Summary. This chapter presents an off-line, text-independent system for writer identification and verification. At the core of the system are Gaussian Mixture Models (GMMs). GMMs provide a powerful yet simple means of representing the distribution of features extracted from the text lines of a writer. For each writer, a GMM is built and trained on text lines of that writer. In the identification or verification phase, a text line of unknown origin is presented to each of the models. As a result of the recognition process each model returns a log-likelihood score. These scores are used for both the identification and verification task. Three types of confidence measures are defined on the scores: simple score based, cohort model based, and world model based confidence measures. Experiments demonstrate a very good performance of the system on the identification and the verification task.

Key words: Writer Identification, Writer Verification, Off-Line Handwriting, Gaussian Mixture Model

Off-line Writer Identification and Verification Using Gaussian Mixture Models

Andreas Schlapbach and Horst Bunke

Institute of Computer Science, University of Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
{schlpbch,bunke}@iam.unibe.ch

1 Introduction

In recent years, significant progress has been made in recognizing a person based on biometric features [15, 16, 17]. Different biological traits such as face, fingerprint, iris, signature, and voice are being used to identify a person or verify its identity. This paper addresses the problem of personal identification and verification based on a person's handwriting.

Writer identification is the task of determining the author of a sample handwriting from a set of writers [29]. Related to this task is writer verification, i.e., the task of determining whether or not a handwritten text has been written by a certain person. If any text may be used to establish the identity of the writer, the task is *text independent*. Otherwise, if a writer has to write a particular predefined text to identify himself or herself, or to verify his or her identity, the task is *text dependent*.

If temporal and spatial information about the writing is available, writer identification and verification can be performed *on-line*, otherwise if only a scanned image of the handwriting is available the recognition is performed *off-line*. The system we propose in this paper performs text independent writer identification and verification using off-line handwritten text lines. Examples of handwritten text lines from our database, produced by different writers, are given in Fig. 1. Possible applications of our system include forensic writer identification [41], the retrieval of handwritten documents from a database [1] or authorship determination of historical manuscripts [2].

In this paper we use Gaussian Mixture Models (GMMs) to model a person's handwriting. GMMs provide a powerful yet simple means of representing the distribution of features extracted from text lines written by a person. Formally, a GMM consists of a weighted sum of uni-modal Gaussian densities. While GMMs have been used in speech recognition [32, 33] they have not yet been applied to off-line, text independent writer identification and verification, to the best of our knowledge.

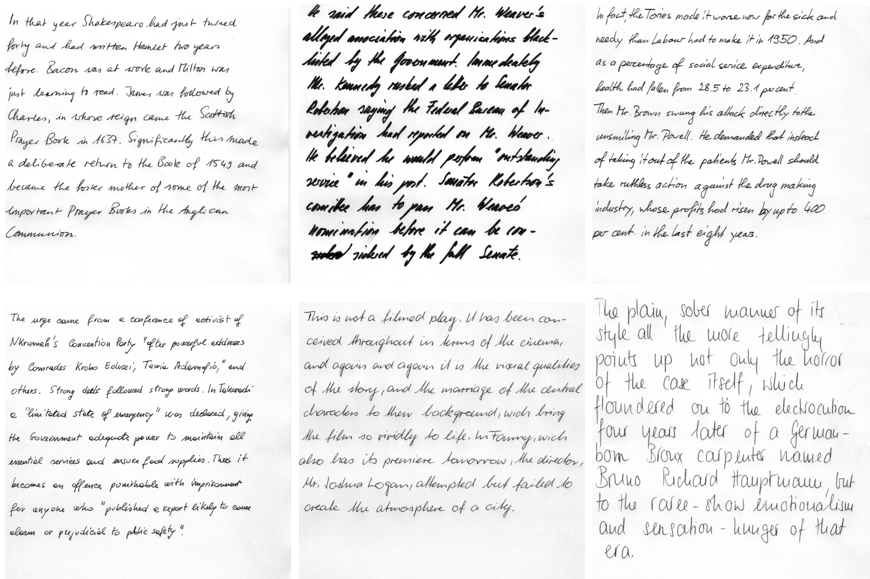


Fig. 1. Examples of text lines

For each writer in the considered population, an individual GMM is trained using data from that writer only. Thus for n different writers we obtain n different GMMs. Intuitively, each GMM can be understood as an expert specialized in recognizing the handwriting of one particular person. Given an arbitrary text line as input, each GMM outputs a recognition score. Assuming that the recognition score of a model is higher on input from the writer the model is trained on than on input from other writers, we can utilize the scores produced by the different GMMs for the task of identifying the writer of a text line or of verifying whether a text line has actually been written by the person who claims to be the writer.

Our approach has several advantages compared to other approaches: GMMs have a mathematically simple, well understood structure and there exist standard algorithms for training and testing [33]. For every writer there is exactly one model which is trained with a set of simple features. We do not need to model characters or words. Therefore we do not need a transcription of the text, but can use any unlabeled text for training and testing. This property makes our system language independent.

The rest of this paper is structured as follows. In Sect. 2 we present related work in the field of writer identification and verification. The GMMs used by our system are introduced in Sect. 3. An overview of our writer identification and verification system is given in Sect. 4 and in Sect. 5 we present several confidence measures for our system. Results of a number of experiments are presented and discussed in Sect. 6. Finally, Sect. 7 concludes the paper and proposes future work.

2 Related Work

Surveys covering work in automatic writer identification and signature verification until 1993 are given in [19, 29]. Recently, several additional approaches to writer identification and verification have been proposed.

Said et al. [35] treat the writer identification task as a texture analysis problem. They use global statistical features extracted from the entire image of a text using multi-channel Gabor filtering and grey-scale co-occurrence matrix techniques.

Srihari et al. [10, 42] address the problem of writer verification, i.e., the problem of determining whether two documents are written by the same person or not. In order to identify the writer of a given document, they model the problem as a classification problem with two classes, *authorship* and *non-authorship*. Given two handwriting samples, one of known and the other of unknown identity, the distance between two documents is computed. Then the distance value is used to classify the data as positive or negative.

Zois et al. [43] base their approach on single words by morphologically processing horizontal projection profiles. The projections are partitioned into a number of segments from which feature vectors are extracted. A Bayesian classifier and a neural network are then applied to the feature vectors.

In Hertel et al. [14] a system for writer identification is described. The system first segments a given text into individual text lines and then extracts a set of features from each text line. The features are subsequently used in a k-nearest-neighbor classifier that compares the feature vector extracted from a given input text to a number of prototype vectors coming from writers with known identity.

Bulacu et al. [9] use edge-based directional probability distributions as features for the writer identification task. The authors introduce edge-hinge distribution as a new feature. The key idea behind this feature is to consider two edge fragments in the neighborhood of a pixel and compute the joint probability distribution of the orientations of the two fragments. Additionally, in [39] the histogram of connected-component contours (CO^3) for upper-case handwriting is introduced as a new feature. Combining this feature with the edge-hinge feature achieves better results than each of the features used separately. In [40] this approach is extended to mixed-style handwriting using fragmented connected-component contours.

In a number of papers [4, 5, 26] graphemes are proposed as features for describing the individual properties of handwriting. Furthermore, it is shown that each handwriting can be characterized by a set of invariant features, called the writer's invariants. These invariants are detected using an automatic grapheme clustering procedure. In [6] these graphemes are used to address the writer verification task based on text blocks as well as on handwritten words.

Leedham et al. [20] present a set of eleven features which can be extracted easily and used for the identification and the verification of documents containing handwritten digits. These features are represented as vectors, and by using the Hamming distance measure and determining a threshold value for the intra-author variation a high degree of accuracy in authorship detection is achieved.

Previously, we have proposed to use Hidden Markov Model (HMM) [31] based text recognizers for the purpose of writer identification and verification [36, 37]. For each writer, an individual recognizer is built and trained on text lines of that writer. This results in a number of recognizers, each of which is an expert on the handwriting of exactly one writer. Assuming that correctly recognized words have a higher score than incorrectly recognized words and that the recognition rate of a system is higher on input from the writer the system was trained on than on input from other writers, the scores produced by the different HMMs are used to decide who has written the input text line.

In this paper, instead of HMM based recognizers, we use GMMs to model a persons handwriting. While GMMs have been used in the speech recognition community [32, 33], they have not been applied, to the best of our knowledge, to off-line writer identification and verification. A GMM can be viewed as a single-state HMM with a Gaussian mixture observation density. The advantages of using GMMs over HMMs are manifold. First, GMMs are conceptually less complex than HMMs consisting of only one state and one output distribution function, which leads to significantly shorter training times. Second, in GMMs only the parameters of the output distribution function have to be estimated during training compared to HMMs where the state transition probabilities have to be estimated as well. Third, neither words nor characters have to be modeled using GMMs, because every writer is represented by exactly one model. Finally, no transcription of the text lines are needed during training.

3 Gaussian Mixture Models

We use Gaussian Mixture Models (GMMs) to model the handwriting of each person of the underlying population. The distribution of the feature vectors extracted from a person's handwriting is modeled by a Gaussian mixture density. For a D-dimensional feature vector $\mathbf{x}$ the mixture density for a specific writer is defined as

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} w_i p_i(\mathbf{x}). \tag{1}$$

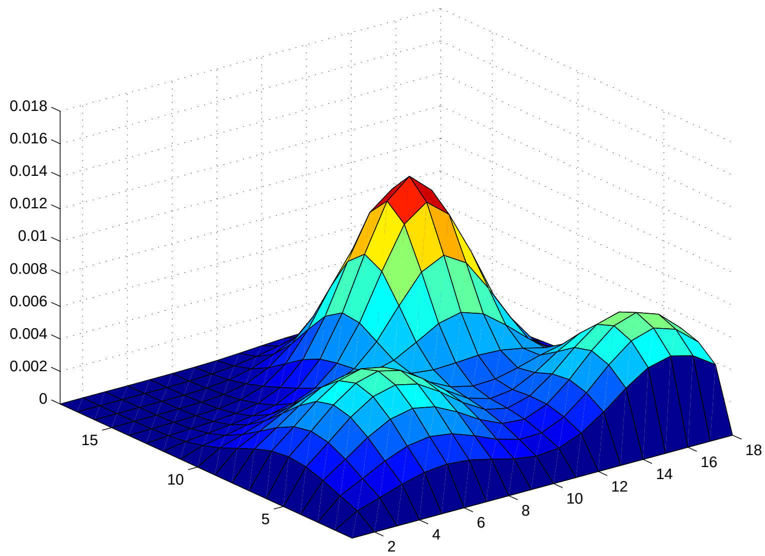


Fig. 2. A two-dimensional GMM consisting of a weighted sum of three uni-modal Gaussian densities

where the mixture weights w_i sum up to one. The mixture density is a weighted linear combination of M uni-modal Gaussian densities $p_i(\mathbf{x})$, each parametrized by a $D \times 1$ mean vector μ_i and a $D \times D$ covariance matrix C_i:

$$p_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|C_i|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_i)'(C_i)^{-1}(\mathbf{x} - \mu_i)\}. \quad (2)$$

The parameters of a writer's density model are denoted as $\lambda = \{w_i, \mu_i, C_i\}$ for all $i = 1, \ldots, M$. This set of parameters completely describes the model and enables us to concisely model a person's handwriting.

The GMMs are trained using the Expectation-Maximization (EM) algorithm [12]. The EM algorithm follows the *Maximum Likelihood (ML)* principle by iteratively refining the parameters of the GMM to monotonically increase the likelihood of the estimated model for the observed feature vectors. The algorithm starts with a data set X of N feature vectors $\mathbf{x}_j$, an initial set of M uni-modal Gaussian densities $N_i \hat{=} N(\mu_i, C_i)$, and M mixture weights w_i. Then, in the first step, for each training data point $\mathbf{x}_j$ the responsibility of each component N_i is determined. In the second step, the component densities, i.e., the mean vector μ_i and the variance matrix C_i for each component and the weights w_i are re-estimated based on the training data. These two steps are repeated until the likelihood score of the entire data set does not change substantially or a limit on the number of iterations is reached.

While the general model supports full covariance matrices, often only diagonal covariance matrices are used. This simplification is motivated by the following observations: first, theoretically the density modeling of an M dimensional full covariance matrix can equally well be achieved using a larger order diagonal covariance matrix. Second, diagonal covariance matrices are computationally more efficient than full covariance matrices, and third, diagonal matrix GMMs outperformed full matrix GMMs in various experiments [33]. An example of a two dimensional GMM with a diagonal covariance matrix is shown in Fig. 2.

The Gaussian component densities can either be initialized randomly or by using vector quantization techniques such as k-means clustering [13]. Furthermore, often variance flooring is employed to avoid an overfitting of the variance parameters [25]. The idea of variance flooring is to impose a lower bound on the variance parameters as a variance estimated from only few data points can be very small and might not be representative of the underlying distribution of the data [25]. The minimal variance value is defined by

$$\sigma^2_{min} = \alpha * \sigma^2_{global} \tag{3}$$

where α denotes the *variance flooring factor* and the global variance σ^2_{global} is calculated on the complete data set. The minimal variance, σ^2_{min}, is used to initialize the variance parameters of the model. During the EM update step, if a calculated variance parameter is smaller than σ^2_{min} then the variance parameter is set to this value.

During decoding, the feature vectors $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ extracted from a text line are assumed to be independent. The log-likelihood score of a model λ for a sequence of feature vectors X is defined as

$$\log p(X|\lambda) = \sum_{t=1}^{T} \log p(\mathbf{x}_t|\lambda), \tag{4}$$

where $p(\mathbf{x}_t|\lambda)$ is computed according to Eq. 1.

In this work, we use diagonal covariance matrices and the models are initialized using k-means clustering. The GMMs are implemented using the Torch library [11].

4 System Overview

We use GMMs as the building blocks of our writer identification and verification system. A schematic overview of the system is shown in Fig. 3. For each writer, a GMM as described in the previous section is built and trained with data coming from this writer only. As a result of the training procedure, we get a model for each writer.

A set of features is extracted from each text line to train the GMMs. Before feature extraction, a series of normalization operations are applied to each text line. The operations are designed to improve the quality of the features extracted without removing writer specific information.

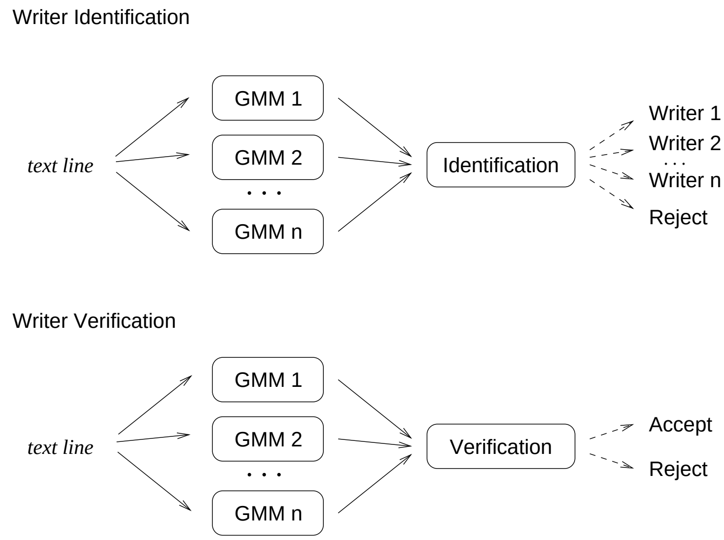


Fig. 3. Schematic overview of the writer identification and verification system

For the purpose of normalization, the contrast of the grey-scale images is enhanced first, leading to images with black strokes written on white background. Then vertical scaling and thinning normalization operations are applied, which are described in the following two paragraphs. The aim of vertical scaling is to normalize the height of the text line and thinning assures independence of the writing pen.

To perform vertical scaling a text line is divided into three zones: a zone containing the ascenders, a middle zone, and a zone containing the descenders. These three zones are to be normalized to a predefined height which is important in order to reduce the variability of the features used to train the GMMs. To actually perform this operation, the upper and the lower baseline of the text line have to be determined. To find the two baselines, the histogram of the horizontal projection of the image of the text line is used. The real histogram is matched with an ideal histogram. The location of the upper and the lower baseline are detected and the three main writing zones are determined. Each of these three zones is then individually positioned and scaled to a predefined height.

Different pens of different width have been used to write the text lines. In order to eliminate the effect of the pen width on the performance of the system, all text lines are thinned using the iterative MB2 thinning algorithm [7]. After thinning, all strokes in a text line image are at most two pixels wide. In Fig. 4 a text line before and after normalization and thinning is shown.

In the next step, features are extracted by a sliding window. The window moves from left to right one pixel per step. For every column of pixels in the

Text line before normalization and thinning

Text line after normalization and thinning

Fig. 4. A text line before and after normalization and thinning

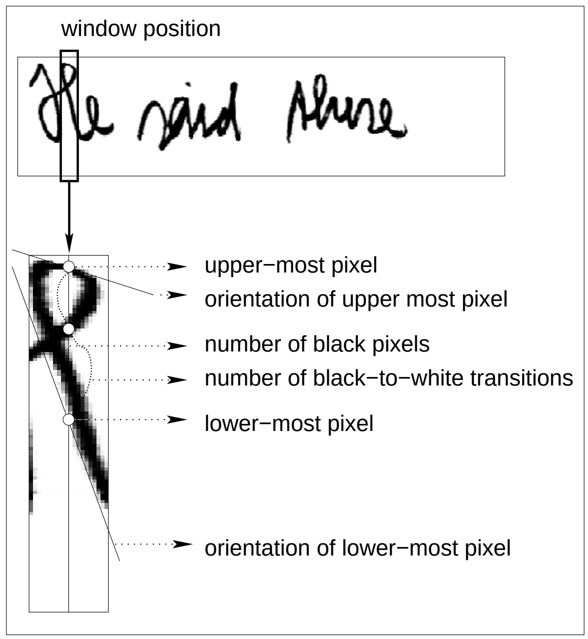


Fig. 5. Six local features extracted from each row in the sliding window

sliding window, nine geometrical features are extracted. These features have shown to produce good results on both the text recognition task [21] as well as on the writer identification and verification task [38].

The feature set consists of three global and six local features. The three global features describe the distribution of the pixels in the column, e.g., the fraction of black pixels in the window, the center of gravity and the second order moment. The six local features describe specific points in the column. The features describe the position and the orientation of the upper- and the lower-most pixel, the number of black-to-white transitions in the window, and the fraction of black pixels between the upper- and the lower-most black pixel (see Fig. 5 for an illustration of the six local features). The feature vectors of every column in the sliding window are averaged to produce the final feature vector. At last, the feature vectors which only describe white space are deleted.

The width of the sliding window was optimized in an independent experiment involving 571 text lines from 20 writers. These 20 writers are not part of the data set used to train the GMM models in the subsequent experiments. A fixed number of 100 Gaussian mixture components and a variance flooring factor of 0.001 were used for training. The window width was varied from 2 to 32 by steps of two. The highest writer identification rate of 99.05% was achieved using a window width of 14 pixels. This window width was used in all subsequent experiments to extract the features from a text line.

The sequences of nine-dimensional feature vectors extracted from the text lines are used to train the GMMs. After training, each GMM is especially adapted to the individual handwriting style of a particular writer. During identification, a text line to be classified is presented to the GMM of each writer. Each GMM outputs a log-likelihood score and a standard deviation for the given text line. These scores are the basis for identification and verification as described below.

5 Confidence Measures

In order to assign a text line to a certain person or to verify the identity of a text line with a claimed identity we need a means of measuring how sure the system is about the given text line. A confidence measure enables us to judge the quality of the recognition and to implement a rejection mechanism based on this measure.

For writer identification we define the following rejection mechanism. If the confidence measure of a text line is above a given threshold, the system returns the identity of the text line with the highest ranked score; otherwise the system rejects the input. Thus if we have n writers, the writer identification problem is a n-class classification problem with a reject option.

The decision criterion for writer verification is similar. If the confidence measure of a text line is above a certain threshold, we assume that the text line was in fact written by the claimed writer; otherwise the input is classified as not being of the claimed identity. In writer verification we deal with a two-class classification problem that is independent of the number of writers under consideration.

Various confidence measures for off-line handwriting recognition have been presented in the literature [23, 27, 28]. In this paper, three common types of confidence measures are used. The first type of confidence measure is solely based on the score of the model under consideration and therefore is not normalized. The other two types of confidence measures normalize the recognition score based on a *cohort model* and a *world model* approach, respectively. The cohort model approach normalizes the score of the model of the claimed writer with respect to the score of the most competitive writers [34]. The world model approach normalizes the score of the claimed writer by a model which is trained on a large number of samples from many writers [24].

5.1 Confidence Measures for Writer Identification

A text line is presented to each model and the returned log-likelihood scores are sorted. Given a text line t of an unknown author, the simplest confidence measure is to judge the quality of the recognition based on the log-likelihood score of the first ranked model:

$$cmIdent_{\mathrm{LLScore}}(t) = ll_{\mathrm{firstRanked}} \tag{5}$$

The next confidence measure is inspired by the cohort model approach. The confidence measure is calculated from the difference of the log-likelihood score of the first ranked model, $ll_{\mathrm{firstRanked}}$, and the log-likelihood score of the second ranked model, $ll_{\mathrm{secondRanked}}$:

$$cmIdent_{\mathrm{CohortModel}}(t) = ll_{\mathrm{firstRanked}} - ll_{\mathrm{secondRanked}} \tag{6}$$

The third confidence measure uses a world model to normalize the log-likelihood score of the first ranked writer. The world model is trained on a large number of text lines coming from different writers. The confidence measure is calculated on the difference of the log-likelihood score of the first ranked writer, $ll_{\mathrm{firstRanked}}$, and the world model, $ll_{\mathrm{worldModel}}$:

$$cmIdent_{\mathrm{WorldModel}}(t) = ll_{\mathrm{firstRanked}} - ll_{\mathrm{worldModel}} \tag{7}$$

All the confidence measures for writer identification presented in this section need to determine the system which produces the highest log-likelihood score. A text line has to be presented to the model of each writer under consideration. Then the returned scores have to be sorted, which means that the calculation of these confidence measures depends on the number of writers.

5.2 Confidence Measures for Writer Verification

The confidence measures for writer verification are similar to the ones defined for writer identification in the previous section. Compared to the writer identification case where the log-likelihood score of the first ranked system is normalized, the log-likelihood score of the claimed system is normalized instead.

The first simple confidence measure for a text line t is the log-likelihood score of the model of the claimed identity, $ll_{\mathrm{claimedID}}$:

$$cmVerif_{\mathrm{LLScore}}(t) = ll_{\mathrm{claimedID}} \tag{8}$$

The next confidence measure is inspired by the cohort model approach. Based on the ranking of the scores the confidence measure is calculated from the difference of the log-likelihood score of the claimed identity, $ll_{\text{claimedID}}$, and the first best ranked competing writer, $ll_{\text{bestRankedCompeting}}$:

$$cmVerif_{\text{CohortModel}}(t) = ll_{\text{claimedID}} - ll_{\text{bestRankedCompeting}} \tag{9}$$

The third confidence measure implements a world model approach. The difference of the score of the model of the claimed identity and the world model is computed:

$$cmVerif_{\text{WorldModel}}(t) = ll_{\text{claimedID}} - ll_{\text{worldModel}} \tag{10}$$

In comparison to the world model based confidence score in the identification case (Eq. 7), in the verification case we do not need to present the text line in question to all the models, but to the model of the claimed identity and the world model only.

6 Experiments

6.1 Data sets

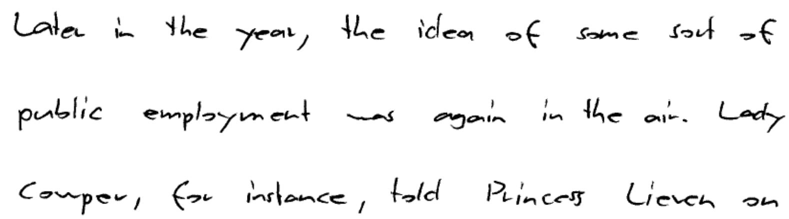
The text lines used in our experiments are part of the IAM handwriting database [22][1]. The database currently contains over 1,500 pages of handwritten text. For each writer we use five pages of text from which between 27 and 54 text lines are extracted.

Five-fold cross validation is used in our experiments. Cross validation enables us to use all text lines for training without committing the error of training or of optimizing meta parameters on the test set [18]. For each writer, the set of available text lines is split into five sets. The idea is to train the system on three sets, use the fourth set to find an optimal set of meta parameters and then test on the fifth set. This procedure is iterated five times such that each set is used for testing once. The final recognition rate is obtained by averaging the five results from each of the test sets.

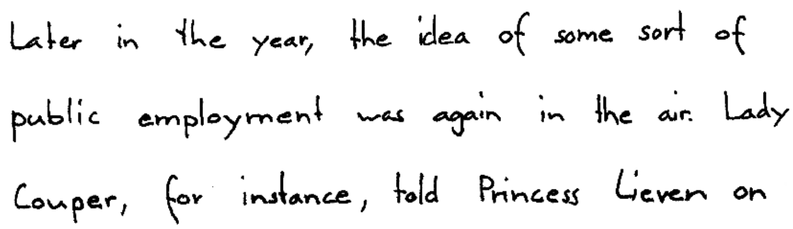
In this experimental setup, the data set consists of text lines from 100 different writers. All in all, 4,103 text lines are available and due to cross validation we can use all the text lines for both training and testing.

A verification system can make two types of errors. First, the system can falsely reject a text written by a client and, second, it can falsely accept a text coming from an impostor [8]. Therefore we need two sets for testing a writer verification system: one set consisting of clients and one set containing impostors. The impostor set can be composed of unskilled forgeries, where the impostor makes no effort to simulate a genuine handwriting, and of skilled forgeries, where the impostor tries to imitate the handwriting of a client as closely as possible [30].

[1] The IAM handwriting database is publicly available at: www.iam.unibe.ch/~fki/iamDB



Example of original text lines



Example of skillfully forged text lines

Fig. 6. Examples of original and skillfully forged text lines

The unskillfully forged test set used in our experiments consists of two disjoint subsets coming from clients and impostors. The unskilled forgeries that form the impostor set are obtained from the database by extracting 571 text lines produced by 20 writers. The writers of these text lines are disjoint from the 100 clients and no model exists that is trained on the handwriting of any of these 20 writers. Based on these text lines the impostor data set is constructed by assigning, to each of these text lines, seven identities of writers known to the system. In total, the impostor data set consists of $7 \times 571 = 3{,}997$ lines and the complete test set of 8,100 text lines. The rationale is that the number of text lines to be accepted should be approximately the same as the number of text lines that have to be rejected, i.e., the two classes under consideration should be balanced.

The skillfully forged test set is again composed of two subsets, a client and an impostor subset. The client data set consists of one page of text each from 20 different writers which are part of the 100 clients. A total of 169 text lines are extracted from these 20 pages. The same 20 pages are then skillfully forged. The acquisition protocol is as follows. A person is presented with a page of handwritten text and given 10 minutes to train the writing. Then he or she is asked to forge the text. An example of three original and three skillfully forged text lines are given in Fig. 6. From the forgeries thus created, another 169 text lines are extracted. Hence, in total 338 text lines are used in this test set.

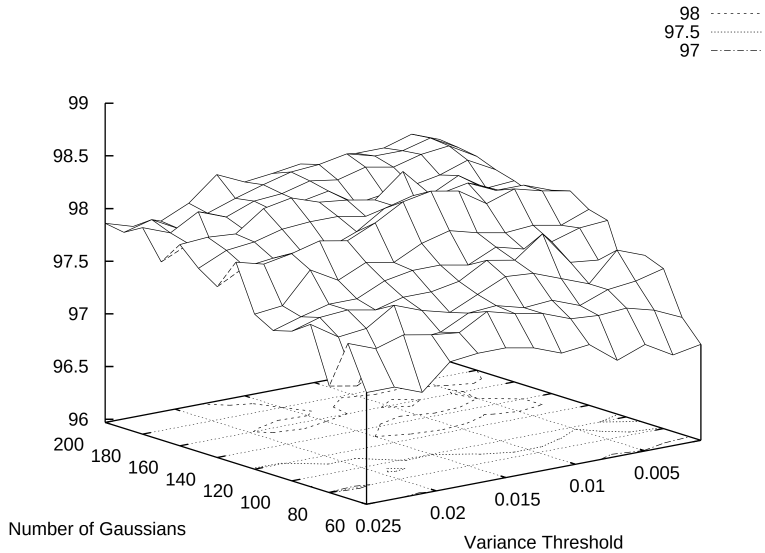


Fig. 7. Writer identification rate as a function of the number of Gaussian mixture components and the variance flooring factor on the validation set

6.2 Writer Identification Experiments

We first conducted an experiment to measure the influence of the number of Gaussian mixture components and the variance flooring factor on the writer identification rate. The number of Gaussian mixture components is varied from 60 to 200 by steps of 10 and the variance flooring factor is varied from 0.001 to 0.025 by steps of 0.002.

The writer identification rate as a function of the number of Gaussian mixture components and the variance flooring factor on the validation set is shown in Fig. 7. On the validation set, the highest writer identification rate of 98.20% is achieved using 130 Gaussian mixture components and a variance flooring factor of 0.011. An identification rate higher than 97.03% is achieved using 60 Gaussian mixture components or more on the validation set. The two meta parameters optimized on the validation set are then used to calculate the final writer identification rate of 97.88% on the test set. We also use the world model trained with these meta parameters in the subsequent experiments.

In Fig. 8, the n-best list is shown where the writer identification rate based on the first n ranks is plotted. The error rate of the system drops below 1% if the first three ranks, and below 0.5% if the first seven ranks are considered.

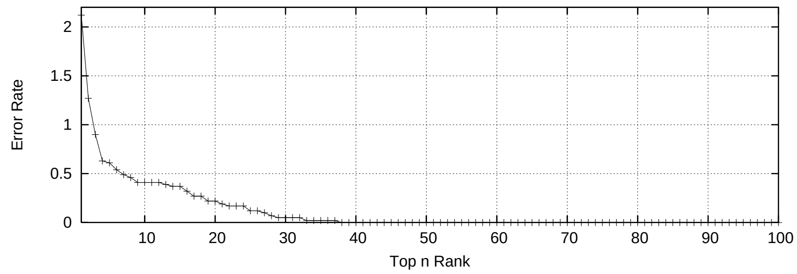


Fig. 8. *n*-best list for the writer identification experiment

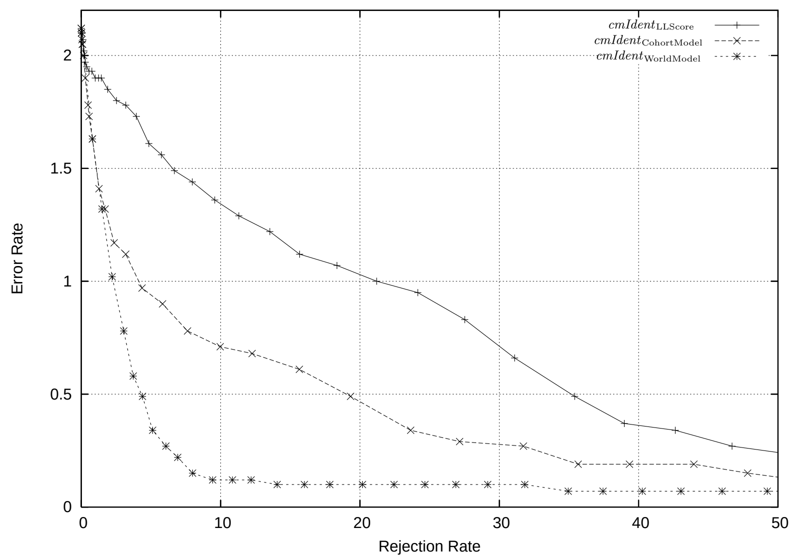


Fig. 9. Error-rejection curves for different confidence measures

The error-rejection curves obtained from the identification test set are shown in Fig. 9. The simple log-likelihood score based confidence score ($cmIdent_{\mathrm{LLScore}}$) produces the lowest performing error-rejection curve. The error rate drops below 1% only if more than 22% of the text lines with the lowest confidence score are rejected. The next best error-rejection curve is produced by the world model based approach ($cmIdent_{\mathrm{WorldModel}}$). The error rate is smaller than 1% if less than 5% of the text lines are rejected. The cohort model based approach ($cmIdent_{\mathrm{CohortModel}}$) yields the best error-rejection curve. Fewer than 5% of the text lines have to be rejected to obtain an error rate smaller than 0.5%.

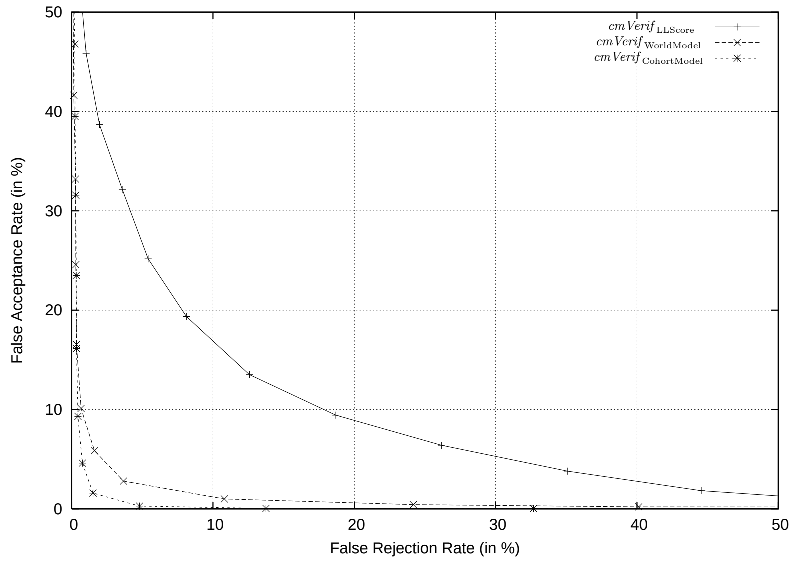


Fig. 10. ROC curves on the unskillfully forged test set

The observation that the cohort model based approach performs best can be explained by the fact that the normalization is based on the actual text line being presented, i.e., the adequate model to normalize the text line is selected anew for each text line. In comparison, the world model approach normalizes the score of a text line by the score of a general world model which is independent of the text line under consideration.

6.3 Writer Verification Experiments

The results of the writer verification experiments are reported as Receiver Operator Characteristic (ROC) curves in Figs. 10 and 11. An ROC curve describes the performance of a verification system on a test set by plotting the False Acceptance Rate (FAR) against the False Rejection Rate (FRR) [8]. In Table 6.3 the estimated Equal Error Rates (EERs) for the ROC curves are given [8]. The Equal Error Rate estimates the point on an ROC curve where the FAR is identical to the FRR.

In Fig. 10 the ROC curves on the unskillfully forged test set are shown. The ROC curves produced by the simple log-likelihood score ($cmVerif_{\text{LLScore}}$) has the lowest performance with an EER of 13.0%. The world model based confidence measure ($cmVerif_{\text{WorldModel}}$) achieves an EER of 3.2%. The ROC curve based on the cohort model approach ($cmVerif_{\text{CohortModel}}$) performs best and yields an EER of around 1.5%.

The ROC curves on the skillfully forged test set for the GMM based systems are shown in Fig. 11. The ROC curve with the lowest performance re-

Table 1. Equal Error Rates (EERs) for the unskillfully and skillfully forged test set

Equal Error Rate (ERR)	Unskilled Forgeries	Skilled Forgeries
$cmVerif_{\mathrm{LLScore}}$	13.0%	41.0%
$cmVerif_{\mathrm{WorldModel}}$	3.2%	18.6%
$cmVerif_{\mathrm{CohortModel}}$	1.5%	9.3%

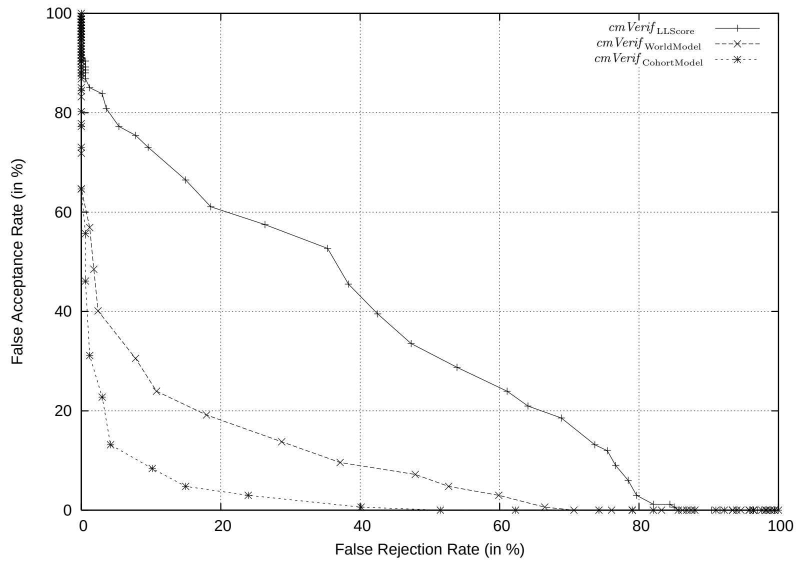


Fig. 11. ROC curves on the skillfully forged test set

sults from the simple log-likelihood score confidence measure ($cmVerif_{\mathrm{LLScore}}$) with an EER of around 41.0%. The world model based confidence measure ($cmVerif_{\mathrm{WorldModel}}$) yields an EER of around 18.6%. The best ROC curve is produced by the cohort model based confidence measure ($cmVerif_{\mathrm{CohortModel}}$) with an EER of around 9.3%.

In both verification experiments, the ROC curves show the same hierarchy of performance: the simple log-likelihood score based confidence measure yields the lowest performing ROC curve, the next best ROC curve is produced by the world model based confidence measure which itself is outperformed by the cohort model based ROC curve. This behavior is consistent with the writer identification case, where the best error-rejection curve is achieved when the score of a text line is normalized with respect to the score of the most competitive writer.

The calculation of the cohort model based confidence measure however is costly compared to the world model based confidence measure. For every text line, the log-likelihood scores of all writers models have to be computed and sorted to determine the best performing model. In comparison, to compute the world model based confidence measure, only the score of the claimed system and the world model is needed and is independent of the number of client models.

7 Conclusion

We have used Gaussian Mixture Models (GMMs) to address the task of off-line text-independent writer identification and verification. GMMs provide a powerful yet simple means of representing the distribution of features extracted from handwritten text lines. A sliding window extracts a sequence of simple, language independent feature vectors from a text line. The feature sequences are used to train one model for each writer. During recognition, a text line of unknown origin is presented to each of the models. Each model returns a log-likelihood score for the given input. The scores are the basis for writer identification and verification.

On the writer identification task, a text line is assigned to the writer of the first ranked model if the confidence measure is above a given threshold. We achieve a correct writer identification rate of 97.88% in a 100 writers experiment using 4,103 text lines. If we consider not only the first, but the three highest ranked writers, in over 99.0% of all cases the writer of the text line under question is correctly identified. Furthermore, if we reject fewer than 5% of the text lines with the lowest confidence score, the writer identification rate improves to over 99.5% using the best performing confidence measure.

Similarly, on the writer verification task a text line is accepted if its confidence score is above a certain threshold; otherwise it is rejected. Two sets of experiments have been conducted: the unskillfully forged test set contains in total 8,100 text lines from 100 clients and 20 impostors. The skillfully forged test set contains 338 text lines from 20 clients and 20 impostors. An Equal Error Rate (EER) of around 1.5% is achieved on the unskillfully forged test set and an EER of approximately 9.3% is obtained on the skillfully forged test set by the best confidence measure.

Three types of confidence measures have been presented in this paper: simple score based, cohort model based and world model based confidence measures. For both writer identification and verification, the cohort model based confidence measure performs best. This observation can be explained by the fact that the normalization depends on the actual text line being presented, i.e., the relevant model to normalize the text line is selected anew for each text line. In comparison, the world model confidence measure normalizes the score of a text line by the score of a general world model.

In future work we plan to measure the influence of using less data to train the GMMs. A possible approach would be to use a universal background model [33] and then adapt this model to a specific writer model. Another interesting question is to investigate whether modifications of the world model based confidence measure as presented in [3] would yield performances similar to the ones obtained by the cohort model based confidence measure. Furthermore we plan to compare the performance of this system to the HMM based system developed previously.

Acknowledgement. This research is supported by the Swiss National Science Foundation NCCR program "Interactive Multimodal Information Management (IM2)" in the Individual Project "Visual/Video Processing".

References

1. Henry S. Baird. Digital libraries and document image analysis. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 2–14, 2003.
2. Henry S. Baird and Venu Govindaraju, editors. *Proc. 1st Int. Workshop on Document Image Analysis for Libraries*. IEEE Computer Society, 2004.
3. Claude Barras and Jean-Luc Gauvain. Feature and score normalization for speaker verification of cellular data. In *Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 49–52, 2003.
4. Ameur Bensefia, Ali Nosary, Thierry Paquet, and Laurent Heutte. Writer identification by writer's invariants. In *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 274–279, 2002.
5. Ameur Bensefia, Thierry Paquet, and Laurent Heutte. Information retrieval based writer identification. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 946–950, 2003.
6. Ameur Bensefia, Thierry Paquet, and Laurent Heutte. Handwriting analysis for writer verification. In *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 196–201, 2004.
7. Thierry M. Bernard and Antoine Manzanera. Improved low complexity fully parallel thinning algorithm. In *Proc. 10th Int. Conf. on Image Analysis and Processing*, pages 215 – 220, 1999.
8. Frédéric Bimbot and Gérard Chollet. Assessement of speaker verification systems. In Dafydd Gibbon, Roger Moore, and Richard Winski, editors, *Handbook of Standards and Resources for Spoken Language Systems*, pages 408–480. Mouton de Gruyter, 1997.
9. Marius Bulacu, Lambert Schomaker, and Louis Vuurpijl. Writer identification using edge-based directional features. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 937–941, 2003.
10. Sung-Hyuk Cha and Sargur N. Srihari. Multiple feature integration for writer verification. In *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 333–342, 2000.
11. Ronan Collobert, Samy Bengio, and Johnny Mariéthoz. Torch: a modular machine learning software library. IDIAP-RR 46, IDIAP, 2002.
12. Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39:1–38, 1977.

13. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* Wiley Interscience, 2001.
14. Caroline Hertel and Horst Bunke. A set of novel features for writer identification. In *Audio- and Video-Based Biometric Person Authentication*, pages 679–687, 2003.
15. Anil K. Jain, R. Bolle, and S. Pankanti, editors. *Biometrics – Personal Identification in Networked Society.* Springer, 2002.
16. Anil K. Jain, Lin Hong, and Sharath Pankanti. Biometric identification. *Communications of the ACM*, 43:91–98, 2000.
17. Josef Kittler and Mark S. Nixon, editors. *Audio- and Video-Based Biometric Person Authentication.* Springer, 2003.
18. Ludmila I. Kuncheva. *Combining pattern classifiers: methods and algorithms.* Wiley-Interscience, 2004.
19. Franck Leclerc and Réjean Plamondon. Automatic signature verification: The state of the art 1989–1993. In Réjean Plamondon, editor, *Progress in Automatic Signature Verification*, pages 13–19. World Scientific Publ. Co., 1994.
20. Graham Leedham and Sumit Chachra. Writer identification using innovative binarised features of handwritten numerals. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 413–417, 2003.
21. Urs-Viktor Marti and Horst Bunke. 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*, 15:65–90, 2001.
22. Urs-Viktor Marti and Horst Bunke. The IAM–database: An English sentence database for off-line handwriting recognition. *Int. Journal of Document Analysis and Recognition*, 5:39–46, 2002.
23. Sanparith Marukatat, Thierry Artières, Patrick Gallinari, and Bernadette Dorizzi. Rejection measures for handwriting sentence recognition. In *Proc. 8th Int. Conf. on Frontiers in Handwriting Recognition*, pages 25–29, 2002.
24. Tomoko Matsui and Sadaoki Furui. Likelihood normalization for speaker verification using a phoneme- and speaker-independent model. *Speech Communications*, 17:109–116, 1995.
25. Håkan Melin, Johan Koolwaaij, Johan Lindberg, and Frédéric Bimbot. A comparative evaluation of variance flooring techniques in HMM-based speaker verification. In *Proc. 5th Int. Conf. on Spoken Language Processing*, pages 2379–2382, 1998.
26. Ali Nosary, Laurent Heutte, Thierry Paquet, and Yves Lecourtier. Defining writer's invariants to adapt the recognition task. In *Proc. 5th Int. Conf. on Document Analysis and Recognition*, pages 765–768, 1999.
27. John F. Pitrelli and Michael P. Perrone. Confidence modeling for verification post-processing for handwriting recognition. In *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 30–35, 2002.
28. John F. Pitrelli and Michael P. Perrone. Confidence-scoring post-processing for off-line handwritten-character recognition verification. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 278–282, 2003.
29. Réjean Plamondon and Guy Lorette. Automatic signature verification and writer identification – the state of the art. In *Pattern Recognition*, volume 22, pages 107–131, 1989.

30. Réjean Plamondon and Sargur N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:63–84, 2000.
31. Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. of the IEEE*, volume 77, pages 257–285, 1989.
32. Douglas A. Reynolds. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17:91–108, 1995.
33. Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.
34. Aaron E. Rosenberg, Joel Deong, Chin-Hui Lee, Biing-Hwang Juang, and Frank K. Soong. The use of cohort normalized scores for speaker verification. In *Proc. Int. Conf. on Spoken Language Processing*, pages 599–602, 1992.
35. H. E. S. Said, Tienu N. Tan, and Keith D. Baker. Personal identification based on handwriting. *Pattern Recognition*, 33:149–160, 2000.
36. Andreas Schlapbach and Horst Bunke. Off-line handwriting identification using HMM based recognizers. In *Proc. 17th Int. Conf. on Pattern Recognition*, volume 2, pages 654–658, 2004.
37. Andreas Schlapbach and Horst Bunke. Using HMM based recognizers for writer identification and verification. In *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 167–172, 2004.
38. Andreas Schlapbach and Horst Bunke. Off-line writer identification and verification using gaussian mixture models. In *Learning in Document Analysis and Recognition*. Springer, 2007. to appear.
39. Lambert Schomaker and Marius Bulacu. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:787–798, 2004.
40. Lambert Schomaker, Marius Bulacu, and Katrin Franke. Automatic writer identification using fragmented connected-component contours. In *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 185–190, 2004.
41. Sargur Srihari and Zhixin Shi. Forensic handwritten document retrieval system. In *Proc. 1st Int. Workshop on Document Image Analysis for Libraries*, pages 188–194, 2004.
42. Bin Zhang, Sargur N. Srihari, and Sangjik Lee. Individuality of handwritten characters. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, volume 7, pages 1086–1090, 2003.
43. Elias N. Zois and Vassilis Anastassopoulos. Morphological waveform coding for writer identification. *Pattern Recognition*, 33:385–398, 2000.