A Semi-Supervised Ensemble Learning Approach for Character Labeling with Minimal Human Effort

Szilárd Vajda, Akmal Junaidi, Gernot A. Fink
Department of Computer Science
TU Dortmund
Dortmund, Germany
{*szilard.vajda,akmal.junaidi,gernot.fink*}@*udo.edu*

***Abstract*—One of the major issues in handwritten character recognition is the efficient creation of ground truth to train and test the different recognizers. The manual labeling of the data by a human expert is a tedious and costly procedure. In this paper we propose an efficient and low-cost semi-automatic labeling system for character datasets. First, the data is represented in different abstraction levels, which is clustered after in an unsupervised manner. The different clusters are labeled by the human experts and finally an unanimity voting is considered to decide if a label is accepted or not. The experimental results prove that labeling only less than 0.5% of the training data is sufficient to achieve 86.21% recognition rate for a brand new script (Lampung) and 94.81% for the MNIST benchmark dataset, considering only a K-nearest neighbor classifier for recognition.**

***Keywords*-semi-supervised character labeling; clustering, ensemble learning; Lampung characters;**

I. Motivation

During the last few years the focus in handwritten character recognition has shifted from Arabic digits [1], Chinese [2] and Kanji handwritten character recognition toward scripts like Farsi [3], Devnagari, Telegu, Oriya, Bengali [4], [5] etc.

Such a broad interest in these ancient scripts shows the endeavor of some countries to preserve these scripts as being a relevant part of their cultural heritage. Our interest is to help such initiatives by proposing to recognize an Indonesian script, the Lampung [6], [7]. In our best knowledge, there is little or no work available regarding this Indic related script.

The script is called “kaganga” which comes from the first 3 letters, ka, ga and nga respectively. Some districts in Sumatra Island, Indonesia, are having traditional scripts which became a remarkable trait of those areas. All those scripts were not genuine scripts of the native but originated from the ancient script in South India [6], [7]. The Lampung script is one of the scripts in Sumatra Island which was inherited from this ancient script. More precisely, it descended of Devnagari script [4], a cluster of Brahmi script [7] from South India.

Beside the Devnagari script as a core, the Arabic script structure [7] also influenced the Lampung script. The concept of developing a sound syllable using diacritics on the top and the bottom in Arabic writing system is adopted as well. Furthermore, the Lampung script has another concept by putting diacritics on the right side of a letter.

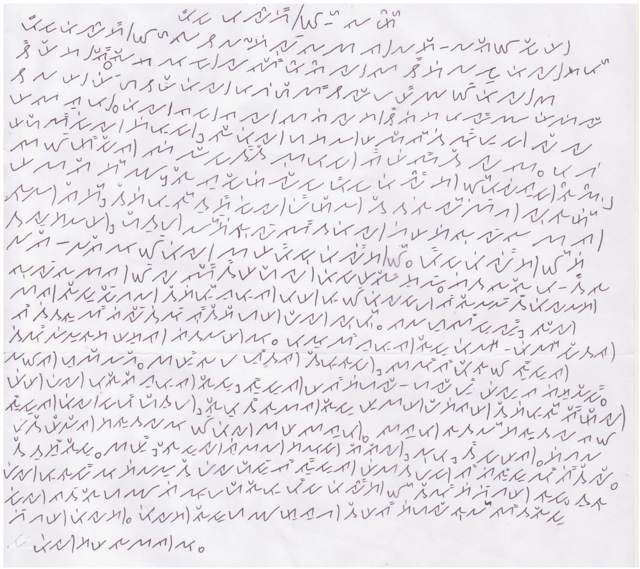

Figure 1: A Lampung document

The Lampung script is not a cursive writing system. It has 20 main letters with 11 diacritics putting on one of three possible positions around the letter and 6 punctuation marks. A Lampung document sample can be seen in Fig.1. Some Lampung characters are depicted in Fig.3.

Recognizing such an unknown script as Lampung is a real challenge as there is no labeled data or not even synthetic data available to train the different character recognizers.

The different database collecting initiatives [1], [4], [8] described in the literature over the years address this considering mainly manually labeled sets, involving a tremendous human effort which can just grow with the amount of data available. As stated by Stamatopoulos et al. [9] the efficient ground truth for document image processing should be a “quick and low cost” solution.

In order to reduce the human effort from the processing chain, we propose to label the character data automatically,

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.60

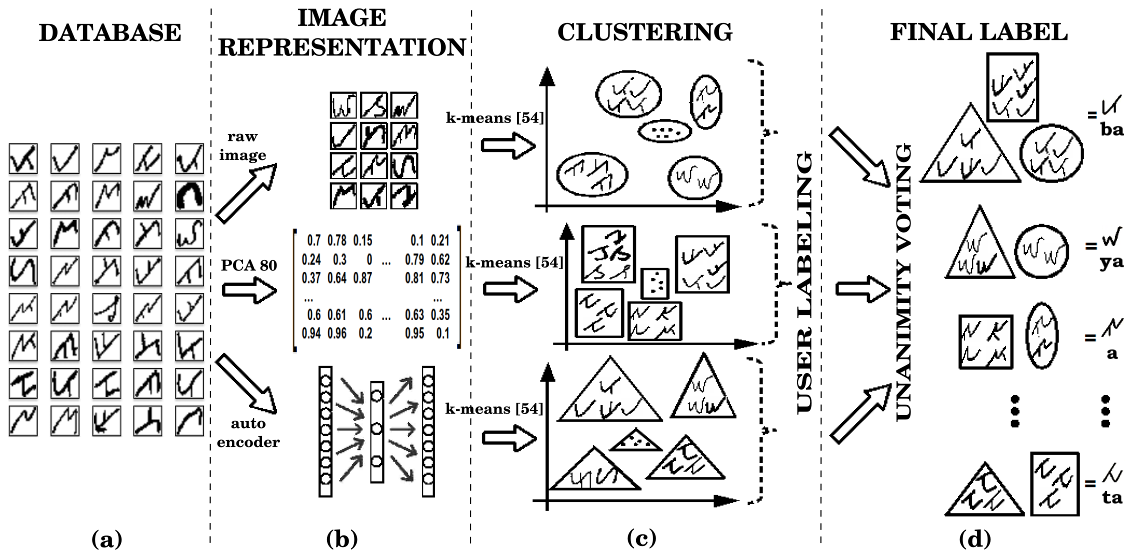

Figure 2: General overview of the proposed semi-automatic labeling procedure

considering the least possible human interaction, involving different complementary data representations, unsupervised clustering, minimal human knowledge and ensemble learning. Such semi-automatic labeling strategy can help to create easily new character datasets and provide the scientific community with new benchmark datasets.

The rest of the paper is organized as follows. Section II. describes in details the proposed labeling strategy. Next, Section III. presents a broad description of the datasets used in the experiment. Finally, a summary of the current paper can be found in Section IV.

II. Semi-Supervised Character Labeling

Accurate ground truth creation is mandatory to train and test the different machine learning solutions proposed in document analysis [9]. To produce accurate results a huge and accurately labeled dataset is necessary implying tremendous work load and costs. The goal of the semi-supervised character labeling is to produce such amount of data without involving tedious labeling processes (performed by humans) and achieve that goal with reasonable costs.

A. General Overview

The semi-automatic labeling system described hereinafter is a 3 stage process to produce labels for unknown character shapes. The overall process is depicted in Fig. 2. The first stage of the process involves 3 different data representations starting with the raw pixel image, going through some data reduction process by PCA (Principal Component Analysis) and ending up with another data reduction, the so-called autoencoder network proposed by Hinton et al. [10]. These different data representations are then clustered using unsupervised clustering and those clusters are labeled by the human expert. The cluster identifiers are derived from the label of the cluster centroid. Finally, in the last stage a voting scheme is implemented to decide for the final label. Only those samples are labeled where there is unanimity regarding the label choice.

The main distinction between other semi-supervised learning strategies [11] and our method lies in the fact that we do not classify based on the votes, but we assign labels only to the training data and the final classifier is built on top of the inferred labels.

B. Different data abstraction levels

In order to implement ensemble learning type voting mechanism the ideal is to find complementarity [11] between the data representations and classifiers involved in the scheme.

As we will use the same clustering strategy (details to be found in Subsection II-C) we focus our effort to consider different abstraction levels for our data representation (see Fig. 2b).

Our first choice is the raw binarized image, considering as input the different pixel values of the image. Even though this representation seems to be rather simple, it has been used with success for digit recognition [1], [5]. Such a representation is advised for small size images together with images centered around their gravity center and normalized with respect to size.

The second choice was the reduction of the original pixel data using PCA, an orthogonal linear transformation such that the greatest variance lies in the first components. This well known data reduction strategy allows to cope with the correlated pixel values from the original representation.

Finally, we considered a rather new data reduction strategy proposed by Hinton et al. [10], where the data reduction is optimized with respect to the reconstruction performances of the so-called autoencoder network. The idea behind is to train a multilayer neural network with a reduced size hidden layer to reconstruct the original input. For more details please refer to [10]. The authors claim that this reduction produces much better reconstructions than a PCA would do. For our purpose the output of this hidden layer was considered as the new data representation.

The PCA and the autoencoder are two different data strategies focusing on data reduction, hence a certain level of complementarity can be assumed. The more sophisticated data representations we consider the more orthogonal data representation could be derived, thus more complementarity can emerge from the data.

C. Unsupervised clustering and manual labeling

Once we have the different data representations described in details in Subsection II-B, we cluster in an unsupervised manner the different data. For this purpose the general Lloyd algorithm[1] was selected. The only parameter of this algorithm is the k defining the number of clusters into which the partition should separate the data samples. This parameter is regulating the human effort involved in this semi-automatic process. The bigger the k is the more clusters need to be labeled.

Once the unsupervised clustering is done, each sample will “inherit” the label of the cluster centroid in which it is partitioned (see Fig. 2c). The manual labeling effort is reduced to label the centroids of each cluster, exactly k images for each type of data representation.

In our scenario this will imply $3k$ labeling operations. For datasets consisting of several thousand of samples such a labeling ($k < 100$) can be considered as a negligible effort. There is no restriction to consider different cluster numbers for the different data representations. The more cluster we have the more fine results can be achieved.

The method itself does not exclude the usage of more complex/sophisticated clustering algorithms, however such thorough analysis of these algorithms is beyond the scope of the current paper.

D. Voting

The clustering and the labeling will allocate a specific label to each sample from our data. The goal of the voting scheme [11] is to decide for a final label for each data sample (see Fig. 2d).

Assume that the labels are given as a d-dimensional binary vectors $[l_{i,1}, \ldots, l_{i,d}]^T \in \{0,1\}^d$, $i = 1, \ldots C$, where $l_{i,j} = 1$ if classifier C_i labels a samples p in class ω_j and 0 otherwise.

The unanimity vote will result in an ensemble decision for the class ω_k if

$$\sum_{i=1}^{C} l_{i,k} = C. \tag{1}$$

For simple majority voting the condition would change to

$$\sum_{i=1}^{C} l_{i,k} \geq \left[\frac{C}{2}\right] + 1. \tag{2}$$

This voting scheme will allow to decide for the final label of the data. In our voting scenario, the unanimity vote (see Eq. 1) counts 3 similar votes, while for the simple majority votes it is sufficient to have 2 similar classifiers voting for the same label.

E. Recognition

The voting scheme (unanimity) will provide a label for some image samples from the dataset. Only these images will be further considered in our experiments, mentioned as training data. For recognition a K-nearest neighbor algorithm is considered. For each character pattern the closest training samples’ labels will be assigned.

Our primary goal was not to achieve the best scores as possible, but rather to show the great potential of the semi-automatic labeling. More powerful tools like neural networks [1], [5] would provide even better recognition scores.

III. Experiments

To prove the efficiency of the proposed labeling we considered the Lampung character dataset and the well known MNIST digit dataset [1].

A. Lampung character dataset

The Lampung dataset used in our experiments was derived from a data collection written by 82 high school students from Bandar Lampung, Indonesia. The Lampung texts are created as transcriptions of some fairy tales. The media to perform their handwriting was an A4 paper that was designed to provide a space for the Lampung handwriting with a small trailing part for filling the contributors identity. Every handwritten document was created by only one writer, hence producing a complex, multi-scriptor dataset. Each handwritten sheet was scanned at 300 dpi. Such a document can be seen in Fig.1.

Initially, the image documents were binarized using Niblack’s method with a local thresholding. The results of this binarization became the sources for producing connected components (CCs) that ultimately considered as the main representations of the Lampung characters. In order to discard the noise, the clutter and the different side effects coming from the binarization the extracted CCs were filtered based on size, area, aspect ratio, pixel density [12]. Finally, each CC image was linearly normalized into 20x20 pixel image.

From 82 image documents, the filtering step succeeded and generated $35,193$ CCs images in total. These CCs contain

[1] We use the name “Lloyd algorithm” to refer to k-means clustering.

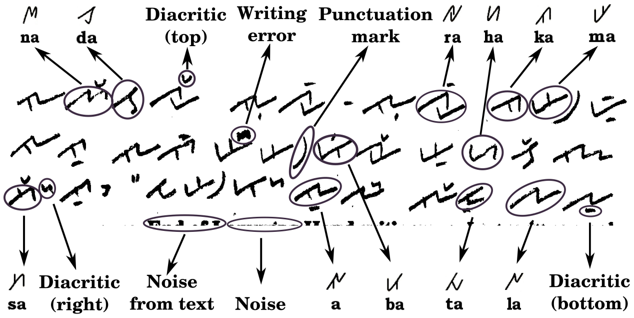

Figure 3: Some Lampung characters from a text paragraph

18 main characters (i.e. some labeled character samples can be seen in Fig.3), where the letters "ra" and "gha" are not to be found. Both letters have two elements, so that the CC extraction algorithm defined each element as a separated letter.

We separated from each available document the first 20 characters for test purpose, in total $1,640$ characters which were labeled manually. The remaining $33,553$ character samples were considered for training purpose without any label attached to them.

B. MNIST digit dataset

MNIST [1] is a well known benchmark dataset containing separated Arabic digits. The images coming mainly from census forms, are size normalized to 28x28 gray level images. The dataset contains $60,000$ and $10,000$ images for training and test, respectively. For our experiments, we used the training set but without any label information. The selection of this dataset was two fold: a) labels are available and b) we can directly compare our results with similar, state-of-the-art methods.

C. Results

For the raw image representation 20x20 and 28x28 size images were considered for Lampung and MNIST, respectively. In the PCA reduction process, the 80 most relevant principal components were used. This choice was motivated by the fact that similar parameter selection is reported in [1], so a direct comparison is possible. For the autoencoder network's bottleneck also a 80 size layer was considered for the same reasoning.

The arbitrary selections of $k = 54$ and $k = 80$ for the k-means clustering of the Lampung and MNIST can be motivated by two facts. First, the larger the cluster number we consider, the larger intra class variance will be obtained. Secondly, this parameter controls the size of the data to be labeled. In our case the human experts should label 162 and 240 images for the Lampung and MNIST data, respectively. In percentages this would be 0.48% for the Lampung characters and 0.4% for MNIST.

	ka*	nga*	pa*	ta	da	na*	ca*	nya	ya	wa	ne.*
ka*	360	0	0	2	1	0	0	0	0	0	4
nga*	3	256	1	8	0	4	0	0	0	0	0
pa*	1	0	373	1	0	0	0	0	0	0	3
ta	9	14	0	133	2	0	0	0	0	0	3
da	8	1	1	19	66	1	0	0	0	0	8
na*	6	43	0	0	0	46	0	0	0	0	0
ca*	2	0	6	0	0	0	46	0	0	0	0
nya	0	13	3	2	0	2	0	0	6	0	0
ya	1	1	3	0	0	0	1	0	33	0	0
wa	1	5	2	5	0	0	0	0	0	0	0
ne.*	10	6	6	5	1	0	1	0	1	0	93

Table I: Confusion matrix for Lampung using a K-nearest neighbor ($K = 1$)

For the Lampung character dataset $33,553$ image samples were considered for the semi-automatic labeling. After the final voting in 45.44% only 2 classifiers agreed, while in 45.99% all 3 classifiers agreed upon the label. The remaining 8.57% cases were undecidable as each classifier voted for a different label. Considering only the samples where there was an unanimity on the selected labels, the result of the K-nearest neighbor was 60% for the manually labeled test set. This rather low recognition score is due to the facts that the labels agree only in 45.99% of the cases, and the K-nearest classifier is sensitive to distortions and can not distinguish between almost identical character shapes.

Analyzing the confusions we realized the fact that some classes are really similar and only just short strains differ in the different characters. Re-labeling and merging the initial 20 classes into 11 classes, the results improved considerably.

The characters ka(), ga() and sa() were merged into class ka*. The characters nga(), a() and la() were merged into class nga*. The characters pa(), ba() and ma() were merged into class pa*. The characters na() and ja() were merged into class na*. The characters ca() and ha() were merged into class ca*. Similarly, the characters nengen( ) and noise() were merged into class ne.*.

The unanimity vote (Eq. 1) increased to 75.40%, while the votes for only 2 classifiers (Eq. 2) dropped to 22.27%. In that case the recognition scores on the test set also ameliorated considerably. The good recognition score obtained considering only 11 classes achieved 86.21%. A detailed result table with confusions can be seen in Table I.

For MNIST data $60,000$ image samples were considered in the semi-automatic labeling. After the final vote in 37.69% of cases only 2 classifiers agreed (Eq. 2), in 54.76% of the cases there was an unanimity (Eq. 1) about the label's choice and in the remaining 7.55% the classifiers voted differently. For the unanimously voted patterns the correct labeling was 96.37%. This error measurement was possible due to the available labels for the training set.

Considering only the data where all of the classifiers agreed (Eq. 1), we used the simple K-nearest neighbor to measure the quality of the labeling performed on our

	0	1	2	3	4	5	6	7	8	9
0	971	2	0	0	0	1	3	1	2	0
1	0	1130	3	0	1	0	1	0	0	0
2	13	4	991	2	0	0	3	10	9	0
3	6	2	10	942	0	11	0	10	19	10
4	1	9	2	0	921	0	9	1	2	37
5	27	1	0	32	2	770	19	2	29	10
6	10	2	0	0	2	2	942	0	0	0
7	1	20	9	0	7	0	0	974	0	17
8	15	2	7	19	5	8	5	6	901	6
9	13	9	3	5	14	1	1	15	9	939

Table II: Confusion matrix for MNIST using a K-nearest neighbor ($K = 1$)

data. The accuracy of this simple and basic classifier already produced 94.81% (K=1) and 94.77% (K=3). A detailed result table with confusions can be seen in Table II.

This result is directly comparable with the result (95.0%) reported by LeCun et al. [1] for K-nearest neighbor classifier. While they used the label knowledge for all the $60,000$ training samples, in our case just 240 "labelings" were necessary.

The confusions between classes like: (4,9), (3,5,8), (7,9) can be explained with the poor capabilities of the K-nearest neighbor and the underlaying distance metric used in our experiment, namely the Euclidean distance.

IV. Conclusion

In this paper a new strategy for separated character labeling is presented. To create new benchmark character datasets we propose – instead of labeling the data manually – a new semi-automatic method which is fast and limited to a negligible amount of human interaction.

The method considers as input the images to be labeled and different data abstractions like the raw image, Principal Components and an autoencoder network are used the represent the data. The different representations are than clustered in an unsupervised manner. The only labeling effort is made to label the clusters based on their centroids. Finally, to exploit the complementarity of the different data representations an ensemble voting scheme will decide for the labels based on unanimity vote.

The 86.21% recognition rate for Lampung character -to our best knowledge being the very first attempt to recognize this script - and 94.81% for MNIST considering only 162 and 240 "labeling operations", show the importance of the method and provides a reliable labeling framework to handle unknown datasets.

The more complex feature representations are used in data representation combined with more sophisticated unsupervised clustering techniques the more precise data separation can be achieved which can lead to more accurate labeling.

Acknowledgement

This work has been supported by the German Research Foundation (DFG) within project **Fi799/3** and by the Directorate General of Higher Education, The Ministry of National Education, Republic of Indonesia. The authors would also like to acknowledge the support of students of SMKN 4 Bandar Lampung, Indonesia for being contributors to the Lampung dataset.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Intelligent Signal Processing*. IEEE Press, 2001, pp. 306–351.

[2] K. C. Leung and C. H. Leung, "Recognition of handwritten Chinese characters by critical region analysis," *Pattern Recognition*, vol. 43, pp. 949–961, March 2010.

[3] S. Mozaffari and H. Soltanizadeh, "ICDAR2009 handwritten Farsi/Arabic character recognition competition," in *International Conference on Document Analysis and Recognition*, 2009, pp. 1413–1417.

[4] U. Bhattacharya and B. Chaudhuri, "Databases for research on recognition of handwritten characters of Indian scripts," in *International Conference on Document Analysis and Recognition*, vol. 2, 2005, pp. 789 – 793.

[5] S. Vajda and G. Fink, "Exploring pattern selection strategies for fast neural network training," in *International Conference on Pattern Recognition*, 2010, pp. 2913 –2916.

[6] P. T. Daniels, *The World's Writing Systems*. Oxford University Press, 1996.

[7] D. Ghosh, T. Dube, and A. Shivaprasad, "Script recognition: A review," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2142–2161, December 2010.

[8] S. Mozaffari, K. Faez, F. Faradji, M. Ziaratban, and S. M. Golzan, "A comprehensive isolated Farsi/Arabic character database for handwritten OCR research," in *International Workshop on Frontiers in Handwriting Recognition*, 2006.

[9] N. Stamatopoulos, G. Louloudis, and B. Gatos, "Efficient transcript mapping to ease the creation of document image segmentation ground truth with text-image alignment," in *International Conference on Frontiers in Handwriting Recognition*, 2010, pp. 226–231.

[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

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

[12] S. Vajda, T. Plötz, and G. A. Fink, "Layout analysis for camera-based whiteboard notes," *Journal of Universal Computer Science*, vol. 15, no. 18, pp. 3307–3324, 2009.