit one or several of the transformations. The method is demonstrated in Fig. 9. The original human written text line is on top, and below there are five synthetically generated versions of that line. It can be seen that all of the characters have somewhat changed in each generated line. Note that due to the random nature of the perturbation method, virtually all generated text lines are different. Other examples are given in Section 5.

4 Handwriting Recognition System

The application considered in this chapter is the off-line recognition of cursively handwritten text lines. The recognizer used is the Hidden Markov Model (HMM) based cursive handwritten text line recognizer described in [12]. The recognizer takes, as a basic input unit, a complete line of text, which is first normalized with respect to skew, slant, baseline location and writing width.[4]

[4] Text line normalization is also applied in the training phase. Since the text lines to be distorted have to be skew and slant corrected, synthetic training text line

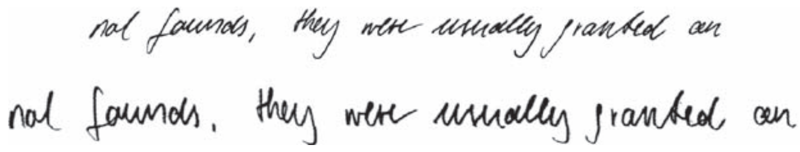


Fig. 10. Example of an input text line, before (above) and after (below) normalization

An example is shown in Fig. 10. Normalization with respect to baseline location means that the body of the text line (the part which is located between the upper and lower baselines), the ascender part (above the upper baseline), and the descender part (below the lower baseline) will be vertically scaled to a predefined height. Writing width normalization is performed by a horizontal scaling operation, and its purpose is to scale the characters so that they have a predefined average width value.

For feature extraction, a sliding window of one pixel width is moved from left to right over the input text line, and nine geometrical features are extracted at each window position. Thus an input text line is converted into a sequence of feature vectors in a 9-dimensional feature space. The nine features used in the system are the average gray value of the window, the center of gravity, the second order moment of the window, the position and the gradient of the upper and lower contours, the number of black-white transitions in vertical direction, and the average gray value between the upper and lower contour [12].

For each character, an HMM is built. In all HMMs the linear topology is used, i.e. there are only two transitions per state, one to itself and one to the next state. In the emitting states, the observation probability distributions are estimated by mixtures of Gaussian components. In other words, continuous HMMs are used. The character models are concatenated to represent words and sequences of words. For training, the Baum-Welch algorithm [71] is applied. In the recognition phase, the Viterbi algorithm [71] with bigram language modeling [17] is used to find the most probable word sequence. As a consequence, the difficult task of explicitly segmenting a line of text into isolated words is avoided, and the segmentation is obtained as a byproduct of the Viterbi decoding applied in the recognition phase. The output of the recognizer is a sequence of words. In the experiments described in the following, the recognition rate will always be measured on the word level.

---

generation takes place right after the skew and the slant of the text line have been normalized

5 Experimental Evaluation

The purpose of the experiments is to investigate whether the performance of the off-line handwritten text recognizer described in Section 4 can be improved by adding synthetically generated text lines to the training set. Two configurations with respect to training set size and number of writers are examined: small training set with only a few writers, and large training set with many writers.

For the experiments, subsets of the IAM-Database [20] were used. This database includes over 1,500 scanned forms of handwritten text from more than 600 different writers. In the database, the individual text lines of the scanned forms are extracted already, allowing us to perform off-line handwritten text line recognition experiments directly without any further segmentation steps.[5]

All the experiments presented in this section are writer-independent, i.e. the population of writers who contributed to the training set is disjoint from those who produced the test set. This makes the task of the recognizer very hard, because the writing styles found in the training set can be totally different from those in the test set, especially if the training set was provided by only a few writers. However, when a given training set is less representative of the test set, greater benefit can be expected from the additional synthetic training data.

If not mentioned otherwise, all the three steps described in Subsection 3.5 are applied to distort a natural text line. Underlying functions are obtained by summing up two randomly generated *CosineWave* functions (two is the minimum number to achieve peaks with different amplitudes, see Figs. 1 and 2). Concerning thinning and thickening operations, there are only three possible events allowed: one step of thinning, one step of thickening, or zero steps (i.e. nothing happens), with zero steps having the maximal probability of the three alternatives, while the two other events are equally probable.

5.1 Small Training Set with a Small Number of Writers

The experiments described in this subsection are conducted in order to test the potential of the proposed method in relatively simple scenarios, i.e. the case of a small training set and only of few writers. For the experiments, 541 text lines from 6 different writers, were considered.[6] The underlying lexicon consisted of 412 different words. The six writers who produced the data used in the experiments will be denoted by a, b, c, d, e and f in the following. Subsets of writers will be represented by sequences of these letters. For example, abc stands for writers a, b, and c.

Three groups of experiments were conducted, in which the text lines of the training sets were distorted by applying three different subsets of the

[5] See also: http://www.iam.unibe.ch/~fki/iamDB

[6] Each writer produced approximately 90 text lines

Table 1. Results of the experiments described in Subsection 5.1 (in %)

	original	all dist.	line level	cc. level
a	33.14	48.98	47.06	38.69
b	38.68	43.07	40.41	42.61
c	39.16	49.31	46.80	44.41
d	30.56	53.14	48.62	43.02
e	54.40	59.61	58.88	54.24
f	18.83	31.98	26.90	27.76
ab	60.69	73.46	75.79	54.92
cd	56.84	61.30	62.44	59.66
ef	63.84	68.46	67.54	67.51
abc	75.19	74.11	75.78	74.83
def	65.35	68.87	67.04	68.74

distortions described in Section 3. The three subsets were the set of *all distortions*, the set of geometrical transformations on the *line level*, and the set of *connected component level* geometrical transformations. In each case, five distorted text lines per given training text line were generated and added to the training set. So the extended training set was six times larger than the original one.

Fig. 11 shows examples of natural and synthetically generated pairs of text lines used in the experiments where all the distortions were applied. For each pair of text lines the natural one is shown below, while the synthetic one is above it. The first pair belongs to writer *a*, the second to writer *b*, and so on.

The recognition results of the three experiments are shown in Table 1, where the rows correspond to the different training modalities. The test set is always the complement of the training set, and consists of natural text only. For example, the test set corresponding to the first row consists of all natural text lines written by writers *bcdef*, while the training set is given by all natural text lines produced by writer *a* plus five distorted instances of each natural text line. In the first column, the results achieved by the original system that uses only natural training data are given for the purpose of reference. The other columns contain the results of the three groups of experiments using expanded training sets, i.e. the results for all, line level, and connected component level distortions, respectively. In those three columns each number corresponds to the median recognition rate of three independent experimental runs. In each run a different recognition rate is usually obtained because of the random nature of the distortion procedure.

In Table 1 it can be observed that adding synthetic training data leads to an improvement of the recognition rate in 29 out of 33 cases. Some of the improvements are quite substantial, for example, the improvement from 33.14% to 48.98% in row *a*.

Augmenting the training set of a handwriting recognition system by synthetic data as proposed in this chapter may have two adversarial effects on

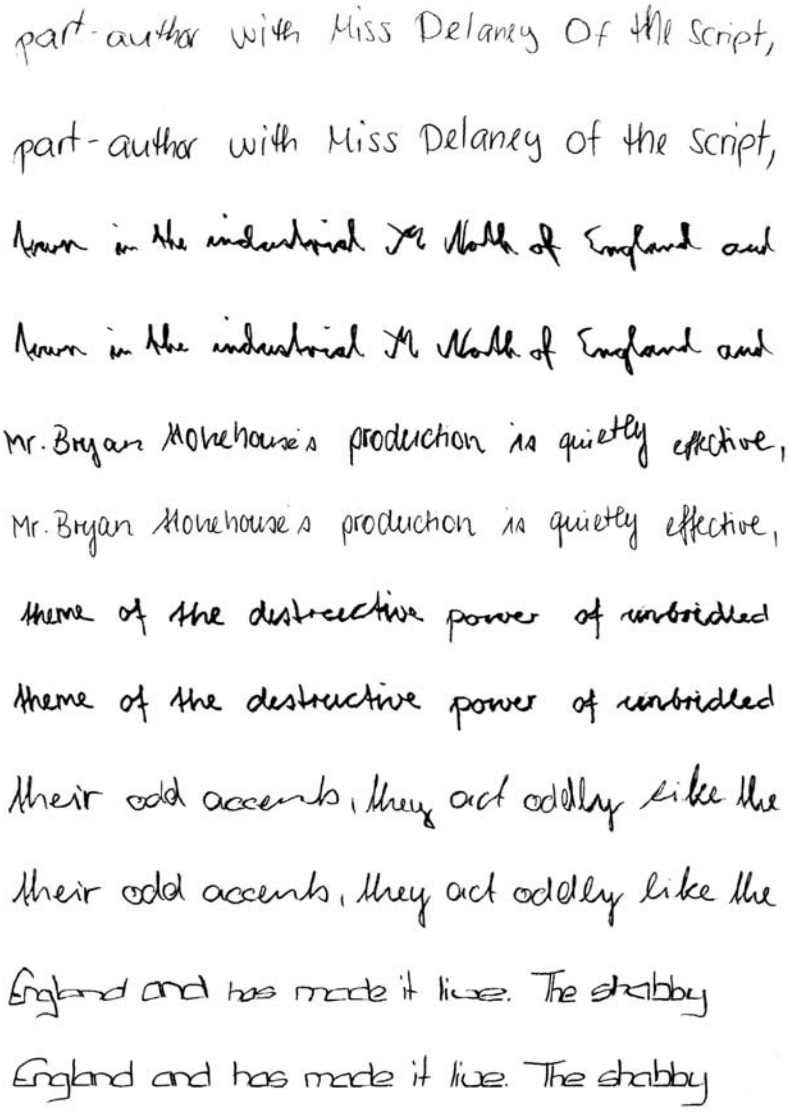


Fig. 11. Natural (below) and synthetic (above) text lines for writers *a-f*

the recognition rate. First, adding synthetic data increases the variability of the training set, which may be beneficial when the original training set has a low variability, i.e. when it was produced by only one or a few writers. On the other hand, the distortions may produce unnatural looking words and characters, which may bias the recognizer in an undesired way, because the test set includes only natural handwriting.

The greatest increase in recognition performance can be observed in Table 1 for those cases when there is only one writer in the training set. Then the variability of the training set is low and the addition of synthetic data leads to a better modeling of the test set. In this case, the application of all distortions outperforms the use of only line level or connected component level distortions. Where multiple writers are used for training, the variability of the training set is larger and the increase in recognition performance becomes smaller when synthetic training data is added. Also, in this case using all distortions does not always result in higher recognition rate than applying just line level or connected component level distortions.

Since in the majority of the experimental runs, an improvement of the recognition rate was observed, it can be concluded that the use of synthetic training data can potentially lead to improved handwriting recognition systems, in case of only a few writers in the training set.

In all experiments described in this subsection, single Gaussians were used in the HMMs' states to estimate observation probability distributions (see also Section 4). As we will see in the following, the number of Gaussians should be increased if the training set contains handwriting samples from many writers.

5.2 Large Training Set with Many Writers

In the following, the case where there are many writers and a large training set is considered. For the experiments, a subset of the IAM-Database different from that used in the previous subsection was considered, consisting of 1,993 text lines produced by 400 different writers, and the underlying lexicon contained 6,012 words. This set of text lines was randomly divided into *training*, *validation* and *test set*, such that their sets of writers were pairwise disjoint. The training and validation set contained 1,433 lines from 288 writers, and 160 text lines from 32 writers, respectively. The test set contained 400 text lines from 80 writers.

First, the training and the validation set were used to find the optimal parameters for the system that uses natural training data only, and for the system that uses a mixture of natural and synthetic training data. In the following, these two optimized systems will be referred to as *Original System* and *Expanded System*, respectively.

The optimization was performed in terms of *capacity* and *distortion strength*. The capacity of the recognition system is defined as the number of free parameters to be estimated from the training set. It determines how much information the recognizer can store to express its knowledge about the handwriting represented by the training set. A capacity too high may cause overfitting on the training data. On the other hand, a capacity too low may lead to a poor handwriting model. Since the synthetically expanded training set contains increased variability (both natural and unnatural), its optimal capacity is expected to be higher than the recognizer's optimal capacity for the original training set. That is, if the capacity of the system is not increased after the expansion of the training set, there is the danger that the capacity may be too low, such that the system is biased towards the unnatural variability introduced by the additional synthetic text lines, to an extent which may cause the recognition performance to drop. In the experiments, the capacity was varied through changing the number of Gaussian mixture components used for estimating the feature value distributions in the states of the Hidden Markov Models (see Section 4). The number of Gaussian mixtures, Ga, is the same in all HMMs. If this parameter, Ga, is increased, then it enables the system to model the distributions of the features extracted from the handwriting more accurately. Thus the capacity of the system is increased.

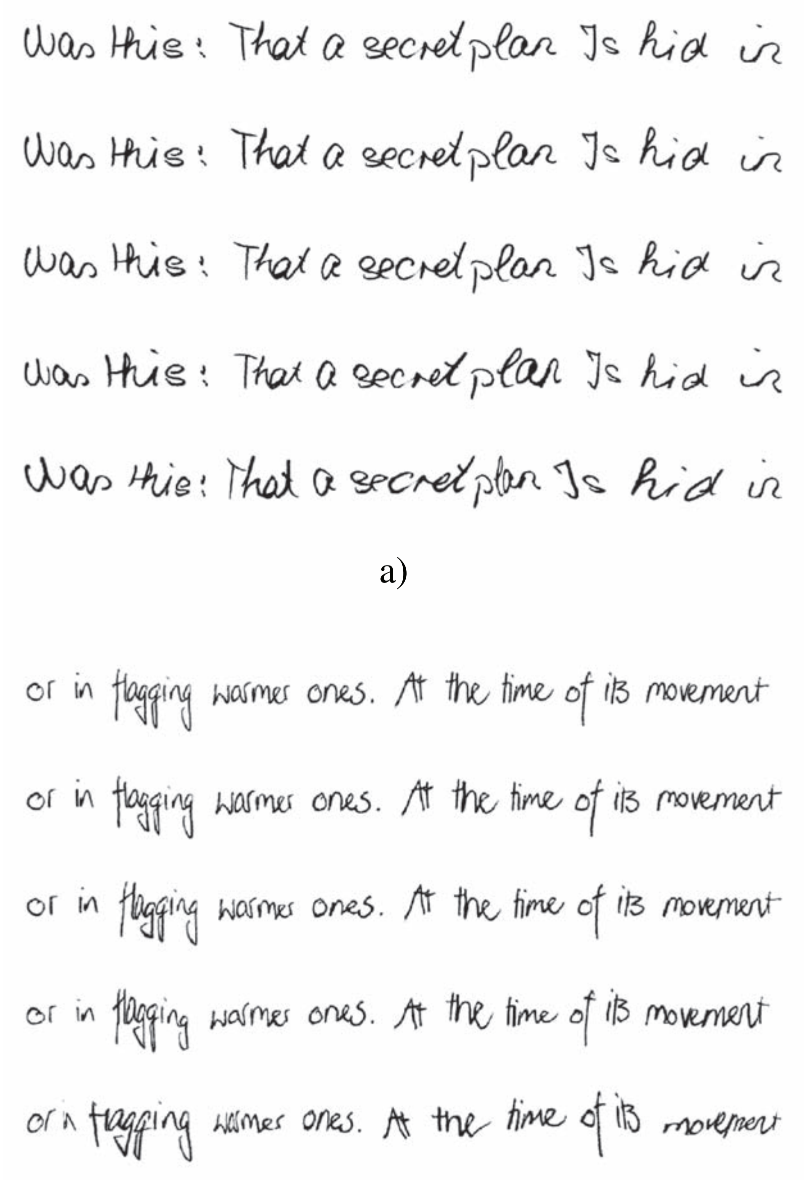
The second parameter to optimize was the distortion strength, which can be controlled by changing the interval of the possible amplitude values for the underlying functions described in Section 3. Four levels of strength were defined based on a subjective assessment: *very weak*, *weak*, *middle* and *strong*. Note that these terms indicate only the relative order of the four levels, rather than absolute categories.[7] In Fig. 12, two examples are shown, where the text lines on top were distorted using all four different distortion strengths. For the distorted text line generation, all of the distortions were applied, in the way described in Subsection 3.5. A trade-off between quality and variability of the generated text lines can be observed, which is governed by the distortion strength. That is, stronger distortions usually introduce more variability, but on the other hand, the generated text lines tend to look less natural. Thus tuning the distortion strength is expected to be beneficial.

Detailed results of the optimization stage are reported in Table 2. In the HMM training procedure, the training set, consisting of natural and synthetic training data, was used, while the recognition rates were measured on the validation set, which consisted of natural text lines only. Column *original* corresponds to the system using exclusively natural training data. According to the best result, the system with $Ga = 15$ is chosen as the *Original System*, which achieved a recognition rate of 70.48%. The other four columns, namely *very weak*, *weak*, *middle* and *strong*, show the recognition rates of the system using a mixture of natural and synthetic training data. For each text line in the training set, always five distorted text lines were generated, thus the expanded training set was always six times larger than the original one. Those results which correspond to statistically significant improvements with respect to the *Original System* (with a significance level higher than 90%), are highlighted using boldface.[8]

It can be seen that increasing the capacity is beneficial for expanded training sets. Rows $Ga = 6$ and $Ga = 12$ show the effects of low capacity after

[7] The strength was increased by *jointly* increasing the amplitude parameters for all the transformations, sampling in equal steps in terms of the parameter values of the perturbation model. For thinning/thickening, the probability of zero steps was decreased

[8] The significance level of an improvement was calculated from the writer level recognition rates, by applying a statistical z-test for matched samples



a)

b)

Fig. 12. Illustration of levels of distortion strength used in the experiments of Subsection 5.2. From top to bottom, for both a) and b) parts: original, very weak, weak, middle and strong

Table 2. Results of the optimization stage of the experiments of Subsection 5.2 (in %). Statistically significant improvements are highlighted using boldface

	original	very weak	weak	middle	strong
Ga=6	67.04	65.45	66.12	65.52	62.81
Ga=12	69.95	69.69	71.41	69.76	70.09
Ga=15	70.48	70.88	**72.27**	**71.54**	70.48
Ga=18	70.15	**72.20**	**72.47**	**72.40**	71.01
Ga=21	69.62	71.61	**72.40**	**72.01**	71.54
Ga=24	70.48	71.34	**73.00**	**73.33**	71.21
Ga=27	70.22	71.48	**72.87**	**73.86**	**71.67**
Ga=30	69.49	**71.67**	**72.14**	**73.20**	**71.74**

training set expansion with synthetic data, resulting in lower recognition rates in the majority of the cases. With an increasing strength of the distortions, the optimal capacities become higher: from column *original* to column *strong* the optimal Ga's were 15, 18, 24, 27 and 30, respectively. This can be explained by the increasing variability of the training set. (Note that for strength *strong*, the optimal capacity is possibly above $Ga = 30$.) The most significant improvements came at strengths *weak* and *middle*. All significant improvements in these columns have a significance level greater than 95%. The most significant area is at strength *middle*, from $Ga = 24$ to $Ga = 30$. Here the significance level is greater than 99%. Thus the *Expanded System* was chosen among these, namely the one with $Ga = 27$, where the recognition rate was 73.86%.

After the optimization stage, the *Original System* was trained on the union of the training and validation set, and the *Expanded System* on the union of the expanded training and expanded validation set. For each natural text line in the validation set, five synthetic text lines were generated at strength *middle* to get the expanded validation set. Then, using the test set for testing on previously unseen examples, the recognition results of the *Original System* and the *Expanded System* were 76.85% and 79.54%, respectively, as shown in Table 3. This shows that using synthetic text lines, the recognition performance could be improved by more than 2.5%. The significance level of this improvement is greater than 99%. (The recognition rates on the test set differ a lot from those measured on the validation set. This can be explained by the relatively small size of the validation set. The magnitude of the validation set is limited by the amount of text lines in the training set, so that the training set has approximately the same optimal capacity as its union with the validation set. This way the negative effects of too low capacity can be avoided at the testing phase. But the choice of the training set size is also constrained by the computational complexity of the training process, since the training of HMMs using a large number of Gaussian mixtures is a rather time consuming procedure.)

Table 3. Results on the test set of the experiments of Subsection 5.2

	Ga	strength	recognition rate
Original System	15	–	76.85%
Expanded System	27	middle	79.54%

We also note that the synthetic training set expansion methodology presented above consists of such optimizations that must always be done, independently of the underlying datasets:

- The most appropriate distortion strength between zero and extremely strong can only be found empirically, because it may depend on the details of the recognizer under consideration, as well as on the concrete dataset.
- Finding the optimal number of Gaussians (or more generally, the optimal capacity) is a must in a multi-Gaussian system, because it is dependent on the characteristics of the training set. The same optimization is needed for the synthetically expanded training set, in order to have a fair comparison with the original system.[9]

Thus, the experiments show that expansion of the available set of text lines by synthetically generated instances makes it possible to significantly improve the recognition performance of a handwritten text line recognizer, even when the original training set is large and contains handwriting from many writers.

5.3 Capacity and Saturation

The main goal of synthetic training set expansion was to improve the recognition performance, by adding synthetic text lines to the original, i.e. human written, training set. With respect to this goal, an important observation of the experiments was that the number of Gaussians needed to be appropriately increased so that the synthetic training set expansion can improve the recognition rate.

To further examine this phenomenon, an experiment was conducted using gradually increasing training sets of an increasing number of writers, while keeping the test set as well as the number of Gaussian components (i.e. the capacity) fixed. The natural training and validation set defined in Subsection 5.2 was used for training and testing, respectively. The numbers of Gaussians considered were 1 and 6. The two corresponding curves of recognition rates are shown in Fig. 13, where different proportions of the training set were used for training, while the test set was always the same. The percentages on the horizontal axis are to be understood with respect to the union of the training set and the validation set (the union consists of $1433+160 = 1,593$ text lines).

Based on these curves, two statements can be made:

[9] It was also demonstrated in this subsection why the optimization of the capacity should not be overlooked, see Table 2

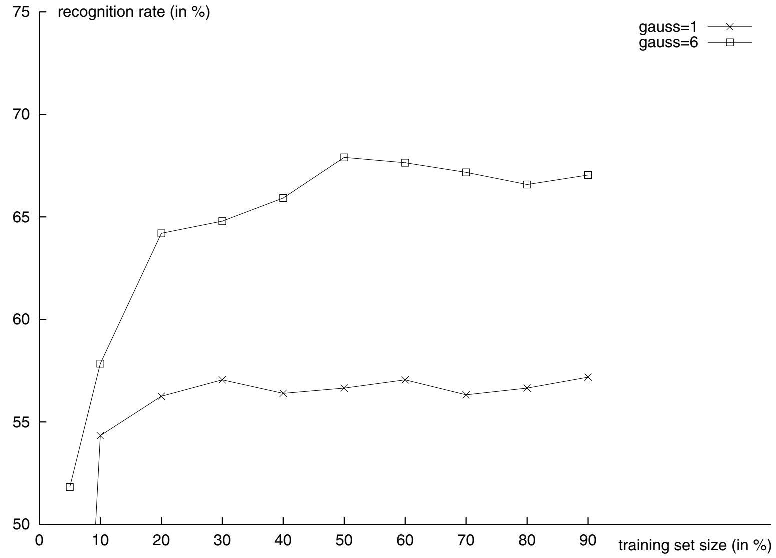


Fig. 13. Recognition rates on the test set using increasing training set sizes and fixed capacity of the recognizer

- For 1 Gaussian, we cannot expect further improvements above approximately 20% of $1,593 \approx 320$ training text lines.
- For 6 Gaussians, we cannot expect further improvements above approximately 50% of $1,593 \approx 800$ training text lines.

This leads to the intuitive notion of *saturation*, which means that given a fixed capacity of the handwriting recognition system, from a certain amount of natural training data no further improvements in the recognition rate can be expected. In other words, it cannot be predicted whether increasing the training set size yields (slightly) improved or deteriorated recognition performance. Furthermore, in Fig. 13 it also can be seen that in case of a higher capacity of 6 Gaussians, the recognizer needs more natural training data to get saturated.

Apparently, if the amount of natural training data already causes the system to be saturated, we cannot expect any positive change in the recognition rate through the expansion with synthetic data either, since even additional natural data does not help.[10] To the contrary, the negative effect of unnaturality inherent in the synthetic data can become dominant, causing the recognition rate to drop.

As an example for 6 Gaussians, in Table 2 the recognition rate dropped because the system was already saturated (note that the same data was used here

[10] Assuming that natural data is more appropriate than synthetic data for the estimation of details of natural handwriting

to illustrate saturation). In other words, the too low capacity of the system after synthetic training set expansion manifested itself through saturation.

To overcome the problem of saturation, in Subsection 5.2 the capacity of the recognizer had to be increased, in order to make room for further improvement when synthetic training set expansion is applied.

6 Conclusions and Future Work

In this chapter, the generation and use of synthetic training data in handwriting recognition was discussed. First, an overview of the related works of the field was given, including both machine printed and handwritten synthetic text generation.

The most important results of the authors' research in the field of synthetic handwriting generation for training purposes were also presented. A method for training set expansion by generating randomly perturbed versions of natural text lines rendered by human writers was presented and evaluated under several experimental conditions in writer-independent experiments. It was demonstrated that using such expanded training sets, improvements in the recognition rate can be achieved rather easily when the original training set is small and contains handwriting from only a limited number of writers. In the second experiment, it was shown that significant improvement in the recognition rate is possible to achieve even in the case of a large training set provided by many writers. In this case, the applied distortion strength needs to be adjusted, and the capacity of the recognizer (i.e. the number of Gaussians used for distribution estimations) plays an important role. The capacity has to be optimized after training set expansion, because the optimal capacity of the recognition system trained on the expanded training set is expected to be higher than the optimal capacity of the system trained on the original training set. If the capacity is not properly adjusted when using the synthetically expanded training set, there is the danger that the capacity may become too low, such that the system is biased towards unnatural handwriting styles in an undesired way, causing the recognition performance to drop.

Finally, based on the empirical observations of the experiments, the intuitive concept of saturation was introduced. The most important point is that the saturation has to be taken into account, because neither synthetic nor natural training set expansion can improve the recognition rate when the recognition system is already saturated by the available amount of natural training data. To cope with this problem, in the experiments the capacity of the recognizer was increased to open up room for further improvement.

As for possible future work, we plan to use not only one but several distortion strengths when expanding the training set. This may produce smoother training data than, for example, having only natural and strongly distorted text lines, but nothing between these two levels. Another idea is not to add all the generated texts to the training set, but perform a kind of pre-selection of the most appropriate ones, by using an rejection mechanism. Style dependent distortions as well as distortion strengths may also facilitate the creation of expanded training sets of better quality.

Since the problem of synthetic training data was addressed from a rather general point of view in the experiments, many questions mostly related to the enhancement of the baseline perturbation method are still open, e.g. considering other types of distortions as well as underlying functions, or examining the suitability of the individual distortions.

Our current work makes use of HMM for handwritten text line recognition. However, similar effects can be expected when dealing with other types of recognizers, for example, nearest neighbor classifier [21, 46] and neural networks [43, 45].

References

1. Bunke, H.: Recognition of Cursive Roman Handwriting – Past, Present and Future. In: Proc. 7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland (2003) 448–459
2. Plamondon, R., Srihari, S.: On-line and Off-line Handwriting Recognition: A Comprehensive Survey. IEEE Trans. on Pattern Analysis and Machine Intelligence **22**(1) (2000) 63–84
3. Srihari, S.: Handwritten Address Interpretation: a Task of Many Pattern Recognition Problems. Int. Journal of Pattern Recognition and Artificial Intelligence **14**(5) (2000) 663–674
4. Impedovo, S., Wang, P., Bunke, H., eds.: Automatic Bankcheck Processing. World Scientific (1997)
5. Gopisetty, S., Lorie, R., Mao, J., Mohiuddin, M., Sorin, A., Yair, E.: Automated Forms-processing Software and Services. IBM Journal of Research and Development **40**(2) (1996) 211–230
6. Ye, X., Cheriet, M., Suen, C.: A Generic Method of Cleaning and Enhancing Handwritten Data from Business Forms. Int. Journal on Document Analysis and Recognition **4**(2) (2001) 84–96
7. Arica, N., Yarman-Vural, F.: An Overview of Character Recognition Focused on Off-line Handwriting. IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews **31**(2) (2001) 216–233
8. Mori, S., Suen, C., Yamamoto, K.: Historical Review of OCR Research and Development. In O'Gorman, L., Kasturi, R., eds.: Document Image Analysis. IEEE Computer Society Press (1995) 244–273
9. Simon, J.C.: Off-line Cursive Word Recognition. Proceedings of the IEEE **80**(7) (1992) 1150–1161
10. Steinherz, T., Rivlin, E., Intrator, N.: Offline Cursive Script Word Recognition – a Survey. Int. Journal on Document Analysis and Recognition **2**(2) (1999) 90–110
11. Vinciarelli, A.: A Survey on Off-line Cursive Word Recognition. Pattern Recognition **35**(7) (2002) 1433–1446
12. Marti, U.V., Bunke, H.: Using a Statistical Language Model to Improve the Performance of an HMM-based Cursive Handwriting Recognition System. Int. Journal of Pattern Recognition and Artificial Intelligence **15**(1) (2001) 65–90

13. Kittler, J., Hatef, M., Duin, R., Matas, J.: On Combining Classifiers. IEEE Trans. on Pattern Analysis and Machine Intelligence **20**(3) (1998) 226–239
14. Roli, F., Kittler, J., Windeatt, T., eds.: Proc. 5th Int. Workshop on Multiple Classifier Systems, Cagliari, Italy, Springer (2004)
15. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
16. Lorette, G.: Handwriting Recognition or Reading? – What is the Situation at the Dawn of the 3rd Millenium? Int. Journal on Document Analysis and Recognition **2**(1) (1999) 2–12
17. Rosenfeld, R.: Two Decades of Statistical Language Modeling: Where do We Go from Here? Proc. of the IEEE **88**(8) (2000) 1270–1278
18. Elliman, D., Sherkat, N.: A Truthing Tool for Generating a Database of Cursive Words. In: Proc. 6th Int. Conf. on Document Analysis and Recognition, Seattle, WA, USA (2001) 1255–1262
19. Guyon, I., Haralick, R., Hull, J., Phillips, I.: Data Sets for OCR and Document Image Understanding Research. In Bunke, H., Wang, P., eds.: Handbook of Character Recognition and Document Image Analysis. World Scientific (1997) 779–799
20. Marti, U.V., Bunke, H.: The IAM-Database: an English Sentence Database for Off-line Handwriting Recognition. Int. Journal on Document Analysis and Recognition **5**(1) (2002) 39–46
21. Cano, J., Pérez-Cortes, J., Arlandis, J., Llobet, R.: Training Set Expansion in Handwritten Character Recognition. In: Proc. 9th SSPR / 4th SPR, Windsor, Ontario, Canada (2002) 548–556
22. Günter, S., Bunke, H.: Multiple Classifier Systems in Offline Handwritten Word Recognition – On the Influence of Training Set and Vocabulary Size. Int. Journal of Pattern Recognition and Artificial Intelligence **18**(7) (2004) 1302–1320
23. Rowley, H., Goyal, M., Bennett, J.: The Effect of Large Training Set Sizes on Online Japanese Kanji and English Cursive Recognizers. In: Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition, Niagara-on-the-Lake, Ontario, Canada (2002) 36–40
24. Velek, O., Nakagawa, M.: The Impact of Large Training Sets on the Recognition Rate of Off-line Japanese Kanji Character Classifiers. In: Proc. 5th IAPR Workshop on Document Analysis Systems, Princeton, New Jersey, USA (2002) 106–109
25. Varga, T., Bunke, H.: Generation of Synthetic Training Data for an HMM-based Handwriting Recognition System. In: Proc. 7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland (2003) 618–622
26. Varga, T., Bunke, H.: Off-line Handwritten Textline Recognition Using a Mixture of Natural and Synthetic Training Data. In: Proc. 17th Int. Conf. on Pattern Recognition, Cambridge, United Kingdom (2004) 545–549
27. Kasturi, R., O'Gorman, L., Govindaraju, V.: Document Image Analysis: A Primer. Sādhanā **27**(1) (2002) 3–22
28. Nagy, G., Nartker, T., Rice, S.: Optical Character Recognition: An Illustrated Guide to the Frontier. In: Proc. IS&T/SPIE Symposium on Electronic Imaging: Science and Technology. Volume 3967., San Jose, CA, USA (2000) 58–69
29. Rice, S., Jenkins, F., Nartker, T.: The Fifth Annual Test of OCR Accuracy. Technical Report ISRI-TR-96-01, University of Nevada, Las Vegas, Nevada, USA (1996)

30. Baird, H.: The State of the Art of Document Image Degradation Modeling. In: Proc. 4th IAPR Workshop on Document Analysis Systems, Rio de Janeiro, Brasil (2000) 1–13
31. Stork, D.: Toward a Computational Theory of Data Acquisition and Truthing. In Helmbold, D., Williamson, B., eds.: Computational Learning Theory. Volume 2111 of Lecture Notes in Computer Science. Springer (2001) 194–207
32. Vuurpijl, L., Niels, R., Erp, M., Schomaker, L., Ratzlaff, E.: Verifying the UNIPEN devset. In: Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition, Kokubunji, Tokyo, Japan (2004) 586–591
33. Baird, H.: Document Image Defect Models. In Baird, H., Bunke, H., Yamamoto, K., eds.: Structured Document Image Analysis. Springer (1992) 546–556
34. Doermann, D., Yao, S.: Generating Synthetic Data for Text Analysis Systems. In: Proc. 4th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, USA (1995) 449–467
35. Kanungo, T., Haralick, R., Phillips, I.: Global and Local Document Degradation Models. In: Proc. 2nd Int. Conf. on Document Analysis and Recognition, Tsukuba Science City, Japan (1993) 730–734
36. Baird, H.: Document Image Defect Models and their Uses. In: Proc. 2nd Int. Conf. on Document Analysis and Recognition, Tsukuba Science City, Japan (1993) 62–67
37. Ho, T., Baird, H.: Large-Scale Simulation Studies in Image Pattern Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence **19**(10) (1997) 1067–1079
38. Märgner, V., Pechwitz, M.: Synthetic Data for Arabic OCR System Development. In: Proc. 6th Int. Conf. on Document Analysis and Recognition, Seattle, WA, USA (2001) 1159–1163
39. Baird, H., Fossey, R.: A 100-Font Classifier. In: Proc. 1st Int. Conf. on Document Analysis and Recognition, St.-Malo, France (1991) 332–340
40. Baird, H., Nagy, G.: A Self-Correcting 100-Font Classifier. In: Proc. IS&T/SPIE Symposium on Electronic Imaging: Science and Technology. Volume 2181., San Jose, California, USA (1994) 106–115
41. Ho, T., Baird, H.: Evaluation of OCR Accuracy Using Synthetic Data. In: Proc. 4th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, USA (1995) 413–422
42. Helmers, M., Bunke, H.: Generation and Use of Synthetic Training Data in Cursive Handwriting Recognition. In: Proc. 1st Iberian Conf. on Pattern Recognition and Image Analysis, Puerto de Andratx, Mallorca, Spain (2003) 336–345
43. Drucker, H., Schapire, R., Simard, P.: Improving Performance in Neural Networks Using a Boosting Algorithm. In Hanson et. al., S., ed.: Advances in Neural Information Processing Systems 5. Morgan Kaufmann (1993) 42–49
44. Ha, T., Bunke, H.: Off-line Handwritten Numeral Recognition by Perturbation Method. IEEE Trans. on Pattern Analysis and Machine Intelligence **19**(5) (1997) 535–539
45. Mao, J., Mohiuddin, K.: Improving OCR Performance Using Character Degradation Models and Boosting Algorithm. Pattern Recognition Letters **18** (1997) 1415–1419
46. Mori, M., Suzuki, A., Shio, A., Ohtsuka, S.: Generating New Samples from Handwritten Numerals based on Point Correspondence. In: Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition, Amsterdam, The Netherlands (2000) 281–290

47. Kaufmann, G., Bunke, H., Ha, T.: Recognition of Cursively Handwritten Words Using a Combined Normalization/Perturbation Approach. In Downton, A., Impedovo, S., eds.: Progress in Handwriting Recognition. World Scientific (1997) 21–28
48. Setlur, S., Govindaraju, V.: Generating Manifold Samples from a Handwritten Word. Pattern Recognition Letters **15** (1994) 901–905
49. Velek, O., Liu, C.L., Nakagawa, M.: Generating Realistic Kanji Character Images from On-line Patterns. In: Proc. 6th Int. Conf. on Document Analysis and Recognition, Seattle, WA, USA (2001) 556–560
50. Govindan, V., Shivaprasad, A.: Artificial Database for Character Recognition Research. Pattern Recognition Letters **12**(10) (1991) 645–648
51. Tung, C., Lee, H.: Performance Analysis of an OCR System via a Handwritten Character Generator. Pattern Recognition **27**(2) (1994) 221–232
52. Simard, P., Cun, Y., Denker, J.: Efficient Pattern Recognition Using a New Transformation Distance. In et al., S.H., ed.: Advances in Neural Information Processing Systems 5. Morgan Kaufmann (1993) 50–58
53. Guyon, I.: Handwriting Synthesis from Handwritten Glyphs. In: Proc. 5th Int. Workshop Frontiers in Handwriting Recognition, Essex, United Kingdom (1996) 309–312
54. Choi, H., Cho, S., Kim, J.: Generation of Handwritten Characters with Bayesian Network based On-line Handwriting Recognizers. In: Proc. 7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland (2003) 995–1001
55. Choi, H., Cho, S., Kim, J.: Writer-Dependent Online Handwriting Generation with Bayesian Networks. In: Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition, Kokubunji, Tokyo, Japan (2004) 130–135
56. Wang, J., Wu, C., Xu, Y.Q., Shum, H.Y., Ji, L.: Learning based Cursive Handwriting Synthesis. In: Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition, Niagara-on-the-Lake, Ontario, Canada (2002) 157–162
57. Lee, D.H., Cho, H.G.: A New Synthesizing Method for Handwriting Korean Scripts. Int. Journal of Pattern Recognition and Artificial Intelligence **12**(1) (1998) 46–61
58. Plamondon, R., Guerfali, W.: The Generation of Handwriting with Delta-lognormal Synergies. Biological Cybernetics **78** (1998) 119–132
59. Baird, H., Coates, A., Fateman, R.: PessimalPrint: a Reverse Turing Test. Int. Journal on Document Analysis and Recognition **5**(2-3) (2003) 158–163
60. Ahn, L., Blum, M., Hopper, N., Langford, J.: The CAPTCHA Web Page. `http://www.captcha.net` (2000)
61. Ahn, L., Blum, M., Hopper, N., Langford, J.: CAPTCHA: Telling Humans and Computers Apart. In Biham, E., ed.: Advances in Cryptology. Volume 2656 of Lecture Notes in Computer Science. Springer (2003) 294–311
62. Ahn, L., Blum, M., Langford, J.: Telling Humans and Computers Apart Automatically – How Lazy Cryptographers Do AI. Communications of the ACM **47**(2) (2004) 57–60
63. Baird, H., Luk, M.: Protecting Websites with Reading-based CAPTCHAs. In: Proc. 2nd Int. Web Document Analysis Workshop, Edinburgh, Scotland (2003) 53–56
64. Baird, H., Riopka, T.: ScatterType: a Reading CAPTCHA Resistant to Segmentation Attack. In: Proc. 12th SPIE/IS&T Document Recognition and Retrieval Conference. Volume 5676., San Jose, California, USA (2005)

65. Chew, M., Baird, H.: BaffleText: a Human Interactive Proof. In: Proc. 10th SPIE/IS&T Document Recognition and Retrieval Conference. Volume 5010., Santa Clara, California, USA (2003) 305–316
66. Lillibridge, M., Abadi, M., Bharat, K., Broder, A.: Method for Selectively Restricting Access to Computer Systems. U.S. Patent No. 6,195,698 (2000)
67. Simard, P., Szeliski, R., Couvreur, J., Calinov, I.: Using Character Recognition and Segmentation to Tell Computer from Humans. In: Proc. 7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland (2003) 418–423
68. Rusu, A., Govindaraju, V.: Handwritten CAPTCHA: Using the Difference in the Abilities of Humans and Machines in Reading Handwritten Words. In: 9th Int. Workshop on Frontiers in Handwriting Recognition, Kokubunji, Tokyo, Japan (2004) 226–231
69. Manzanera, A., Bernard, T.: Improved Low Complexity Fully Parallel Thinning Algorithm. In: Proc. 10th Int. Conf. on Image Analysis and Processing, Venice, Italy (1999) 215–220
70. Soille, P.: Morphological Image Analysis. Springer (1999)
71. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE **77**(2) (1989) 257–286