International Journal of Pattern Recognition and Artificial Intelligence

AUTOMATIC GENDER DETECTION – COMBINING ON-LINE AND OFF-LINE SYSTEMS

MARCUS LIWICKI, ANDREAS SCHLAPBACH, and HORST BUNKE

Institut für Informatik und angewandte Mathematik
Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
{liwicki,schlpbch,bunke}@iam.unibe.ch
http://www.iam.unibe.ch/~ liwicki

In this paper, the problem of classifying handwritten data with respect to gender is addressed. A state-of-the-art classification method based on Gaussian Mixture Models is appled to distinguish between *male* and *female* handwriting. Two sets of features have been used for the classification, on-line and off-line features, respectively. In our experiments, the on-line features produced a higher classification rate. Furthermore, we combined both feature sets and investigated several combination strategies. The maximum voting finally turned out to give the best results. The final gender detection rate on the test set is 67.57 % which is significantly higher than the performance of the on-line system with about 64.25 %. The combined system also shows an improved performance over human-based classification.

Keywords: Gender Detection, Handwriting Analysis, Gaussian Mixture Models, Multiple Classifier Combination

1. Introduction

A population of individuals can often be partitioned into sub-categories based on various criteria. Dividing a population into sub-categories is interesting for numerous reasons, for example, if a researcher is only interested in one specific subcategory, or if specifically processing each sub-category leads to improved results. For example, in the field of face recognition, much research has been conducted on classifying a face image according to gender[27,28]. Classification results up to 94% have been reported for this two-class problem.

For handwriting there exist several criteria for sub-categories. Whereas in KANSEI the sub-categories are feelings for character patterns[11], handwriting can also be divided into writer-specific sub-categories including gender, handedness, age and ethnicity[24]. Correlations between these sub-categories and handwriting features have been presented in Ref. 13. Special interest has been focused on determining the gender of the writer. In Ref. 10 humans were asked to classify the writer's gender of a given handwriting example. A classification rate of about 68% has been reported. Further studies in Ref. 3, which inlcude a detailed analysis of the raters background, reported results in the same range.

Beside being an interesting research topic of its own, automatically identifying sub-categories can be used to improve a handwriting recognition system. The variability within a certain category is smaller than within a complete population, which allows us to train specialized recognizers. Another application is demographic studies. A concrete example would be to study the handwriting available on the world wide web and to find out how many people from each category contributed to the data.

Especially the classification of gender from handwriting has been a research topic for many decades [4,22,25]. However, there exist conflicting results ranging from slightly more than 50% to more than 90%. An overview of several manual approaches detecting gender from handwriting can be found in Ref. 12. This thesis tries to semi-automatically classify the handwriting, while it is done automatically in this paper.

Little work exists on automatically identifying sub-categories, such as gender or handedness, from handwriting. In Ref. 5 a system for classifying the handwriting based on images of individual letters is presented. Results of 70.2% for gender classification and 59.5% for handedness have been achieved. If longer texts are available and multiple classifier approaches are applied even better results are reported[2]. However, these systems are restricted to the off-line case and either the transcription of the text has to be known or even identical texts have to be provided by all writers.

In this paper we present a system that classifies gender of on-line, Roman handwriting. This problem is a two class problem, i.e., *male/female*. On-line handwriting means that temporal information about the handwriting is available. The handwriting is unconstrained, thus any text can be used for classification. However, two feature sets are investigated in this paper. While the first feature set is based on online features, the second set of features is extracted from off-line images generated from the on-line data. We applied Gaussian Mixture Models (GMMs) to model the classes. In our experiments, the classifier working with on-line features outperforms the off-line classifier. Furthermore, we combined both feature sets and investigated several combination strategies. The maximum voting finally turned out to give the best results. **:TODO:** **:TODO: Do we really want to present our results in the introduction already?** The final gender detection rate on the test set is 67.57 % which is significantly higher than the performance of the on-line system with about 64.25 %. For the purpose of comparison, also an experiment with humans classifying the same on-line data set is performed.

Note that preliminary results on the topic of gender detection have been published in Ref. 18. However, Ref. 18 is only loosely related to the current paper because several enhancements are present in this paper. First, a new classifier based on off-line features is used in this paper. Second, a combination of both classifiers is investigated in this paper. Next, the experimental evaluation has been improved, i.e., while the purpose of Ref. 18 was to find the highest possible performance on the validation set, and independent test set is used in this paper. Finally, the com-

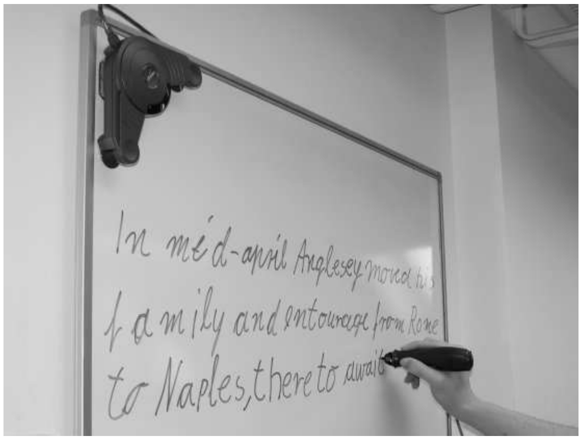

Fig. 1. Illustration of the recording process

parison with human performance has been elaborated, i.e., in this paper the on-line information has been made available to the humans.

The rest of the paper is organized as follows. Section 2 introduces the normalization and the features extracted from the data. Section 3 describes the GMM classifier. The combination strategy is described in Section 4. Experiments and results are presented and discussed in Section 5, while Section 6 draws some conclusions and gives an outlook for future work.

2. Data Acquisition and Feature Extraction

:TODO: Argue that both the on-line and off-line feature sets have shown excellent performance on the writer identification and verification task (Re

:TODO:

To acquire the handwritten data, the eBeam[a]interface is used. It outputs a sequence of (x, y)-coordinates representing the location of the tip of the pen together with a time stamp for each location. The data is in XML-format and the frame rate of the recordings varies from 30 to 70 frames per second. An illustration of the recording process is shown in Fig. 1.

A series of normalization operations are applied before feature extraction. The operations intend to improve the quality of the features without removing writer specific, i.e. class specific information. The recorded on-line data contain noisy points and gaps within strokes, which are caused by loss of sampling data during acquisition. In Fig. 2 examples of both types of distortion are shown. In the word *await* a spurious point occurs that leads to the introduction of a large artifact, i.e. two long additional strokes. Furthermore, we observe in the first line that there are many gaps within the text, which are caused by loss of data. Thus we first

[a]eBeam System by Luidia, Inc. – www.e-Beam.com

apply some on-line preprocessing operations to recover from these artifacts. These operations are described in Ref. 16. An example of the results of these preprocessing steps is shown in Fig. 3. Obviously, now the handwriting is of much better quality.

The cleaned paragraph of text is then automatically divided into lines using a simple heuristics. If there is a pen-movement to the left and down greater than a predefined threshold the start of a new line is assumed.

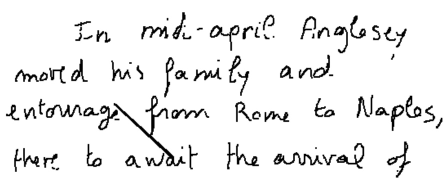

Fig. 2. Recorded text

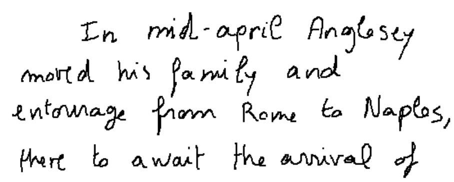

Fig. 3. Text after removing noise

2.1. *On-Line Feature Set*

For the on-line features, the next step is to divide each text line into sub-parts which can then be normalized independently of each other. For each sub-part the skew angle is corrected to horizontally align the text. Next each sub-part is divided into three regions: the upper area, which contains the ascenders of the letters; the median area with the corpus of the letters; and the lower area containing the descenders of the letters. These three areas are normalized to predefined heights. Finally, the width of each sub-part is normalized. The number of characters is estimated as a fraction of the number of strokes crossing the horizontal line between the base line and the corpus line. The text is then horizontally scaled according to this value.

The on-line feature set used in this experiment contains on-line features as well as pseudo off-line features extracted from an off-line representation of the on-line data. (In the remainder of this section, the number in round brackets behind the

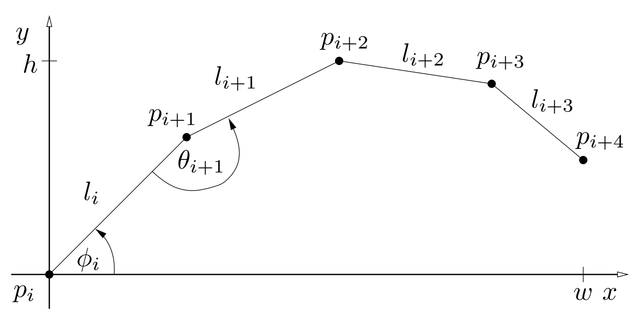

Fig. 4. Features extracted from the on-line handwriting.

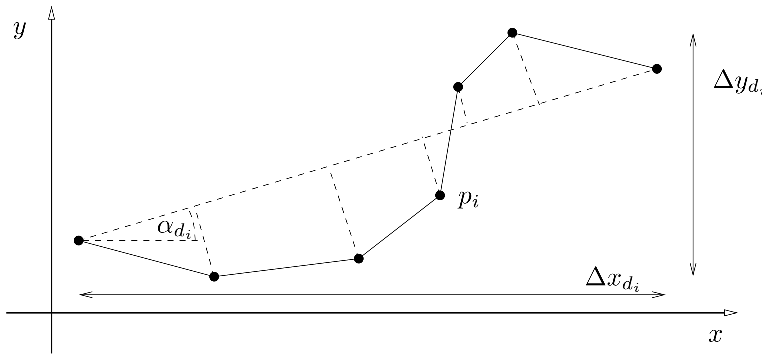

Fig. 5. Illustration of vicinity features.

name of a feature indicates the number of individual features.) For a given stroke s consisting of points p_1 to p_n, the following 18 on-line features for each consecutive pair of points (p_i, p_{i+1}) are computed (see Fig. 4 for an illustration):

- *x/y-coordinate (2):* the relative x/y-position of the point p_i. The relative x-coordinate is calculated by subtracting the x-coordinate of a point from a moving average coordinate.
- *speed (1):* the speed v_i of the segment
- *speed in x/y-direction (2):* the speed v_{i_x}/v_{i_y} in x/y-direction
- *acceleration (1):* the overall acceleration a_i
- *acceleration in x/y-direction (2):* the acceleration a_{i_x}/a_{i_y} in x/y-direction
- *log curvature radius (1):* the curvature radius is the length of the circle which best approximates the curvature at the point p_i. It is derived from the local velocities and the local accelerations as follows:

$$r = \frac{(v_{i_x} * a_{i_y} - a_{i_x} * v_{i_y})}{\sqrt{(v_{i_x}^2 + v_{i_y}^2)}^3}$$

- *writing direction (2):* the cosine and the sine of the angle between the line segment of the starting point and the x-axis
- *curvature (2):* the cosine and sine of the angle between the lines to the previous and to the next point
- *vicinity aspect (1):* the aspect of the trajectory in the vicinity $d_i = \{p_{i-n}, \ldots, p_i, \ldots, p_{i+n}\}$ of the point p_i:

$$va = \frac{\Delta y_{d_i} - \Delta x_{d_i}}{\Delta y_{d_i} + \Delta x_{d_i}}$$

It characterizes the ratio of height to width of the bounding box containing the preceding and the succeeding points[14]. Fig. 5 illustrates the computation of this feature. The vicinity of a point is also used to de-

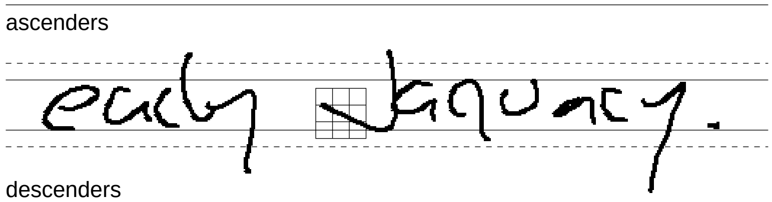

Fig. 6. Illustration of pseudo off-line features.

fine the following three features: vicinity curliness, vicinity linearity, and vicinity slope.

- *vicinity curliness (1):* this feature describes the deviation from a straight line in the vicinity d_i (see Fig. 5). It is computed from the length of the trajectory in the vicinity divided by $\max(\Delta x_{d_i}, \Delta y_{d_i})$[14].
- *vicinity linearity (1)*: the average square distance between every point in the vicinity and the straight line linking the first and the last point in the vicinity[14].
- *vicinity slope (2):* the cosine and the sine of the angle α_{d_i} of the straight line from the first to the last point in the vicinity (see Fig. 5)[14].

The pseudo off-line features are computed using a two-dimensional matrix representing an off-line version of the data. The matrix is obtained by projecting the on-line strokes on the two-dimensional plane (see Fig. 6 for an illustration). The following 11 features are used:

- *ascenders/descenders (2):* the number of points above/below the corpus line whose x-coordinates are in the vicinity of the point and which have a minimal distance to the corpus/base line (denoted by the two dashed lines in Fig. 6). The distance is set to a predefined fraction of the corpus height.
- *context map (9):* the two-dimensional vicinity of the point is divided into three regions for each dimension (illustrated by the 3×3 matrix in Fig. 6). The number of black points in each region is taken as a feature value.

Overall the feature set consists of 29 features.[17]

2.2. *Off-Line Feature Set*

Since the preprocessed data is still in the on-line format, it has to be transformed into an off-line image, so that it can be used as input for the off-line recognizer. The recognizer was originally designed for the off-line IAM-Database[20] and optimized on gray-scale images scanned with a resolution of 300 dpi. To get good recognition results in the considered application, the produced images should be similar to these off-line images. Consequently the following steps are applied to generate the images.

First all consecutive points within the same stroke are connected. This results in one line segment per stroke. Then the lines are dilated to a width of eight pixels. The center of each line is colored black and the pixels are getting lighter towards the periphery. Fig. 7 shows an example of a generated image. Compared to Figs. 2 and 3 the handwriting looks more similar to the IAM-Database (see Fig. 8). In general, the realistic generation of off-line data is a quite complex problem. Ref. 26 proposes methods to create images that look even more similar to scanned images. In the experiments reported in Ref. 26 the recognizer was trained and tested on computer generated images, and the best performance has been achieved using a constant thickness. During our experiments the recognition rate increased when we supplemented this simple approach with the generation of different gray values.

Fig. 7. Generated gray-scale image

Fig. 8. Image of the IAM-Database (produced by a writer different from Fig 7)

Unlike the on-line data, the normalizations for the off-line data is applied to entire text lines at once. First of all the image was rotated to account for the line skew. Then the mean slant of the text was estimated, and a shearing transformation was applied to the image to bring the handwriting to an upright position. Next the baseline and the corpus line were normalized. Normalization of the baseline means that the body of the text line (the part which is located between the upper and lower baselines), the ascender part (located above the upper baseline), and the descender part (below the lower baseline) are vertically scaled to a predefined size each. Finally the image was scaled horizontally so that the mean character width was approximately equal to a predefined size.

Figure 9 illustrates the off-line preprocessing. The first line shows the original image, whereas the normalized image appears on the second line. The normalization includes skew correction, slant correction, baseline positioning, and width normalization.

To extract the feature vectors from the normalized images, a sliding window approach was used. The width of the window was one pixel, and nine geometrical features were computed at each window position. Each text line image was therefore converted to a sequence of 9-dimensional vectors. The nine features were as follows:

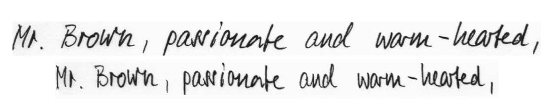

Fig. 9. Pre-processing of an image of handwritten text

- The mean gray value of the pixels
- The center of gravity of the pixels
- The second order vertical moment of the center of gravity
- The positions of the uppermost and lowermost black pixels
- The rate of change of these positions (with respect to the neighbouring windows)
- The number of black-white transitions between the uppermost and lowermost pixels
- The proportion of black pixels between the uppermost and lowermost pixels

For a more detailed description of the off-line features, see Ref. 19.

3. Gaussian Mixture Models

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

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

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:

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

The parameters of a sub-category'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 sub-category's writing on the whiteboard.

While the general model supports full covariance matrices, often only diagonal covariance matrices are used. An example of the two dimensional case is shown in Fig. 10. 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 [23].

Instead of training a sub-category model from scratch for every sub-category, we obtain the models of the sub-categories from a *Universal Background model (UBM)*. The basic idea is to derive the sub-category's model by updating the well-trained parameters from the UBM. In a first step, all data from all writers is used to train a single, writer independent UBM. In the second step, for each sub-category

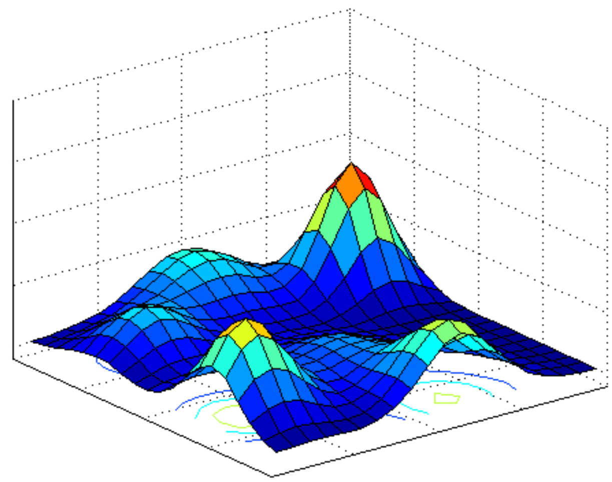

Fig. 10. A two-dimensional GMM consisting of a weighted sum of six uni-modal Gaussian densities.

a sub-category dependent model is build by updating the parameters in the UBM via adaptation using all training data from this sub-category.

The UBM is trained using the Expectation-Maximization (EM) algorithm [7]. 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 $\mathbf{x}_n$, 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 x_n the responsibility $P(i|x_n)$ 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.

The Gaussian component densities of the UBM can either be initialized randomly or by using vector quantization techniques such as k-means clustering [8]. Furthermore, variance flooring is employed to avoid an overfitting of the variance parameter [21]. 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 [21]. The minimal variance value is defined by

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

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

The sub-category models are obtained from the UBM by a modified version of the EM algorithm based on the *Maximum a Posteriori (MAP)* principle. The MAP approach provides a way of incorporating prior information in the training process which is particularly useful for dealing with problems posed by sparse training data for which the ML approach gives inaccurate estimates [9].

Similarly to the EM algorithm, the MAP adaptation algorithm consists of two steps. The first step is identical to the expectation step of the EM algorithm, where estimates of the statistics of the sub-category's training data are computed for each mixture component in the UBM. Unlike the second step of the EM algorithm, however, for adaptation these new statistical estimates are then combined with the old statistics from the UBM mixture parameters using a data-dependent mixture coefficient. This adaptation coefficient (called *MAP adaptation factor*) controls the adaptation process by emphasizing either on the well-trained data of the UBM or on the new data when estimating the parameters [23].

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.

Diagonal covariance matrices are used and initialized by k-means clustering in our system. The number of clusters equals the number of Gaussian mixture components. The GMMs are implemented using the Torch library[6].

4. Combination

:TODO: :TODO: Ask Prof. Bunke for good references.

After decoding, each classifier returns a log-likelihood score, i.e., the on-line classifier returns $ll_{\text{on}-\text{line}}$ and the off-line classifier returns $ll_{\text{off}-\text{line}}$. To combine the results of the on-line and the off-line classifier, the following standard rules for the classifier combination on the score level have been applied [1]:

- *Average Rule*
  The log-likelihood scores of both the on-line and the off-line classifiers are averaged: $ll_{\text{sum}} = \frac{1}{2} * (ll_{\text{on}-\text{line}} + ll_{\text{off}-\text{line}})$.

- *Maximum Rule*

The largest log-likelihood score is chosen: $ll_{\max} = \max(ll_{\text{on-line}}, ll_{\text{off-line}})$.

- *Minimum Rule*
  The smallest log-likelihood score is chosen: $ll_{\min} = \min(ll_{\text{on-line}}, ll_{\text{off-line}})$.

The range of log-likelihood scores of both classifiers vary greatly. Therefore, before combination the results of both classifiers are normalized in respect to mean and standard deviation. Due to the fact, that only two classifiers have been used, other combination rules such as the median rule or voting are not applicable in this case.

5. Experiments and Results

The experiments have been conducted on the IAM-OnDB, a large on-line handwriting database acquired from a whiteboard[15][b]. This database consists of data from more than 200 writers with eight handwritten texts per writer. Each text consists of seven text lines on average. The classification task is to identify the correct gender for a given text line.

For the task of gender classification we randomly selected 40 male and 40 female writers for training the classifiers, 10 male and 10 female writers for the validation of meta-parameters, and 25 male and 25 female writers for testing the final system. This assures that both classes are equally distributed in all sets, the training, the validation, and the test set.

For the GMM the number of Gaussian mixture components G have been optimized between 1 and 250. Next, the variance flooring factor φ has been varied between 0.001 and 0.011 in steps of 0.002. Furthermore, the MAP adaptation factor α has been varied from full adaptation, i.e, $\alpha = 0$ to no adaptation, i.e., $\alpha = 1$ steps of 0.2.

In Table 1 the results on the validation set are given. The first column gives the type of classifier, the second column describes the meta parameters and the third column shows the classification rate on the validation set.

Table 2 shows the classification results for the gender classification task on the test set. The best combination achieved a performance of 67.57 %. This result is significantly higher than the classification of the best individual classifier (using a *z-test* with $\alpha = 0.05$).

To compare the performance of the classifiers to that of humans, we asked 20 persons to classify 24 movies of handwriting from different writers each.[c] For these movies we took writers from the test set. For “training” purposes, these humans also had classified images from other writers available. The average classification rate of the humans is about 63.88 % for the gender recognition task.

[b] http://www.iam.unibe.ch/~fki/iamondb/
[c] The test is available under http://www.iam.unibe.ch/~smueller/

Table 1. Gender classification rates on the validation set

Classifier	Meta Parameters	Classification Rate
On-line	G: 231, $\varphi: 0.8$, $\alpha: 0.009$	69.98 %
Off-line	G: 221, $\varphi: 0.4$, $\alpha: 0.001$	57.88 %
Combination (*Average Rule*)	G: 221, $\varphi: 0.8$, $\alpha: 0.001$	64.51 %
	G: 241, $\varphi: 0.8$, $\alpha: 0.001$	64.51 %
Combination (*Maximum Rule*)	G: 231, $\varphi: 0.2$, $\alpha: 0.009$	70.98 %
	G: 231, $\varphi: 0.6$, $\alpha: 0.011$	70.98 %
Combination (*Minimum Rule*)	G: 221, $\varphi: 0.4$, $\alpha: 0.001$	58.37 %

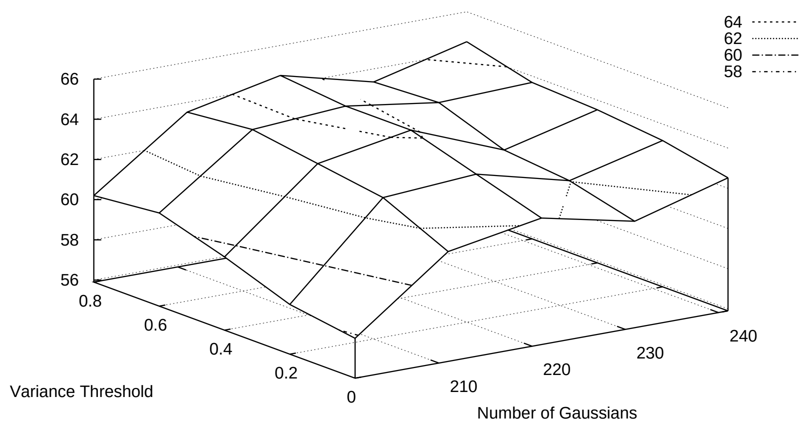

Fig. 11. Optimizing the number of Gaussian mixtures and the MAP adaptation factor.

Table 2. Gender classification rates on the test set

Classifier	Classification Rate
On-line	64.25 %
Off-line	55.39 %
Combination	67.57 %

:TODO: **:TODO: Discuss the difference between the results on the validation and on the test set.**

Table 3. Gender classification rates of several combination approaches

Classifier Combination	Classification Rate
Average Rule	67.57 %
Maximum Rule	61.93 %
Minimum Rule	65.58 %

6. Conclusions and Future Work

In this paper we have presented a system for the classification of gender of handwriting. The data is given in on-line format and we extract two feature sets, a set of 29 on-line features, and a set of 9 off-line features. For the classification, we used Gaussian Mixture Models (GMMs).

In our experiments classification results higher than human classification have been achieved. The GMM results for gender classification are similar to results reported on another data set by Ref. 5.

The combination of classifier's results on the score level has shown to significantly increase the classification rates. Other approaches to combine the classifier's results exist, such as combination on the feature or on the decision level. Comparison of the results presented in this paper with these approaches is left for future work.

:TODO: Think: Is gender verification an interesting topic for future work? :TODO:

Acknowledgments

This work was supported by the Swiss National Science Foundation program "Interactive Multimodal Information Management (IM)2" in the Individual Project "Visual/Video Processing", as part of NCCR. Special thanks go to Petra and Michael Liwicki, who provided assistance in getting references from the Humboldt-University, Berlin. Furthermore, we thank all volunteers who participated in the on-line classification test.

References

1.
2. K. Bandi and S. N. Srihari. Writer demographic classification using bagging and boosting. In *Proc. 12th Int. Graphonomics Society Conference*, pages 133–137, 2005.
3. John R. Beech and Isla C. Mackintosh. Do differences in sex hormones affect handwriting style? evidence from digit ratio and sex role identity as determinants of the sex of handwriting. *Personality and individual differences*, 39(2):459–468, 2005.
4. ME. Broom, B. Thompson, and et al. Sex differences in handwriting. *Journal of Applied Psychology*, 13:159–166, 1929.
5. Sung-Hyuk Cha and Sargur N. Srihari. Apriori algorithm for sub-category classification analysis of handwriting. In *Proc. 6th Int. Conf. on Document Analysis and Recognition*, pages 1022–1025, 2001.

6. R. Collobert, S. Bengio, and J. Mariéthoz. Torch: a modular machine learning software library. IDIAP-RR 46, IDIAP, 2002.
7. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39(1):1–38, 1977.
8. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* Wiley Interscience, 2001.
9. J. Gauvain and C. Lee. Maximum a-posteriori estimation for multivariate Gaussian mixture observation of Markov chains. *IEEE Trans. on Specch & Audio Processing*, 2:291–298, 1994.
10. S. Hamid and K. M. Loewenthal. Inferring gender from handwriting in urdu and english. *The Journal of social psychology*, 136(6):778–782, 1996.
11. Tetsuo Hattori, Tetsuya Izumi, Hiroyuki Kitajima, and Toshinori Yamasaki. Kansei information extraction from character patterns using a modified fourier transform. In *Proc. Sino-Japan Symposium on KANSEI & Artificial Life*, pages 36–39, 2004.
12. Mandred R. Hecker. *Die Untersuchung der Geschlechtsspezifitt der Handschrift mittels Rechnergesttzter Merkmalsextraktionsverfahren.* PhD thesis, Humboldt-University, Berlin, 1996.
13. Roy A. Huber. *Handwriting Identification: Facts and Fundamentals.* CRC Press, 1999.
14. S. Jaeger, S. Manke, J. Reichert, and A. Waibel. Online handwriting recognition: the NPen++ recognizer. *Int. Journal on Document Analysis and Recognition*, 3(3):169–180, 2001.
15. M. Liwicki and H. Bunke. IAM-OnDB - an on-line English sentence database acquired from handwritten text on a whiteboard. In *Proc. 8th Int. Conf. on Document Analysis and Recognition*, volume 2, pages 956–961, 2005.
16. M. Liwicki and H. Bunke. Handwriting recognition of whiteboard notes – studying the influence of training set size and type. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 21(1):83–98, 2007.
17. M. Liwicki, A Schlapbach, H. Bunke, S. Bengio, J. Mariéthoz, and J. Richiardi. Writer identification for smart meeting room systems. In *Proc. 7th IAPR Workshop on Document Analysis Systems*, volume 3872 of *LNCS*, pages 186–195. Springer, 2006.
18. M. Liwicki, A. Schlapbach, P. Loretan, and H. Bunke. Automatic detection of gender and handedness from on-line handwriting. In *Proc. 13th Conf. of the Int. Graphonomics Society*, pages 179–183, 2007.
19. U.-V. Marti and H. 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.
20. U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5:39–46, 2002.
21. H. Melin, J.W. Koolwaaij, J. Lindberg, and F. 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.
22. SM. Newhall. Sex differences in handwriting. *Journal of Applied Psychology*, 10:151–161, 1926.
23. D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.
24. T. Scheidat, F. Wolf, and C. Vielhauer. Analyzing handwriting biometrics in metadata context. In *Proc. 8th SPIE conference at the Security, Steganography, and Watermarking of Multimedia Contents*, volume 6072, pages 182–193, 2006.
25. H. Tenwolde. More on sex differences in handwriting. *Journal of Applied Psychology*,

18:705–710, 1934.

26. O. Velek, C.-L. Liu, S. Jaeger, and M. Nakagawa. An improved approach to generating realistic Kanji character images from on-line characters and its benefit to off-line recognition performance. In *Proc. 16th Int. Conf. on Pattern Recognition*, pages 588–591, 2002.
27. L. Wiskott, J.-M. Fellous, N. Krger, and C. von der Malsburg. Face recognition and gender determination. In *Proc. Int. Workshop on Automatic Face- and Gesture-Recognition*, pages 92–97, 1995.
28. Bo Wu, Haizhou Ai, and Chang Huang. *Audio- and Video-Based Biometric Person Authentication*, volume 2688 of *LNCS*, chapter *LUT-Based Adaboost for Gender Classification*, pages 104–110. Springer, 2003.