Semi-Supervised Learning for Character Recognition in Historical Archive Documents

Jan Richarz [*] Szilard Vajda [**] Rene Grzeszick [*]
Gernot A. Fink [*]

TU Dortmund University, Department of Computer Science, Dortmund, Germany

Abstract

Training recognizers for handwritten characters is still a very time consuming task involving tremendous amounts of manual annotations by experts. In this paper we present semi-supervised labeling strategies that are able to considerably reduce the human effort. We propose two different methods to label and later recognize characters in collections of historical archive documents. The first one is based on clustering of different feature representations and the second one incorporates a simultaneous retrieval on different representations. Hence, both approaches are based on multi-view learning and later apply a voting procedure for reliably propagating annotations to unlabeled data. We evaluate our methods on the MNIST database of handwritten digits and introduce a realistic application in form of a database of handwritten historical weather reports. The experiments show that our method is able to significantly reduce the human effort that is required to build a character recognizer for the data collection considered while still achieving recognition rates that are close to a supervised classification experiment.

Key words: Character recognition; semi-supervised learning; historical documents

1 Introduction

After several thousand years of human history and roughly 2,000 years after the invention of paper, historical archives and museums store tremendous amounts of handwritten documents. They contain information of great value for historians and the wide public.

[*] *Email adress:* **firstname.lastname@udo.edu**
Phone: **(+49)231-755-6151** *Fax:* **(+49)231-755-6116**
[**]*Email adress:* **szilard.vajda@nih.gov**

Accessing this knowledge is typically not an easy task. It is necessary to browse through either printed or digital copies of these documents, which is a very tiresome and time consuming process. With digital copies browsing through those documents became much easier, but it is even more comfortable if they are indexed or transcribed. However, nowadays the transcription of documents is still done manually by experts. Much research has been dedicated to the task of transcribing documents automatically by recognizers, which is a very difficult problem. Scans of old documents are often of bad quality and show various artifacts. In addition, handwritten text shows a very high variability that is dependent on the writer. Typically a recognizer needs to be trained for different scripts and writers, which again requires a tremendous amount of training material that has been annotated before.

So far research has not been able to completely remove the process of manually annotating documents, but in the following we will give an overview of methods for reducing the required manual labeling operations for training a recognizer. The annotation task is executed in a machine-aided manner. Clustering and retrieval operations are used in order to choose representatives that are labeled by an expert annotator.

For evaluation we consider the well known MNIST dataset as well as a realistic set of historical weather reports. In both cases the methods are able to considerably reduce the amount of labeling operations to less than one percent of the original training data. We will show that it is possible to perform labeling with high precision, so that high recognition rates can be achieved with data that has been labeled in a semi-supervised manner.

2 Related Work

The general idea of semi-supervised learning is to reduce the required manual work by combining labeled and unlabeled data (cf. [34]). Typically, in such scenarios the vast majority of data is unlabeled. The known labels must be highly reliable and robustly be propagated to the unknown data.

High reliability of the labels can only be ensured by presenting selected samples to an expert annotator. Additionally, the labeled subset should be representative for the remaining data since propagation is typically achieved by analyzing sample similarity. Consequently, random selection is generally not advisable. We utilize two different approaches for selecting a representative subset. The first (cf. sec. 3.1) relies on clustering and selects the cluster centroids as representatives that are labeled. The second (cf. sec. 3.2) uses a realization of the *active-learning* concept where the system actively selects the data that should get annotated based on its current knowledge in a feedback loop (cf. [26]).

For achieving a robust propagation the concept of *multiview-learning* is adapted by training an *ensemble* of learners (cf. [11,23]). Each of these learners has a different view on the data, e.g., by using different features. Decisions are made by combining the outputs of different learners. A common concept is using a majority vote [11]. The advantages of incorporating ensembles in semi-supervised learning approaches for robust propagation are, for example, discussed in [33].

The problem of propagating a small set of labels to a large dataset has been studied in different fields of research. Applications include, for example, the clusters of text documents [31], image retrieval [3,27] or the active learning of gesture trajectories [25]. Semi-supervised approaches have also been studied in the field of character annotation [1,24]. In [1] it has been shown that the recognition rate of a handwriting recognizer can be improved using self-learning strategies on unlabeled data. However, in all cases an initial set of annotations must be provided manually.

For handwritten graphical multi-stroke symbols an annotation assistance is proposed by Li et al. [13], where the annotation of the symbols is reduced to finding sub-graphs in a relation graph built from different segments. In the graph the nodes are the segments and the arcs represent the spatial relationships between them. The authors show that only 58.2% of the strokes need to be labeled.

With respect to the goal of reducing the manual effort in the transcription of historical documents, the work introduced by Toselli et al. in [29,30] has a similar goal than ours. However, the principle differs from our approach. We propose using a semi-supervised approach to label the data and train a new recognizer for a given document collection, while they rather refine an existing recognizer with feedback from the annotator.

Our own contributions to semi-supervised learning strategies for character labeling have been introduced in [32] for characters of the Lampung script, written in Indonesia and in [21,22] for Latin characters of the dataset of historical weather reports that is also considered in this paper. In the following we present an extended comprehensive overview of our semi-supervised learning methods for character recognition as well as a detailed evaluation of the clustering- and retrieval-based methods on two handwritten character databases.

3 Semi-supervised labeling approaches

In the upcoming sections we present two different methods that allow labeling training data on character level with a minimum amount of manual work: a) clustering-based labeling (CBL), and b) retrieval-based labeling (RBL). Our main goal is not to achieve the best possible classification scores, hence not concentrating on the most appropriate classifier selection and tuning, but rather to show that competitive results can be achieved with semi-supervised approaches using minimal human effort for the labeling process. To achieve this goal a high labeling accuracy is crucial since it strongly influences the subsequent recognition process.

3.1 Clustering-based labeling

In [21,32] we introduced a preliminary version of the clustering based multi-view labeling algorithm for handwritten characters that requires only minimal human effort for labeling the unknown data. The method is illustrated in Figure 1 and can be described by four major steps:

(1) An ensemble of different views of unlabeled data is created using a set of different feature representations.
(2) In all representations the features are clustered unsupervisedly.
(3) A single label is assigned to each cluster center by the expert annotator.
(4) Unanimity voting among the different views is used for determining the label for each data point.

In order to implement an ensemble of representations that have a different view on the data we compute r different setups R_i. A setup is defined as a combination of a feature representation and a clustering method.

In every setup R_i the clustering is computed independently, creating k_i partitions of the data. Note that the number of clusters may vary for each feature representation. Usually the partitions are generated using a vector quantization algorithm, like k-Means clustering [15] or the generalized Lloyd algorithm [14], but other unsupervised methods like Self Organizing Map [10], Growing Neural Gas [8] or Affinity Propagation [7] can also be considered to separate the input space into separate regions.

Once the partitioning is performed, each cluster is labeled manually by an expert annotator. Only the cluster centroids are labeled, all other samples belonging to the same cluster will automatically inherit the label from the centroid. This way, the number of required manual annotations is reduced to $\sum_{i=1}^{r}(k_i)$. Hence, it depends only on the total number of clusters for all feature

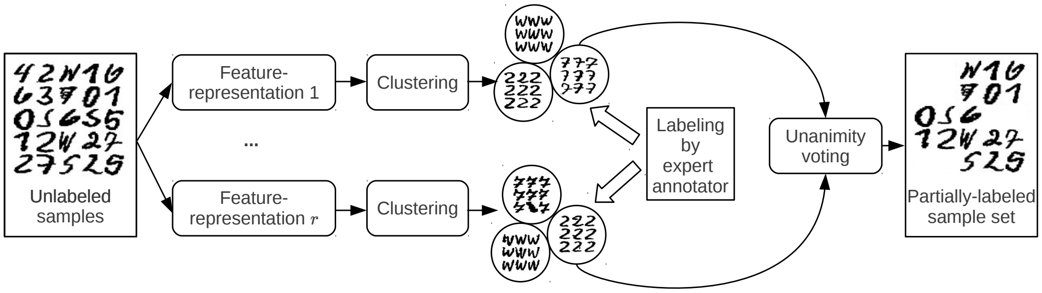
Unlabeled samples
Feature-representation 1
...
Feature-representation r
Clustering
Clustering
Labeling by expert annotator
Unanimity voting
Partially-labeled sample set

Fig. 1. Illustration of the clustering-based labeling approach. Given a set of characters, r different feature representations of the data are computed. Each of those representations is clustered and then a label is assigned to each of those clusters by an expert annotator. Finally, a unanimity voting scheme is applied to assign a label to each character, which results in a partially-labeled sample-set.

representations. Depending on the number of expected classes, large datasets of several thousand samples can easily be labeled using only a few hundred manual annotations.

Considering the number of clusters there are two factors that counteract: The smaller the number of clusters, the less manual work is required, but more clusters will represent the samples more accurately reducing considerably the intra-class and inter-class variances.

The clustering and labeling will usually result in some incorrectly labeled samples due to the limited capacity of the different unsupervised clustering strategies. Assume that the labels are given as d-dimensional binary vectors $[l_{i,1}, \ldots, l_{i,d}]^T \in \{0,1\}^d$, $i = 1, \ldots r$, where $l_{i,j} = 1$ if a sample p is assigned to class ω_j in setup R_i, and 0 otherwise. Applying a majority voting procedure results in an ensemble decision for a specific class label ω_k^{max}. A threshold κ_v on the ensemble decision is used for selecting only those samples where the class membership is determined with high agreement:

$$\omega_k^{max} = \max_k \sum_{i=1}^{r} l_{i,k} \geq \kappa_v. \tag{1}$$

In the following, we use the so-called *unanimity vote* and only retain samples for which all votes agree on the same label ($\kappa_v = r$). This particular voting scheme will provide labels only for a certain amount of data points available in the dataset. In order to avoid introducing noise in the newly labeled data, only those samples will be considered further for training a classifier. The rest of the samples where no unanimity was observed among the different views are discarded from the training set.

Considering the number of parameters that have to be selected heuristically, CBL requires the number of clusters (or some related parameter depending on the clustering method used) for each feature representation and, possibly, the selection threshold in the voting step κ_v as inputs.

Since the feature representations are evaluated independently of each other it is also advisable to limit the number of setups r to a small set of discriminative features. The lower the number of representations and clusters is, the less annotation work must be performed by the human expert.

3.2 Retrieval-based labeling

In [21] we also introduced a method that allows annotating data based on interactive retrieval, which is illustrated in Figure 2. The approach is related to pool-based active learning with relevance feedback (cf. e.g. [28]). In contrast to classical or "passive" learning, where a randomly drawn set of annotated training samples is used for training a classifier in a single training run, the active learning paradigm iteratively refines a classifier based on user feedback and steered sample selection. Initialization is done using a small number of labeled examples. Then, the following steps are iterated until some termination criterion is met:

(1) Given an unlabeled dataset, a classifier and a sample selection function: Select a subset of samples from the data and present it to the annotator.
(2) The annotator manually assigns labels to these samples. In a binary scenario, this corresponds to labeling the retrieved samples as relevant or irrelevant.
(3) Given the newly assigned labels, re-train the classifier.

However, our method differs in a few important aspects from this paradigm. Most importantly, we want to retrieve labels for all possible classes (quasi-) simultaneously. Additionally, selecting relevant samples manually from a potentially very large retrieval list and presenting all of them to an expert annotator counteracts the goal of reducing the burden for the annotator, especially in the case of large multi-class datasets. Consequently, the manual relevance feedback step is replaced by a simple automatic selection rule on the retrieval list, propagating the annotation to unlabeled samples.

Since incorrectly assigned labels will occur in this stage, the method also relies on a multiview voting concept. The intuition is that, if multiple runs for the same query in several feature representations agree on a subset of samples, then those belong to the query class with high confidence. If irrelevant samples are retrieved in one feature representation they will not be relevant in all perspectives and can be filtered.

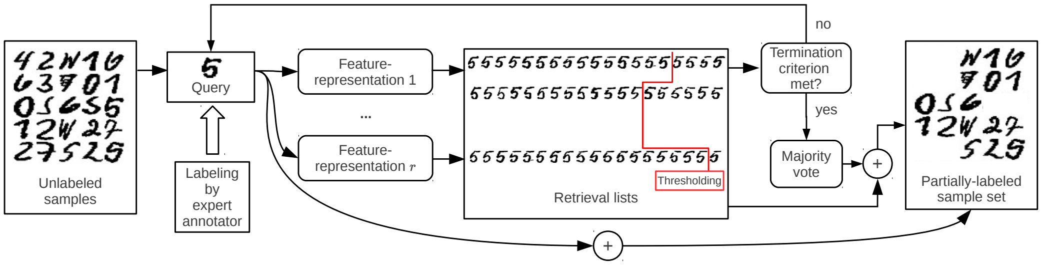
Unlabeled samples
Query
Labeling by expert annotator
Feature-representation 1
...
Feature-representation r
Retrieval lists
Thresholding
Termination criterion met?
no
yes
Majority vote
+
+
Partially-labeled sample set

Fig. 2. Illustration of the Retrieval-based labeling approach. A query image is presented to an expert annotator. Based on different feature representations similar characters are retrieved and the lists are thresholded. The query itself and the characters that appear in the thresholded lists of all representations are added to the labeled samples. The procedure is continued until a termination criterion is met and the remaining characters are added or discarded based on majority voting, which results in a partially-labeled sample-set.

The method starts with a pool of unlabeled data samples. As for CBL, r different feature representations are calculated. One sample is selected randomly from the pool and visualized for the annotator, who then decides on the class this sample represents. If the sample is heavily degraded or not recognizable the annotator can also manually reject it. The annotator's decision is treated as trustworthy and the sample is hence removed from the pool of unlabeled samples and added to the initially empty pool of labeled samples – or taken out of the sample pool altogether in the case of a rejection.

In either case, r retrieval runs are carried out on the remaining data samples, one for each feature representation, using the previously selected sample as query. This results in r retrieval lists, sorted ascendingly according to the samples' distances to the query. The choice of a distance measure that is used during retrieval can be arbitrary and may be chosen differently for each feature representation. It is advisable to select a distance measure that is normalized to facilitate the usage of a fixed threshold. For our experiments, we used the cosine distance.

All r sorted retrieval lists are truncated at a given distance threshold κ_d. Samples with a larger distance are not considered further and put back into the sample pool. For the remaining samples whose distance to the query is smaller than κ_d, the following voting rule is applied: If the respective sample is present in all r truncated lists, it is treated as belonging to the same class as the query with high confidence and assigned the query's label. This is analogous to the unanimity voting step in CBL described above. These samples are added to the pool of trusted labeled samples and removed from the pool of unlabeled samples. Similarly, if the query sample had been rejected by the annotator, all samples passing this criterion are also rejected and taken out of the process.

The remaining samples in the truncated lists are assigned a *soft vote* for the query class. Assume that a sample $\mathbf{X}_p$ is present in N_p of the r thresholded lists. Then, the confidence $\gamma_p(\omega_s)$ of $\mathbf{X}_p$ belonging to the query class ω_s is given by:

$$\gamma_p(\omega_s) = \frac{N_p}{r}. \tag{2}$$

Afterwards, the samples are put back into the pool of unlabeled samples and a new query sample is selected.

For all selection steps but the first, a steered sample selection strategy should be applied in order to systematically explore the available data. Here, we use a very simple heuristic that nevertheless proved effective during our experiments. We keep track of the number of times each sample has been considered in the voting phase by associating a counter with each sample. The counter is increased whenever the respective sample receives a (soft) vote. We then find all samples in the unlabeled pool whose counters are equal to the minimum value and randomly draw the query sample from those. Effectively, this results in an exploration of the data set and some balancing of votes because the selection rule will always select a sample that had a large distance to all previous queries in all feature representations. Thus, the selected sample comes from a volume of the feature space not (often) considered previously and therefore is likely to belong to an unseen or rarely seen class.

The process of sample selection, manual labeling and voting amongst truncated retrieval lists is iterated. However, iterating until all samples are either labeled or rejected is not suitable because the pool of unlabeled samples tends to decrease quickly only at the beginning. The "good" samples whose class membership can be determined easily are typically indentified and removed from the pool in early iterations. As the method proceeds, the pool increasingly consists of samples that are either outliers, considerably degraded or otherwise "difficult" to assess. The multiview voting approach offers no advantages in this case and is better aborted by a termination criterion.

Again favoring a simple and effective solution, we use a predefined maximum number of iterations. A more sophisticated but also potentially very time-consuming approach could, e.g., consist of re-training a classifier after each labeling iteration, then analyzing the class regions or decision boundaries of the classifier and aborting when no significant changes occur any more.

At the end of the procedure there exists a pool of samples with trusted labels that can be used directly for classifier training. Furthermore, the available set of class labels $\Omega = \{\omega_k, k = 1...c\}$ has evolved implicitly based on the labels assigned by the annotator during the process. However, a significant amount of samples will never pass the unanimity voting criterion in the selection phase and, therefore, remain in the unlabeled pool. Most of those will have shown up in some of the truncated lists in several iterations, and therefore will have received a number of soft votes for class labels. Thus, the final step of RBL consists of processing the unlabeled sample pool again and identifying samples whose accumulated votes indicate a class membership with sufficiently high confidence.

For each sample $\mathbf{X}_i$ in the unlabeled pool, the accumulated normalized class confidences $\tilde{\sigma}_i(\omega_k)$ are calculated from the soft votes as follows:

$$\tilde{\sigma}_i(\omega_k) = \frac{r}{(r-1) \cdot n_i} \cdot \sum_t \sum_k \gamma_{i,t}(\omega_k) \delta(u_t - \omega_k). \tag{3}$$

In the above equation, r is the number of feature representations, n_i is the counter associated with sample $\mathbf{X}_i$ keeping track of how often the sample was considered, $\gamma_{i,t}(\omega_k)$ are the individual soft votes from eq. 2 assigned in the t-th iteration, and u_t is the query label of this iteration. The normalization factor $\frac{(r-1)\cdot n_i}{r}$ constitutes the maximum possible accumulated confidence for any class. The sample was considered n_i times, and each time can receive a maximum soft vote value of $\frac{r-1}{r}$. Thus the accumulated class confidence is normalized to $[0, 1]$. Consequently, even if n_i may be quite different for every sample, $\tilde{\sigma}_i(\omega_k)$ constitutes a normalized and comparable measure of class confidence.

Finally, the final class label y_i of $\mathbf{X}_i$ is determined as the one having the maximum accumulated confidence:

$$y_i = \underset{k}{\operatorname{argmax}}(\tilde{\sigma}_i(\omega_k)). \tag{4}$$

If the associated maximum confidence $\tilde{\sigma}_i(y_i)$ is above a threshold κ_s, the sample is added to the list of labeled training samples. Otherwise, the sample is rejected because the assigned label would be too unreliable.

Compared to CBL presented in the previous section, the above procedure offers several advantages. No prior knowledge or assumption about the number of classes is required because they will evolve implicitly based on the labels assigned by the annotator. Also, it is possible to manually reject ”bad” samples, and the required manual effort does not depend on the number of different representations r since they are evaluated simultaneously. On the other hand, the impact of errors in the manual annotation can be expected to be higher. Also, the computational load of the retrieval step depends on both the size of the sample pool and the number of setups r, and can get very high for large data sets. This is critical because the retrieval step is integrated in an interactive feedback loop. Consequently, for large real-world problems, the selection of a small number of compact feature representations is advisable, limiting the time required for distance calculation. Additionally, fast approximate search methods (cf. e.g. [16]) could be utilized to keep the latency low.

As discussed in the previous section CBL requires parameters for the number of clusters and the voting. RBL requires three heuristical parameters, namely the sample selection thresholds κ_d, κ_s and the termination criterion (here, the maximum number of manual labeling iterations I_m).

4 The DWD-Dataset: A real world application

The proposed semi-supervised methods were developed in the course of a project aiming at the automatic transcription and subsequent statistical analysis of official historical weather reports. These reports have been collected from numerous observatories around the world between 1877 and 1999 and consist of pre-printed tabular forms manually filled in by an observer (cf. Fig. 4a). The documents used in the following experiments were kindly provided by the German Weather Service (”Deutscher Wetter Dienst”, DWD). Our dataset consists of 102 documents [1] that were scanned at approximately 200dpi. The complete weather report collection of the DWD consists of several 10,000 pages, but is currently not digitized yet. Accessing and automatically analyzing the information contained in all these documents would provide meaningful insights into long-term weather fluctuations and development over the last 150 years. Because of the pre-printed table structures additional knowledge about the possible occurences (characters or digits) in each cell can be inferred. This knowledge will be used for limiting the set of possible class candidates in the following.

In the context of this paper, this data is interesting for a variety of reasons. Firstly, the weather entries consist of handwritten characters and digits that occur either isolated or in short strings from a very restricted vocabulary. Therefore, a character-level analysis is feasible and sufficient for a complete automatic transcription of all the relevant information. Secondly, although the collection contains documents from a number of different writers, it consists of large clusters of documents written by the same writers. This is because the observers typically were employed fors several years. Consequently, the data is promising for semi-supervised methods because they rely on propagating labels from known samples to large clusters of similar unlabeled samples, which

[1] The complete DWD-dataset containing the 102 scanned pages, extracted handwritten characters and annotations is available for scientific research at http://patrec.cs.tu-dortmund.de/cms/en/home/Resources/

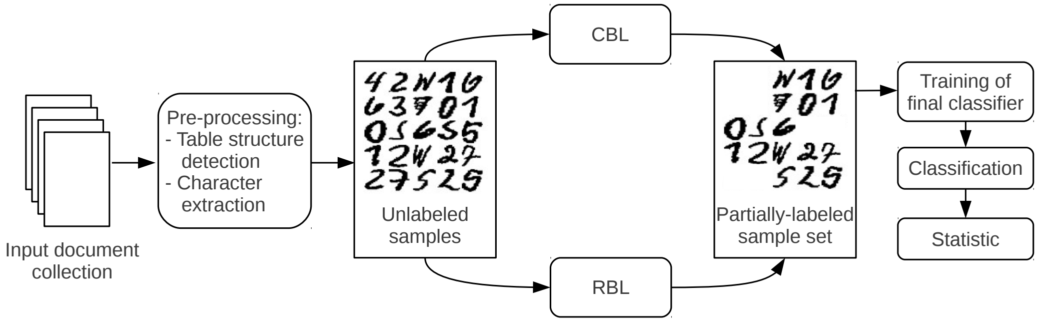
CBL
Pre-processing:
- Table structure detection
- Character extraction
Unlabeled samples
Partially-labeled sample set
Training of final classifier
Classification
Statistic
Input document collection
RBL

Fig. 3. Overview of the processing steps required for the DWD dataset. First the table structure is detected and the characters are extracted from the document collection. Then either CBL or RBL is applied. The partially-labeled sample set is used for training a classifier. The recognition results can be used for a statistical analysis of weather observations.

can be safely assumed to exist given the data characteristics. Finally, since the extracted weather information is to be analyzed statistically afterwards, a certain transcription error rate is tolerable. An unsupervised propagation of labels is always error-prone, and each label set acquired by a semi-supervised method will contain a certain number of wrongly assigned labels. These will have an impact on the final classification accuracy. Consequently, this approach is not the best choice for applications in which a very high recognition accuracy is important. However, it is very well suited for mass-data analysis where a moderate loss of character-level accuracy is acceptable, but the manual workload associated with conventional supervised learning is not.

In the following, we provide an overview of the automatic analysis and transcription system that was developed for the DWD data set (cf. Fig. 3), explaining how the character and digit samples are acquired. Afterwards, we demonstrate the applicability of the proposed semi-supervised labeling schemes for this real world data in our experiments.

4.1 Automatic analysis of tabular documents

In order to extract isolated characters and digits from the tabular weather documents, the table layout has to be analyzed. Given the layout, table cells can be extracted and their handwritten contents can be further evaluated. Matching the table structure with a template allows for identifying relevant cells (e.g., separating pre-printed text from handwritten information) and inferring knowledge about the expected type of content (characters or digits). We provide a condensed overview of the analysis system developed for this purpose. More details are provided in [22].

The process starts with a binarization of the image by applying the Niblack method [17]. This is a locally adaptive method that determines a binarization threshold for each pixel based on gray level statistics of the pixel's surroundings. Next, the Hough parameter space (cf. [6]) is calculated from the binary document image. Line-like structures in the image correspond to local maxima in the Hough representation. Since the expected tabular structure can be quite complex (cf. Fig. 4a), a locally adaptive peak search is applied instead of a global threshold. Given a sliding window, only those local maxima are extracted that are above the mean and a weighted standard deviation of the surrounding pixels in the Hough representation. Additionally, non-maximum suppression is applied, keeping only peaks that are the maximum within their local neighborhood. This results in a list of line hypotheses that will typically contain some false positives due to long text lines and image noise. Most of them can be reliably discarded using a simple criterion: Assuming a rectangular tabular grid, the pairwise inclination angles between valid line hypotheses should be approximately 0 or $\frac{\pi}{2}$. Thus, for each line, a histogram of these angles is calculated. A line hypothesis is discarded if its maximum bin does not correspond to the expected values within a small tolerance threshold.

One reason for choosing the Hough Transform for table line extraction, as opposed to, e.g., profiles (cf. [18]), is that this procedure does not require an upright, rotation- or skew-corrected image. Thus, it is more robust against improper scanning. In-plane rotations of the document can be corrected at this point by analyzing the distribution of angle parameters of the extracted lines.

Next, the line segments that correspond to actual lines in the document have to be determined. First, all pairwise intersection points between the remaining line hypotheses are calculated. These are then merged and arranged in a rectangular axis-parallel grid by applying Mean Shift clustering [4] separately to the x and y coordinates of extracted points. Note that Mean Shift clustering does not require the number of clusters to be known, so no assumption about the document is made here. The cluster centers are then snapped to the local image structure as follows: a subimage is extracted around each center's position, the size of which determines the maximum snapping distance. The horizontal and vertical projection profiles of this subimage are calculated by summing up foreground pixels along the image axes and weighting them with a Gaussian function that penalizes large displacements. Elongated foreground structures parallel to the summation direction will generate peaks in the profiles. The locations of the maximum peaks in the weighted profiles give the snapping position. This procedure thus fits the grid nodes to local distortions of the table structure (Fig. 4c).

For each line segment connecting neighboring nodes, a lineness score s_L is then computed as follows: The subwindow defined by the extremal coordinates of adjacent nodes is extracted. Then, a projection profile is calculated within

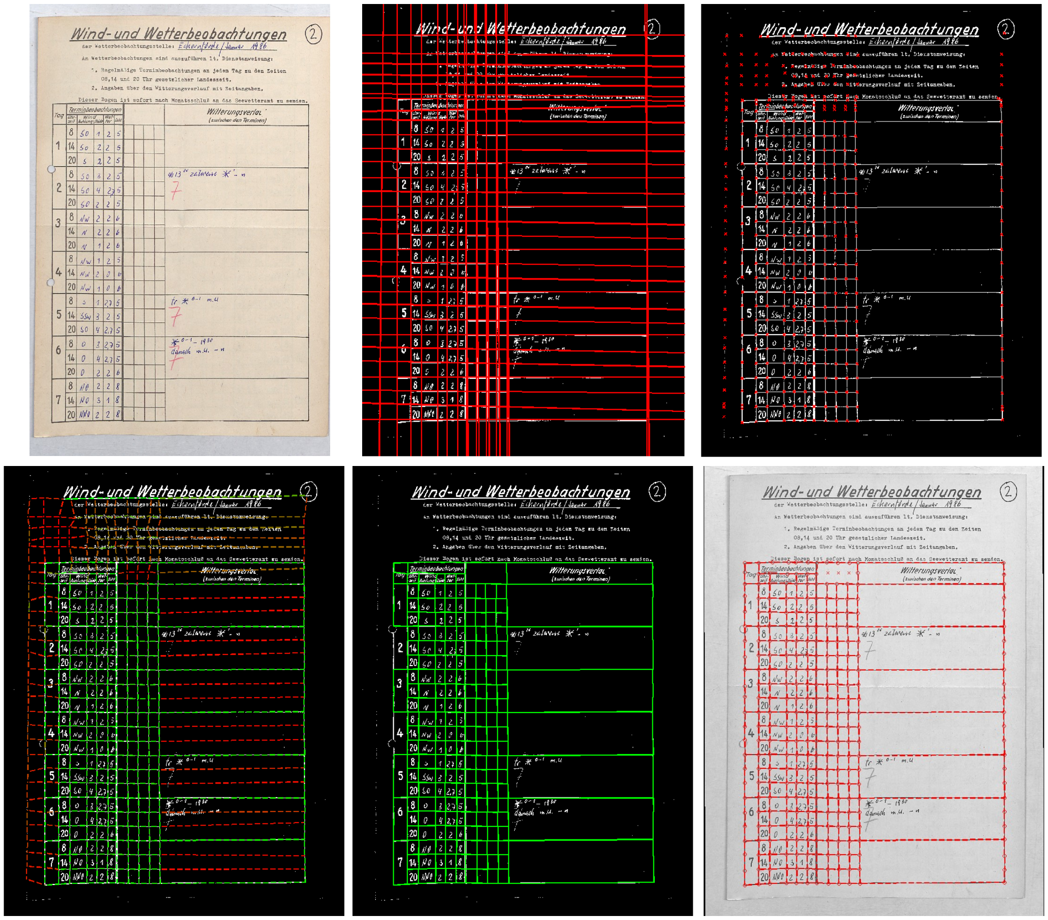

Fig. 4. Example of the table extraction. From top left to bottom right: a) Original image. b) Line extraction on the binarized image. c) Pair-wise line intersection points after clustering and snapping to local image distortions. d) Lineness scores (green: High lineness, red: Low lineness). e) Grid structure after thresholding the lineness scores. f) Fitted table structure. This graphic is best viewed in color.

this window in the direction perpendicular to the line orientation. For an ideal line with no noise and distortions, the profile should be perfectly flat. The less line-like the structure contained in the subwindow is, the more the profile will deviate from this ideal form. Consequently, the normalized profile is treated as a discrete probability distribution $\mathbf{p} = (p_i, i = 1...l)$. In case of a line this distribution should be uniform. Hence, the Bhattacharyya distance [5], a normalized and symmetric distance measure for discrete distributions is used for comparing the profile $\mathbf{p}$ with a uniform distribution $\mathbf{q}$:

$$s_L(\mathbf{p}, \mathbf{q}) = 1 - \sqrt{1 - \sum_i \sqrt{p_i \cdot q_i}}. \tag{5}$$

The lineness score $s_L(\mathbf{p}, \mathbf{q})$ is normalized to the range $[0, 1]$ and yields a measure for how closely the underlying image structure resembles a line (Fig. 4d). False positive line segments are rejected by calculating an adaptive threshold

Table 1
Overview of the evaluation datasets.

Dataset	Character classes	#Samples	Training samples	Test samples
MNIST	10	70,000	60,000	10,000
DWD	17	12,840	∼8,560	∼4,280

on the lineness scores of a document using Otsu's method [19]. Additionally, assuming a closed table structure, isolated lines are removed, yielding the final table hypothesis (Fig. 4e).

However, this structure will still exhibit errors, such as false positive and missing line segments. Therefore, a template database has been created which is queried for finding the best match based on profiles, similar to [18]. Given an extracted grid and a template, the task is to find the translation τ and scale ρ optimizing their alignment. The objective function is given by $f(\tau, \rho) = d(\mathcal{P}_E, \mathcal{P}_T(\tau, \rho))$, where d is a suitable distance function (e.g., Euclidean distance), $\mathcal{P}_E$ a horizontal or vertical profile from the extracted table structure and $\mathcal{P}_T(\tau, \rho)$ the corresponding transformed profile of the template. This objective function has a large number of very sharp local minima because of the regular grid structure and the binary nature of line alignment. To facilitate minimization, the profiles are smoothed with a Gaussian. A number of good starting hypotheses for ρ are determined based on statistics of table cell sizes, and τ is initially selected by aligning dominant profile peaks. The objective function is then minimized using a Simplex algorithm. This procedure is carried out for horizontal and vertical profiles independently, and the best template is selected by minimizing the combined score.

After applying the optimal transformation, the template is finally fitted to the image using the snapping algorithm described above. Single cells containing textual information can then easily be extracted from the fitted table.

5 Experiments

We evaluated our approach on two datasets. First we derived suitable parameters on the MNIST dataset [12] that contains samples for handwritten digits. Then we used the set of historical weather reports in order to evaluate our methods on a realistic task.

An overview of the two evaluation databases is given in Table 1. The MNIST dataset consists of 28×28 pixel images of handwritten digits. The dataset contains 60,000 examples for training and a test set of 10,000 samples. The

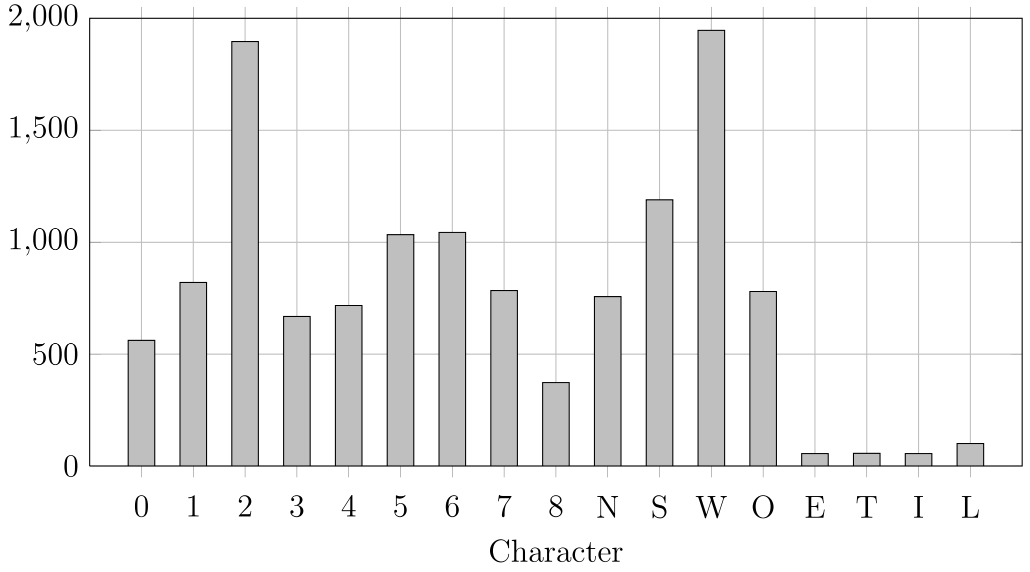

Fig. 5. Distribution of the handwritten digits and characters in the DWD dataset. In total there are 12,840 annotated samples in the dataset.

102 documents of the DWD dataset (see section 4) contain in total 12,840 characters and digits in 17 different classes, collected by extracting connected components from the table cells interiors. The connected component images were also resized to 28 × 28 pixels prior to feature calculation. Ground truth labels are available for all characters. The distribution of the characters and digits in the database is shown in Figure 5. All documents are subdivided in a 3-fold cross-validation setup. In each validation set, approximately 2/3 of the documents constitute the training set and the remaining the test set. Thus, training and test set are disjoint, but overall, all documents are considered once for testing.

5.1 Table extraction on the DWD data

As described in section 4.1 the DWD dataset requires a preprocessing step that analyzes the tabular structure of the documents. This allows to infer further knowledge about the content of the cells to be either characters or digits. Hence, the set of candidate characters can be limited for each cell.

The results in Table 2 show that the proposed algorithm solves this task. 98% of the lines were identified correctly with a small false positive rate of 2.5%. However, most errors were caused by rectangles enclosing the document that was matched and are therefore not relevant for the specific cell content in the recognition step. The template matching step yielded the correct template for the respective document in all cases.

Table 2
Results of the table extraction approach on the DWD dataset. The percentage of false positives is calculated with respect to the total number of detections.

	Total Number	Correct	Missing	False Positive
Nodes	19,805	19,417 (98.0%)	388 (2.0%)	492 (2.5%)
Lines	34,418	33,748 (98.1%)	670 (1.9%)	868 (2.5%)

5.2 Setups for the semi-supervised labeling

Both approaches described for labeling data in a semi-supervised manner rely on a multiview approach of different feature representations. Furthermore, CBL applies different clustering algorithms to the data.

In the following experiments the character images were normalized to 28 × 28 pixels. We evaluated the following feature representations: The original character images as observed in the database (RAW), normalized to 28 × 28 pixels, a PCA of the RAW data using the first 80 components, as well as structural features based on contour chain codes (CC), skeletons (SKEL) and character reservoirs (RES) [9,20]. The reservoirs were modified by considering 5 types of reservoirs (top, bottom, left, righ, loop) and using a soft assignment of the positions of their centers of gravity to image cells. All features except RAW and PCA were calculated on a 4 by 4 cell subdivision of the character images by concatenating the individual cells' representations.

For the Clustering-based annotation we considered three different clustering algorithms. Namely, the generalized Lloyd algorithm [14], Self Organizing Maps (SOM) [10] and Growing Neural Gas (GNG) [8].

In a baseline experiment using the complete available ground truth the features were evaluated in combination with three different classifiers. A k-Nearest Neighbor approach, a Multi Layer Perceptron (MLP) and a Support Vector Machine (SVM). In the semi-supervised setups the labeling was simulated by assigning the correct ground truth label, assuming an error-free annotation.

5.3 Baseline experiments

On both databases we evaluated a baseline experiment using the complete ground truth that is available. Note that for the MNIST database 60,000 samples are available, while the DWD dataset has only approximately 8,560 samples for 17 different digit classes. We evaluated all combinations of the proposed classifiers and feature representations.

Table 3
Overview of handwritten character recognition results (in %) using a 3 Nearest-Neighbor classifier. Confidence intervals are for a significance level of 95%.

Dataset	Method	#Labels	RAW	PCA	CC	SKEL	RES
MNIST	Ground truth	60, 000	96.55±0.38	**97.54±0.32**	95.39±0.43	87.34±0.67	86.14±0.69
	CBL	162	90.88±0.58	**91.57±0.56**	90.49±0.59	85.24±0.71	85.10±0.71
	RBL	162	86.58±0.68	**87.33±0.67**	86.50±0.68	80.84±0.78	78.08±0.82
DWD	Ground truth	8,560	95.11±0.39	95.23±0.38	**95.63±0.37**	92.59±0.47	89.88±0.53
	CBL	162	92.52±0.47	92.45±0.47	**92.96±0.46**	90.68±0.56	88.07±0.57
	RBL	162	92.75±0.46	92.96±0.46	**93.30±0.45**	90.09±0.53	88.04±0.57

Table 3 shows the results of the 3 Nearest-Neighbor classifier, using the Euclidean distance. We also considered different neighboorhood sizes and distance metrics such as the cosine distance. However, informal experiments showed that this parameterization of the Nearest-Neighbor classifier works best for the character recognition tasks considered here.

Table 4 and 5 show the results of the SVM and the MLP. In comparison with the Nearest-Neighbor approach the recognition rates of these methods are significantly lower. The reason for this might be the extensive amount of available ground truth samples. Hence, we considered all three classifiers for the semi-supervised approaches since the number of labeled samples will be smaller and more noisy.

With respect to the different feature representations, reducing the dimension using PCA, as well as the lower dimensional chain code and skeleton features improve the classification rate in comparison with the raw image pixels. In contrast, the character reservoirs perform poorly.

Cluster-based labeling

Even though it would be possible to use all features for better discrimination of the data, for efficiency reasons it is desirable to use a subset only. With the CBL method the the expert annotator has to manually label the clusters of each setup. Therefore, choosing too many different setups would counteract the overall goal of reducing the manual annotation effort.

Hence, we aimed for a small number of clusters. In [22] we demonstrated that the impact of the number of clusters is negligible if increased beyond a certain point. Therefore, we used 54 cluster centers for each feature representation and a combination of three different setups which requires 162 (3 × 54) labeling operations. An exhaustive search was performed in order to determine the best possible combinations of features and clusterings. All different features extracted from the MNIST training material were clustered using the generalized Lloyd algorithm, SOM and GNG. The resulting cluster centers were then annotated, inferring the labels of the samples from the cluster centers.

Table 4
Overview of handwritten character recognition results (in %) using a SVM. Confidence intervals are for a significance level of 95%.

Dataset	Method	#Labels	RAW	PCA	CC	SKEL	RES
MNIST	Ground truth	60, 000	92.15±0.54	92.60±0.53	**95.30±0.43**	85.69±0.70	82.32±0.76
	CBL	162	88.13±0.65	88.64±0.64	**91.28±0.57**	83.83±0.73	81.58±0.77
	RBL	162	84.97±0.71	86.76±0.68	**89.50±0.62**	81.91±0.77	78.36±0.82
DWD	Ground truth	8,560	91.39±0.50	93.44±0.44	**95.33±0.38**	91.88±0.48	88.03±0.57
	CBL	162	89.57±0.54	91.44±0.50	**92.98±0.42**	90.42±0.52	86.92±0.59
	RBL	162	87.86±0.58	92.18±0.48	**94.45±0.41**	90.93±0.51	88.04±0.57

As quality criteria, the sample recall $\mathcal{R}$ (percentage of retained trusted samples after voting) and label precision $\mathcal{P}$ (percentage of correctly labeled retained samples) obtained on MNIST were used. Ranking the different combinations, the best setup was: RAW/GNG, CC/GNG, and CC/k-means. Unanimity vote occured in $\mathcal{R} = 76.15\%$ of the cases with a precision $\mathcal{P} = 96.10\%$. Thus, 45,690 annotations were inferred using only 162 manual labeling operations, corresponding to a relative manual effort of 0.35%.

In general, GNG and Lloyd outperform SOM clustering. The improvments by using GNG in our experiments confirm the observations that were, for example, reported in [2]. Compared to our results in [32], the sample recall increased substantially (approx. 21%) while the same labeling accuracy was obtained. This shows the benefit of incorporating not only different feature representations, but also different clustering methods. The multi-view approach allows to have a better, more diverse view on the data.

Retrieval-based labeling

In order to find suitable values for the parameters κ_d and κ_v for the retrieval based approach, a number of experiments was conducted on the MNIST data set. $I_m = 500$ labeling operations were performed and averaged over 10 runs with identical parametrization in order to smooth the effects of the random selection. The goal was to find a range of parameters offering a good balance between sample recall and label precision.

In accordance with the CBL we also considered 162 labeling operations and a combination of three different setups. Numerous combinations of different features were investigated, showing the best results for the combination of CC + PCA + RES.

The labeling precision is generally high, except for small values of κ_v. It also degrades for small values of κ_d, because then only few samples will be considered in each retrieval run, and the small overall number of votes leads to an unreliable majority decision. In terms of sample recall, the method is more restrictive than CBL retaining large fractions of the data only for small values

Table 5
Overview of handwritten character recognition results (in %) using a MLP. Confidence intervals are for a significance level of 95%.

Dataset	Method	#Labels	RAW	PCA	CC	SKEL	RES
MNIST	Ground truth	60, 000	**97.50±0.32**	91.60±0.56	95.94±0.40	86.43±0.69	84.30±0.73
	CBL	162	88.92±0.63	86.13±0.69	**89.46±0.62**	81.98±0.77	81.52±0.77
	RBL	162	82.54±0.76	82.03±0.76	**87.23±0.67**	80.11±0.79	76.85±0.83
DWD	Ground truth	8,560	94.56±0.41	91.99±0.48	**95.33±0.38**	90.88±0.48	88.03±0.57
	CBL	162	90.74±0.51	89.95±0.53	**92.45±0.47**	88.42±0.57	85.38±0.62
	RBL	162	91.29±0.50	91.12±0.50	**94.07±0.42**	88.47±0.51	85.04±0.60

of κ_v. While CBL enforces exactly the same number of votes for all samples it varies in RBL. Consequently, samples at the boundary of class distributions may get very few or inconsistent votes and thus are rejected.

In order to determine a suitable parametrization, we calculate the F3 score $\mathcal{F}_3 = \frac{10\mathcal{R}\mathcal{P}}{9\mathcal{R}+\mathcal{P}}$, reflecting the assumption that it is more desirable to have correctly labeled samples than retaining large portions of the original data. The maximum score was $\mathcal{F}_3 = 91.54\%$ ($\mathcal{R} = 82.72\%$, $\mathcal{P} = 92.63\%$) for parameter values $\kappa_d = 0.25, \kappa_v = 0.20$. However, the method is not too sensitive against the concrete choice of values (cf. [21] for details). In the following, again favoring high precision, we will use more restrictive parameter values of $\kappa_d = 0.2, \kappa_v = 0.30$, yielding $\mathcal{P} = 97.15\%$, $\mathcal{R} = 59.02\%$, $\mathcal{F}_3 = 91.26\%$ for the above experiment.

5.4 Recognition experiments

The CBL and RBL setups derived in the previous sections are evaluated in a character recognition experiment on the MNIST and DWD database. Our goal is to show that both methods perform reliably on different character datasets and that tuning the parameters to specific problems is not necessary. The later one is very important, since it will not be possible in real applications.

We use the partially labeled training sets that are created by CBL or RBL in order to train the classifier and evaluate it on independent test sets. Since in the proposed methods only a fraction of the original training data is labeled, the performance of the baseline experiment using the complete ground truth can be seen as an upper limit on the accuracy that can be achieved.

Applying the proposed labeling schemes to the MNIST data results in a substantial loss in recognition rates. The reason is that both methods rely on discovering clusters of similar samples, i.e., from the same writer or written in the same style. Since the MNIST data is very diverse and contains hundreds of writers this assumption is violated resulting in a loss of accuracy. RBL performs worse than CBL because propagating the labels based on the retrieval lists proceeds considerably slower than labeling the large portions of data contained within a cluster. Our experiments showed that the performance of RBL keeps increasing until approximately 400–500 manual annotations were performed (still a relative effort of less than 1% of the 60,000 annotations). The saturated recognition score is then comparable to CBL. Nevertheless, we keep the number of manual operations equal in both methods to get comparable results (for further details refer to [21]).

However, for the DWD data, which is much more homogeneous in terms of writing style, the results obtained with RBL are close to or better than CBL. As discussed in [22], a drawback of CBL is that it tends to discard rare classes in the case of highly unbalanced data since they do not form individual clusters and thus are eliminated by the unanimity voting. While this did not occur for the balanced MNIST set, only 13 out of 17 classes were recovered on average on the DWD data. With RBL, all 17 classes were retained. This shows a major advantage of the RBL approach: Because of the steered sample selection, under-represented classes are less likely to be discarded.

In addition, both CBL and RBL are close to the baseline experiment. This clearly shows the potential of semi-supervised methods, provided that large portions of the data show similar characteristics. Hence, the methods are especially promising for large single-writer collections, as, for example, in historical archives. Only very little manual effort was required. Performing the 162 labeling operations manually was a matter of a few minutes, which is an impressive result for robustly recognizing the characters in about 100 documents.

Similar results have been achieved for the annotation of Lampung characters [32], an Indonesian script. Using the CBL method only 0.48 % of the characters had been labeled by the manual annotator. The remaining samples inherited the labels correctly after the majority voting of three views: Raw, PCA and an encoder network. A recognition rate of 86.2% for 11 different lampung character classes could be achieved using a partially-labeled sampleset.

Concerning the different classifiers it can be observed that for the more noisy samples obtained by the semi-supervised labeling the SVM and MLP perform much better in relation to Nearest-Neighbor than in the supervised case. The reason for this is probably the influence of incorrectly labeled samples on the Nearest-Neighbor approach. Even a single incorrectly labeled sample could influence the prediction of several neighbors in the test set. Hence, for CBL classification rates of the SVM and Nearest-Neighbor approach are not significantly different on both datasets. Furthermore, for RBL the best results are achieved using the SVM.

6 Conclusion

In this paper we discussed the advantages of semi-supervised labeling strategies involving minimal human effort, proposing two different methods to label and later recognize characters in large historical archive datasets. Both approaches are based on multi-view learning, incorporating different feature representations, and a voting procedure for reliably propagating labels assigned by an expert annotator.

Our experiments showed that even with a labeling effort of far less than 1% reliable results can be achieved. Several thousand labels were inferred by only 162 manual labeling operations, reducing the effort for training a recognizer to several minutes. We especially demonstrated the effectiveness on large scale single-writer databases. In this case the recognition rates are remarkably close to the upper limit defined by a baseline experiment that uses the complete ground truth. Additionally, we presented a realistic application that is highly dependent on reducing the human effort in order to make a huge collection of historical weather reports available for analysis.

[1] G. R. Ball, S. N. Srihari, Semi-supervised learning for handwriting recognition, in: Proc. Int. Conf. on Document Analysis and Recognition, 2009.

[2] F. Camastra, A. Vinciarelli, Combining neural gas and learning vector quantization for cursive character recognition, Neurocomputing 51 (2003) 147–159.

[3] E. Chang, G. Sychay, K. Goh, G. Wu, CBSA: Content-based soft annotation for multimodal image retrieval using Bayes point machines, IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 26–38.

[4] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Trans. on Pattern Recognition and Machine Intelligence 24 (5) (2002) 603–619.

[5] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. on Pattern Recognition and Machine Intelligence 25 (2) (2003) 564–575.

[6] R. O. Duda, P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Comm. of the ACM 15 (1972) 11–15.

[7] B. J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 972–976.

[8] B. Fritzke, A growing neural gas network learns topologies., in: G. Tesauro, D. S. Touretzky, T. K. Leen (eds.), Neural Information Processing Systems, MIT Press, 1994.

[9] A. Junaidi, S. Vajda, G. A. Fink, Lampung - a new handwritten character benchmark: Database, labeling and recognition, in: Int. Workshop on Multilingual OCR, ACM, 2011.

[10] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78 (9) (1990) 1464–1480.

[11] L. I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.

[12] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Intelligent Signal Processing, IEEE Press, 2001.

[13] J. Li, H. Mouchère, C. Viard-Gaudin, An annotation assistance system using an unsupervised codebook composed of handwritten graphical multi-stroke symbols, Pattern Recognition Letters (2012) on–line.

[14] S. Lloyd, Least squares quantization in PCM, Information Theory, IEEE Transactions on 28 (2) (1982) 129–137.

[15] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: L. M. L. Cam, J. Neyman (eds.), Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967.

[16] M. Muja, D. G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in: In VISAPP International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, 2009.

[17] W. Niblack, An Introduction to Digital Image Processing, Prentice Hall, 1986.

[18] H. Nielson, W. Barrett, Consensus-based table form recognition of low-quality historical documents, Int. Journal on Document Analysis and Recognition 8 (2) (2006) 183–200.

[19] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. on Systems, Man and Cybernetics 9 (1) (1979) 62–66.

[20] U. Pal, S. Kundu, Y. Ali, H. Islam, N. Tripathy, Recognition of unconstrained Malayalam handwritten numeral, in: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'04), 2004.

[21] J. Richarz, S. Vajda, G. A. Fink, Annotating handwritten characters with minimal human involvement in a semi-supervised learning strategy, in: Proc. Int. Conf. on Frontiers in Handwriting Recognition, Bari, Italy, 2012.

[22] J. Richarz, S. Vajda, G. A. Fink, Towards semi-supervised transcription of handwritten historical weather reports, in: Proc. IAPR Int. Workshop on Document Analysis Systems, 2012.

[23] L. Rokach, Pattern Classification Using Ensemble Methods, World Scientific Publishing Company Inc., 2010.

[24] J. Sas, U. Markowska-Kaczmar, Semi-automatic training sets acquisition for handwriting recognition, in: Proc. Int. Conf. on Computer Analysis of Images and Patterns, LNCS 4673, Springer, 2007.

[25] J. Schumacher, D. Sakic, A. Grumpe, G. A. Fink, C. Wöhler, Active learning of ensemble classifiers for gesture recognition, in: Pattern Recognition: 34th DAGM-Symposium Graz, 2012, to appear.

[26] B. Settles, Active learning literature survey, Computer Sciences Technical Report 1648, University of Wisconsin-Madison (2009).

[27] B. Settles, Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances, in: Proc. Conf. on Empirical Methods in Natural Language Processing, 2011.

[28] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: Proc. ACM Int. Conf. on Multimedia, 2001.

[29] A. H. Toselli, V. Romero, M. Pastor, E. Vidal, Multimodal interactive transcription of text images, Pattern Recognition 43 (5) (2010) 1814–1825.

[30] A. H. Toselli, V. Romero, E. Vidal, L. Rodriguez, Computer assisted transcription of handwritten text images, in: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on, vol. 2, IEEE, 2007.

[31] P. Treeratpituk, J. Callan, Automatically labeling hierarchical clusters, in: Proc. Int. Conf. on Digital Government Research, 2006.

[32] S. Vajda, A. Junaidi, G. A. Fink, A semi-supervised ensemble learning approach for character labeling with minimal human effort, in: Proc. Int. Conf. on Document Analysis and Recognition, 2011.

[33] Z.-H. Zhou, When semi-supervised learning meets ensemble learning, in: Proceedings of the 8th International Workshop on Multiple Classifier Systems, MCS '09, 2009.

[34] X. Zhu, A. B. Goldberg, Introduction to Semi-Supervised Learning, Morgan & Claypool, 2007.