J-MOCR-AND 2011

Proceedings of the
Joint Workshop on Multilingual OCR and
Analytics for Noisy Unstructured Text Data

September 17, 2011
Beijing, China

Santanu Chaudhury, Lipika Dey, Venu Govindaraju, Daniel Lopresti, Prem Natarajan, Christoph Ringlstetter, Shourya Roy
Editors

HALAMAN PENGESAHAN

Judul	:	Lampung – A New Hanwdritten Character Benchmark: Database, Labeling and Recognition
Penulis	:	**Akmal Junaidi**, Szilárd Vajda, Gernot A. Fink,
NIP	:	19710129 199702 1 001
Instansi	:	Jurusan Ilmu Komputer, Fakultas MIPA, Universitas Lampung
Publikasi	:	Proceedings of Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data
		ISBN : 978-1-4503-0685-0
		Page 105-112, 17 September 2011
Penerbit	:	ACM International Conference Proceedings Series

Bandar Lampung, 1 Oktober 2020

Mengetahui,
Dekan FMIPA Universitas Lampung

Dr. Eng. Suripto Dwi Yuwono, M.T.
NIP. 19740705 200003 1 001

Penulis,

Dr. rer. nat. Akmal Junaidi, M.Sc.
NIP. 19710129 199702 1 001

Menyetujui,
Ketua Lembaga Penelitian dan Pengabdian kepada Masyarakat
Universitas Lampung

Dr. Ir. Lusmeilia Afriani, D.E.A.
NIP. 19650510 199303 2 008

Best MOCR Student Paper Award

Joint Workshop on Multilingual OCR and
Analytics for Noisy Unstructured Text Data

Presented to

Akmal Junaidi

For the paper

"Lampung – a New Handwritten Character Benchmark:
Database, Labeling and Recognition"

Co-authored with Szilard Vajda and Gernot A. Fink

Santanu Chaudhury, Lipika Dey, Vera Govindaraju, Daniel Lopresti, Prem Natarajan, Christoph Ringlstetter, Shourya Roy

Workshop Co-Chairs

J-MOCR-AND 2011 • September 17 • Beijing, China

Proceedings of the Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data

September 17, 2011
Beijing, China

Edited by
Santanu Chaudhury, IIT Delhi
Lipika Dey, TCS Innovation Labs
Venu Govindaraju, University at Buffalo, SUNY
Daniel Lopresti, Lehigh University
Prem Natarajan, Raytheon BBN Technologies
Christoph Ringlstetter, University of Munich
Shourya Roy, Xerox, India

Sponsored by **Raytheon** BBN Technologies

ACM International Conference Proceedings Series
ACM Press

The Association for Computing Machinery
2 Penn Plaza, Suite 701
New York New York 10121-0701

ACM ISBN: 978-1-4503-0685-0

Conference Organization

Workshop Co-Chairs
Lipika Dey, India
Venu Govindaraju, USA
Daniel Lopresti, USA
Prem Natarajan, USA
Christoph Ringlstetter, Germany
Shourya Roy, India

Publications and WWW Chair
Srirangaraj Setlur, USA

Program Committee
Gady Agam, USA
Sophia Ananiandou, UK
Henry Baird, USA
Roberto Basili, Italy
Abdel Belaïd, France
Anurag Bhardwaj, USA
Indrajit Bhattacharya, India
Huaigu Cao, USA
Mohamed Cheriet, Canada
David Doermann, USA
Tanveer Faruquie, India
Gernot Fink, Germany
Jennifer Foster, USA
Basilis Gatos, Greece
Randy Goebel, Canada
C V Jawahar, India
Gareth Jones, Ireland
Gyeonghwan Kim, Korea
Fumitaka Kimura, Japan
Louisa Lam, Canada
Marcus Liwicki, Germany
Sriganesh Madhvanath, India
Volker Märgner, Germany
Vincent Ng, USA
Ifeoma Nwogu, USA
Shinichiro Omachi, Japan
Jaehwa Park, Korea
Sebastian Peña Saldarriaga, Canada
Xujun Peng, USA
Sitaram Ramachandrula, India
Ag Ramakrishnan, India
B. Ravindran, India
Eric Ringger, USA
Klaus Schulz, Germany
Zhixin Shi, USA
L V Subramaniam, India
Maosong Sun, China
Hironori Takeuchi, Japan
Antal Van Den Bosch, Netherlands

Foreword

It is our great pleasure to welcome all participants to the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data (J-MOCR-AND). This year's workshop is born of a merger between what would have been the Fifth Workshop on Analytics for Noisy Unstructured Text Data and the Third Workshop on Multilingual OCR. The two topics are naturally related and we look forward to an exciting exchange of new research ideas.

J-MOCR-AND is being held on September 17th in Beijing, China, in conjunction with the Eleventh International Conference on Document Analysis and Recognition (ICDAR 2011). The workshop is organized as a single-track, one day event, with eight oral papers drawn from the AND community and ten oral papers drawn from the MOCR community. An invited talk on "Multilingual and Noisy Data Challenges in Large-Scale Book Scanning" by Ashok Popat from Google begins the day.

All papers underwent the standard peer review process and will appear in the official workshop proceedings to be published in the ACM International Conference Proceedings Series, available online as part of the ACM Digital Library. Please note that citations should reference the online proceedings and not the unofficial hardcopy proceedings distributed at the workshop.

Two best student paper awards will be presented at the conclusion of the workshop: one for the best student paper from the AND pool, and another for the best student paper from the MOCR pool.

The success of the J-MOCR-AND workshop is due in large part to the efforts of our program committee members. Special thanks are due to the ICDAR 2011 organizers in Beijing for facilitating the local logistics, especially Cheng-Lin Liu and Cuiling Lan. We express our appreciation to Srirangaraj Setlur for managing the J-MOCR-AND website, and to Jeanne Steinberg at Lehigh University for her help in assembling the hardcopy proceedings. Finally, we wish to take this opportunity to thank Raytheon BBN Technologies for their generous financial support of the workshop.

We hope you find J-MOCR-AND to be a rewarding experience and that you enjoy your time in Beijing.

Santanu Chaudhury, IIT Delhi
Lipika Dey, TCS Innovation Labs
Venu Govindaraju, University at Buffalo, SUNY
Daniel Lopresti, Lehigh University
Prem Natarajan, Raytheon BBN Technologies
Christoph Ringlstetter, University of Munich
Shourya Roy, Xerox, India
J-MOCR-AND Co-Chairs

Table of Contents

Analytics for Noisy Unstructured Text Data

Multilingual OCR

Index

Lampung – a New Handwritten Character Benchmark: Database, Labeling and Recognition

Akmal Junaidi
Computer Science Department
TU Dortmund
Dortmund, Germany
akmal.junaidi@udo.edu

Szilárd Vajda
Robotics Research Institute
TU Dortmund
Dortmund, Germany
szilard.vajda@udo.edu

Gernot A. Fink
Computer Science Department
TU Dortmund
Dortmund, Germany
gernot.fink@udo.edu

ABSTRACT

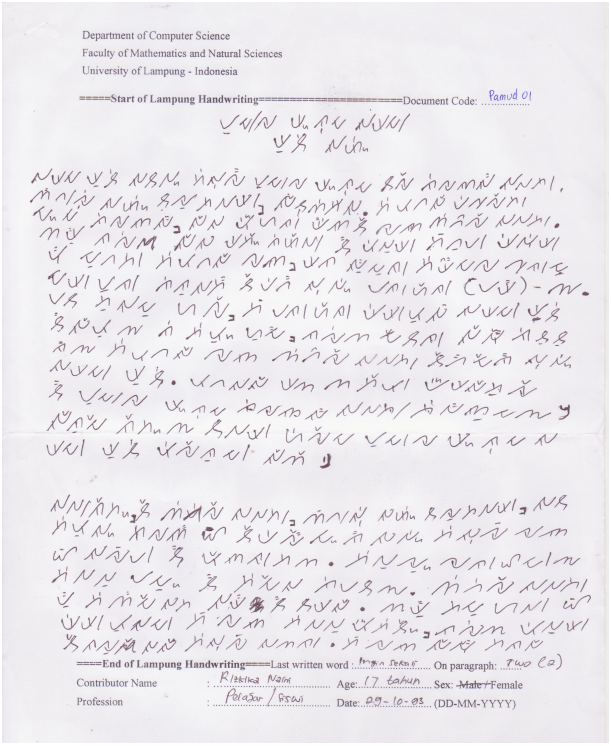
This research paper deals with our effort of creation and recognition of isolated Lampung characters, a script originated from Indonesia. The aim is to describe this new script with all its peculiarities, propose a labeling scheme to manage a large isolated character dataset and finally a recognition scheme based on water reservoir concept. The Lampung script originally descending from Brahmi script is used in Lampung Province and it is close to extinction if no such initiative as ours will direct the focus to this cultural heritage. The collected dataset contains isolated characters coming from fairy tales transcriptions and were annotated with a semi-automatic labeling method using a limited human effort. Our attention is focused not only on the database collection but on recognition as well. For this purpose a water reservoir based feature set is proposed exploiting the different cavities and the subsequent measures of the character shapes. The experimental results (94.27%) prove the efficiency of the method considering a brand new script and feature set.

Keywords

Lampung script, handwritten character recognition, semi-automatic labeling, character database, water reservoir features

1. MOTIVATION

The Lampung script is one of the few scripts belonging to Indonesia. The script has an old-fashioned name "kaganga" as well. The later comes from the name of the 3 first letters, "ka", "ga", and "nga", respectively. Since a long time ago, this script has become a cultural heritage of the Lampung Province. Many ancient documents have been found, some exposed in the local museums, while some others are with international museums worldwide. However, community awareness of having this script is minimalistic. The script is considered rather an ornament with less application on writing. This lamentable situation will not help to preserve this ancient script. If this condition remains unchanged, the heritage will eventually disappear in the future.


Department of Computer Science
Faculty of Mathematics and Natural Sciences
University of Lampung - Indonesia

=====Start of Lampung Handwriting=====Document Code: Pamud 01

=====End of Lampung Handwriting=====Last written word : ... On paragraph: ...

Contributor Name : ... Age: ... Sex: ~~Male~~/Female

Profession : ... Date: ... (DD-MM-YYYY)

Figure 1: A filled form of Lampung document

In this context, we consider that Lampung script is feasible for research in the area of handwriting recognition. However, it is hard to carry out research since in our best knowledge, there is no prior knowledge or databases available to support research on this Indic related script.

This research paper is one endeavor to trigger a larger focus and attention toward the script as well to try retaining the script from its extinction. Furthermore, our research aim is to generate a new character database for Lampung with similar objectives as mentioned in [1, 13]. We completely labeled the handwritten separated Lampung character dataset considering our semi-automatic labeling strategy proposed to label large character datasets involving minimal human efforts [23].

J-MOCR-AND '11 Beijing, China

Even more, we provide not just a possible new benchmark dataset to the handwriting community, but we also report some recognition scores and compare our achievements with the preliminary results reported in [23].

2. LAMPUNG SCRIPT

In the upcoming section a brief description of the script will be given w.r.t. the historical evolution of the script, the alphabet and the specificities of this historic Indonesian script.

2.1 Brief History

The Lampung script was inherited from the ancient script used in South India, originated from the family of the Brahmi script. More precisely, the script descended of Devnagari script [3]. Indic script was not the only script that influenced the Lampung, but Arabic structure also contributed to the development of the Lampung, especially in the usage of diacritics. Similar to Arabic writing system, the Lampung has diacritics which can be combined with the main alphabet. However, not only on the top and bottom, but the script also has diacritics located on the right of the main characters.

The origin of the script is the only historical information about the Lampung. There is no road map about the exact period of when and how the script started and spread in Lampung Province. It is hard to find relevant information or documents concerning this milestone. The only evidences are some authentic ancient documents written in the Lampung available in Indonesia and several foreign countries. Some of them belong to and are stored by Lampung natives. While others are collection of the Province Museum of Ruwa Jurai in Bandar Lampung, the National Library in Jakarta, the University of Leiden in The Netherlands, the School of Oriental and African Studies in London and one document is with the National Library of India [19].

2.2 The Lampung Alphabet

Over the centuries the ancient Lampung script has evolved from its descender, the Devnagari script. This evolution process has changed the script to some extent until the recent shape. The current script is much simpler than the ancient one.

The old Lampung script consisted of 19 letters, while the recent script has one letter more. Each letter consists of one piece of symbol except two of them, the *ra* and *gha* (see Fig. 2), which both have a counterpart symbol forming one letter shape. Some letters contain one or two zigzags, while others have more complex structures. In addition, some of them have a straight line connected to the zigzags. The stroke of each single letter leads to the upper right side, so that the letter does not seem to stand in an upright but instead dictating slightly to the northeast direction. All of those main letters are called "kelabai sughat" and are depicted in Fig. 2.

The writers could compose a word by using one or more of these letters. One syllable in a word can be constructed by a letter with one or more diacritics. In some cases, a syllable can independently be composed by a letter without using any diacritics.

2.3 Diacritics

Diacritics in Lampung are placed around the letters. The placement of this position is intended to make a particular

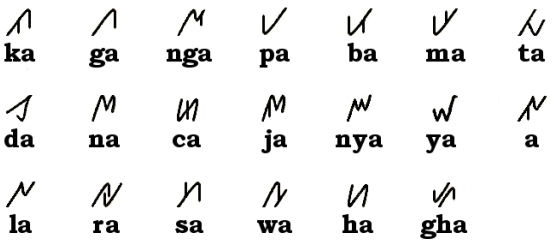


Figure 2: The Lampung alphabet

syllable pronunciation which modifies the sound of its main letters. In one syllable, it could also be possible having more than one diacritic around the letter.

As explained previously, the diacritics around the letter could be placed on one of three possible positions. Thus, based on their position the diacritics are grouped into 3 types. They are 6 diacritics on the top, 3 on the bottom, and 2 on the right side of the letter (see Fig. 3).

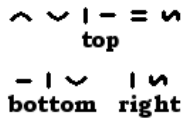


Figure 3: Diacritics in Lampung Script

The diacritic on some documents can be distinguished in a certain appearance and style. There are people who write the vertical diacritic as it is, whereas some people make this mark as right slanted line. Other variations are also to be found for the diacritics of the half circle shape. Some people create this diacritic which has angular side instead of warping segment. Such variations usually depend on the person writing the document.

As shown in Fig. 3, the Lampung script has in total 11 diacritics, and each of this diacritic has its own name. Their size are smaller, around one tenth of the letter. All of these diacritics are called "benah sughat".

2.4 Punctuation Marks

In writing, some punctuation marks exist as a part of Lampung's writing system. The role of the punctuation marks as in other writing systems is to facilitate the reader of a proper interpretation of the written sentences. Fig. 4 shows the symbol of punctuation marks in the Lampung.

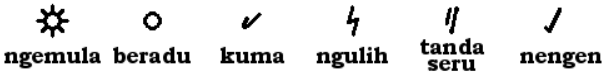


Figure 4: Punctuation marks in Lampung script

Among of those marks, two marks are frequently used, the *nengen* and the *beradu* marks [19]. The *nengen* mark is always employed after a single letter to create one "closed syllable" sound, and it can not act unaccompanied. The *beradu* mark is a circle mark to end a sentence as a full stop

symbol. The rest of other marks i.e. the *kuma* (comma), the *ngulih* (question mark) and *tanda seru* (exclamation mark) are still used in handwritten documents, based on their necessity, but they are not used as frequently as *nengen* and *beradu*. In contrary, the *ngemula* mark, a unique punctuation mark for starting a sentence, is hardly found in any texts or manuscripts. There is no explanation why the writers neglected this mark in their writing.

3. RELATED WORK

In the past decade, many research initiatives aimed to create isolated handwritten character data collections for different scripts like, Persian [12, 13], Arabic [8], and Indic scripts, like Bangla [11], Devnagari and Oriya [1].

These researches initiated proper and large scale databases that could be re-used by other researchers. Using the same dataset, some researches reported with their results for various aspects on handwriting research subjects. The comparative analysis among them enabled many approaches to be improved over the time.

The increasing interest on database development has advocated other specific scripts [4] to be investigated as a new challenge in the area of document analysis and recognition. For a newly introduced script, a new dataset is a mandatory ingredient for further research.

Preparing a dataset is an important phase in building such database. Its availability is essential in handwriting research. One of the hard tasks in dataset design besides collecting the handwritten documents, is the accurate labeling. This task can be accomplished either by manual labeling or a semi-automatic strategy [23]. Up to now, to our best knowledge, no method is offering a fully automatic labeling solution. Thus, most of the work on labeling was still performed manually [1, 8, 13] involving human experts which is a tedious and costly work. However, this is far from an efficient ground truth processing criteria as declared by Stamatopoulos et al. [21].

Once the dataset is ready, it is still immature but potential to be a standard database research. The database itself provides coherent research data which will encourage comprehensive research leading to comparable results. The improvement can be achieved in different research point of view by a larger research community. Researchers worldwide can apply their methods toward this dataset, producing results or improving the existing ones, and update the benchmarks. This is the way how a representative dataset play an important role to accelerate the benchmark.

In separated character recognition different strategies have been considered to extract features for the different type of recognizers [9]. While LeCun et al. [8] considered as input the raw image in the LeNet5, the so-called convolutional neural network, some others have used statistical and structural features [10] to describe the different character shapes. Liu and Suen [9] considered local stroke orientation/direction to recognize Arabic and Farsi digits.

The water reservoir concept is relatively new in handwriting research. In early usage, it was introduced to segment touching numerals [15]. If two numeral are connected to each other, they would create large cavities in between. The idea was to interpret the different cavities as water reservoirs. Pouring water from top or bottom in these reservoirs, measuring the height and width of the different reservoirs can describe structurally the connected shape. This research reported success for segmentation of touching numerals of French bank checks, obtaining 94.8% for separation.

A similar segmentation task was reported in [16] to separate touching characters in a Bangla word recognition scenario. In [18], the approach was applied for pre-segmentation of multi-script strings written in English, Hindi, and Bangla. The approach was also applicable to compose the features for orientation detection of 11 major Indian scripts [2]. In [20] the authors used the water reservoirs to compute text line height estimations.

A first attempt in considering water reservoir type features has been described in [17], where Pal et al. considered a recognition scheme for isolated Malayalam handwritten numerals. The reported results for the rather small dataset were very promising, but the authors combined water reservoir type features with other features like loops, profiles and contour using a decision tree for classification purpose. Such a binary decision tree implies specific knowledge from the analyzed digits, hence the method can not be reused in similar scenarios, because other type of rules, specific to the data have to be established first.

In this paper we concentrate on describing in detail the data collection effort, the labeling process using a semi-automatic tool to initially label the data involving minimal human effort, and, finally the use of the water reservoir concept for recognizing handwritten isolated Lampung characters in a connectionist framework.

4. LAMPUNG CHARACTER DATASET

In order to completely understand such a data collection initiative, a brief description of the whole process is given in the upcoming sections.

4.1 Data Collection

The number of contributors participating in data collection is 82 writers. The contributors are students of a senior high school in Bandar Lampung, Lampung Province, Indonesia. They are the 10^{th} and 11^{th} grade students that are 16 years old in average and have an adequate education to fulfill the task of writing text in Lampung. The contributors were instructed to write portions of text from fairy tales by filling in forms as the one shown in Fig. 1. To preserve the variation in data, each contributor was requested to do a transcription only once. Similar to that purpose, the same page of the tales was written at most by two contributors. This collecting strategy aims at producing a high variability in the textual content of the acquired document samples.

4.2 Data Acquisition

Each handwritten document coming from the different contributors was digitized at 300 dpi using a Canon MP160 scanner. Most of the contributors were lacking the awareness of keeping the forms carefully and wisely before submission. Hence, a small number of the scanned image data contain unwanted image objects due to folding, dirty spots, erroneous writings, etc. Some folding effects can be spotted in Fig. 5.

All digital images coming from scanning process were saved in color JPG format. Each of those images were manually cropped to acquire the writing part and get rid of contributor profile and information texts residing at the top and the bottom of each form filled by the writers (see Fig. 1 and 5).

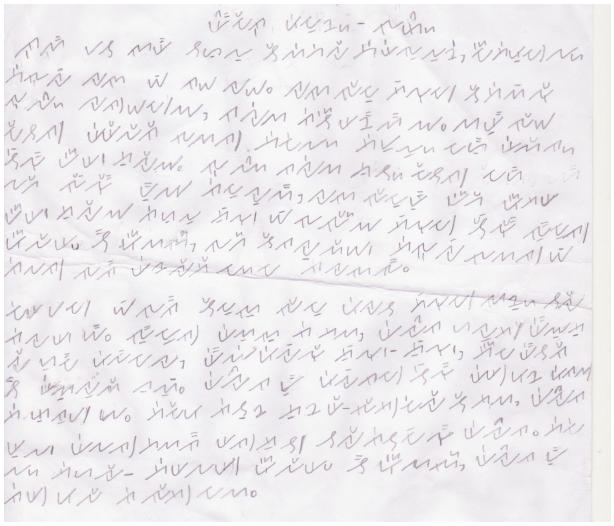


Figure 5: A folded document containing artifacts

4.3 Data Preprocessing and Filtering

Each cropped document was preprocessed in the same way. First, the raw image data was converted into a binary image using Niblack's algorithm with local threshold [14]. The connected components (CC) were extracted from those images and some CC measures were recorded. A simple calculation was derived from all connected component attributes to set up three filtering parameters i.e. size, aspect ratio and pixel density [24]. Since each document page of the dataset is unique, a document wise estimation was considered to estimate the parameter closest to actual value of the data.

These parameters were applied to perform a fully automatic filtering among all CCs so that that noises are eliminated. For the purpose of identifying only the Lampung letters (see Fig. 5), this preliminary research assumed diacritics and punctuation marks as being noises, thus to be removed. Both small and large CCs were discarded based on a threshold calculated on the average CC size.

The aspect ratio was not computed from a document based processing, but it was determined straightforward by a specific value. Therefore, the lower bound and upper bound of the aspect ratio parameters were set to 0.5 and 4.0, respectively. Both parameter values could exclude the noise, which were like a long vertical line if its ratio was less than 0.5 or a long horizontal line if its ratio was greater than 4.0. Such filter was useful to discard artifacts coming from folding, etc.

There were also possibilities the CCs with normal large area to be considered as a noise. For example, CCs with either less or too much black pixels will likely be noise rather than character candidates. To handle those, we filtered them using a document based threshold, calculated from the pixel average density.

The outcome of the filtering produced CCs with different dimensions w.r.t. height and width., Finally, the different CCs were mapped to 20x20 size images by applying a linear normalization. This normalized image size can be explained also by the fact that the labeling process (see Section 5) uses the original image to derive some cluster labels.

4.4 Data Statistics

To give a general overview, we noted some important information about the collected data. We summarized them in the Table. 1.

Table 1: Statistics summary of the data

Stat. Parameters	Attributes
Male Contributors	20 persons
Female Contributors	62 persons
Number of page samples	82 pages
Total number of words	11,722
Total extracted CCs	35,193
Average CCs/Page	429
Collection Period	December 2010

5. SEMI-AUTOMATIC CHARACTER LABELING

For large character data sets the manual labeling is a costly and time consuming process involving multiple human experts to crosscheck the different labels. As stated by Stamatopulous et al. [21] the efficient ground truth for document image processing should be a "quick and low cost" solution.

In our previous work [23] such a semi automatic labeling strategy is described involving minimal human interaction. The scope of this paper is not to label the data, but using the labels to recognize the different Lampung characters. In the upcoming paragraphs just a brief description of the method is given, for further details, please refer to [23].

The labeling process is built upon three major steps. First the data, namely the Lampung characters, should be represented differently in order to exploit further their separability on different data abstraction levels. For this purpose the normalized and centered raw image, the PCA (Principal Component Analysis) reduced subspace and finally another, more sophisticated data dimensionality reduction method, based on so-called autoencoder network, proposed by Hinton et al. [5] were considered to generate different data abstraction levels.

Second, the different data representations should be clustered. For the raw image, the PCA subspace representation and for the autoencoder based data reduction the same unsupervised clustering method was deployed, namely the generalized Lloyd algorithm. The number of partitions to be considered is controlled by a parameter k. The bigger k is the more clusters are generated which then are labeled by the human experts. However, instead of labeling all the samples from each partition, the human expert labels only the centroids of each cluster and the rest of the samples inherit the label of its centroid. For the three different data abstractions this implies $3k$ labeling operations. For the current Lampung character dataset or MNIST handwritten digits [8, 23] such labeling procedure will not imply more than a few hundreds of labeling operations.

The last step in the labeling process is to infer in a robust manner labels to the analyzed Lampung characters. For each character three labels are assigned based on the clustering described above. A simple voting [7] will assign the final label of each character.

Assuming 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

$$\sum_{i=1}^{C} l_{i,k} = C. \quad (1)$$

This voting mechanism will decide for a label for each character. In order to assure an accurate labeling, only those labels will be accepted as being valid where there was an unanimity decision with respect to that label. Finally, we end up with some labeled and unlabeled data alike.

The distinction between the already existing voting schemes and the current solution lies in the fact that our voting scheme serves only to label the training data and a classifier is built on top of this label information.

6. FEATURE EXTRACTION

Feature extraction in separated handwritten character recognition is one of the most important steps [9]. The features, whatever their nature, should be easily computable and should be discriminating ones to help the different classifiers to decide for each character which label to assign with.

Before extracting features, we first skeletonized each normalized image. For some features, these skeletonized images were sufficient, whereas for some others the skeleton images were split into smaller grid areas with size 4x4 pixel each. For the 20x20 images, this grid division provided 25 identical grid areas.

Each of Lampung character contains at least a curve. See Fig. 2 for the details. This structure could be a conceivable choice for feature representation of the Lampung characters. These features combined with any other features (e.g. profiles, density), could serve with success for the recognition of different Lampung characters.

First, we constructed a simple feature vector containing branch points [6], end points [6] and the pixel density [24]. The number of branch points, end points, and the pixel density were counted on each of grid area of an image. The number of those features are normalized by the total number of pixels of the grid. Thus, there were 25 measures for each image, forming a concatenated vector of 75 components.

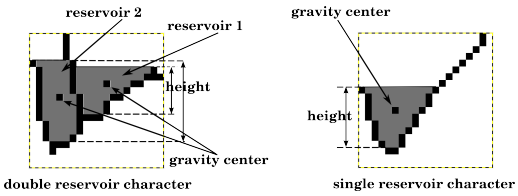


Figure 6: Top reservoir

Inspired by the success of the so-called water reservoir principle in handwriting, we propose to use the measures derived from this concept to serve as features for the recognition. In this approach, first, the algorithm tracks the skeleton character image pixel by pixel and records the first end point. During this inspection, it searches for the cavity. If there are pixel transitions from downward to upward or vice versa,

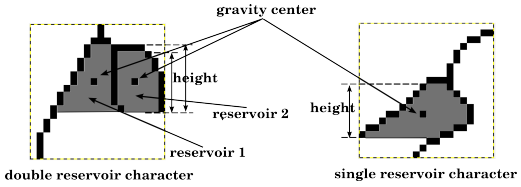


Figure 7: Bottom reservoir

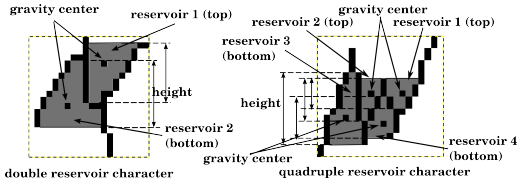


Figure 8: Top and bottom reservoir

the algorithm finds the cavity and records the closest end point which forms the cavity. Then, the algorithm will seek the next cavity with the new searching by repeating the procedure in the same manner. For each cavity, the water is poured into the cavities until the water level reaches the lowest end point. The regions filled by the water were identified as water reservoirs (see Fig. 6, 7, 8).

For each water reservoir we measure the height of the reservoir while computing its volume. The width of each reservoir is also calculated. Instead of direct calculation of width, we approximate the width of the reservoir by dividing the volume over the height. Finally, the gravity centers of the reservoirs were also identified and included as part of the features.

From the nature of the script, we note that there are two types of reservoirs, top reservoir and bottom reservoir. The first type, the top reservoir is the reservoir where its two end points situated on the top side so the water can be directly poured from the top (see Fig. 6). The second one has its end points at the bottom side (see Fig. 7). To distinguish them in the feature vector, we define top (1) and bottom (-1) reservoirs.

To represent our water reservoirs, we arranged the features in such a way that each single reservoir vector consists of 6 values. First value indicates the type of the reservoir. The second and third components were the x and y coordinate of reservoir gravity center normalized with respect to the character height and width. On the fourth position, we considered the size (volume) of the reservoir. And the last two values refer to the height and width.

Ideally, the Lampung characters should have at least one reservoir disregard of its type. After further inspection, we found that a letter could have a maximum of 2 top and 3 bottom reservoirs. Based on this, we structured the feature vectors as 5 consecutive single reservoir vectors (6 elements). This means a feature vector for each character image has 30 dimensions.

However, due to variation in the writing style, effect of normalization process and/or the noise, some letters in our dataset might have more than the maximum number of reservoirs. We only focussed on the big volume size reservoirs, because they certainly belong to the script, whereas the small

ones could accidentally be generated by the aforementioned factors. Thus, in that case, we considered only the 2 top reservoirs and 3 bottom reservoirs and discarded the excess reservoirs after sorting their volume in descending order. On the other hand, if the respective character has less than the maximun number of reservoir or no reservoir at all, there is no feature representations for those, meaning that respective vector elements will be zero.

Finally, the previously mentioned feature representations were concatenated to form a 105 dimensional vector comprising of statistical features (density) and structural features (end points, branch points and water reservoirs). In our recognition experiments we have considered these three feature representations.

7. CHARACTER RECOGNITION

In our previous work [23], we classified the sample letters based on the K-nearest neighbor (kNN) algorithm. While in that scenario the labels were predicted by the method, in this case all the labels are available, partially based on the method mentioned before and correcting only the erroneously assigned labels.

For the current study, we applied a Neural Network (NN) approach [8, 22] to train the different feature representations. The NN used in this recognition scheme is a multi-layer perceptron. The architecture of the network consist of 3 layers, input layer, hidden layer and output layer.

In our case, the size of the input layer was equal to the dimension of the feature vectors, namely 75, 30 and 105. The output layer size was set to 11 due to the fact that there are 11 character classes in this Lampung dataset. We chose 11 classes instead of 20 classes since some image classes are looked very similar each other and they vary only on a very small segment out of a single image size which are 400 pixels. For the size of the hidden layer we experimented several setups. An example of our network configuration $105 - 105 - 11$, meaning that 105 neurons for input layer, 105 neurons for hidden layer and 11 neuron of output layer. This specific setup was derived from different trial runs.

The training algorithm chosen for our experiments was the resilient backpropagation. This algorithm allows the training to be executed without easily getting stuck on the local minima but instead using global optimization information to decide for optimal solution. For activation function the well known sigmoid function was applied.

8. EXPERIMENTS

In the forthcoming section we describe the dataset, the different extracted features and the methodology applied to provide the different results reported in this paper.

8.1 Data Description

The dataset is split into a training set consisting of $21,122$ (60%) training samples, $10,547$ (30%) test samples, and the remaining $3,524$ (10%) samples were considered for validation purposes. The labeling of this character dataset was performed using the labeling presented in our previous work [23]. Instead of labeling manually those $35,193$ character images, we preferred to run the semi-automatic labeling algorithm first and based on visual inspection correct only those labels where a significant deviation in shape could be observed. With this approach we had to re-label only about 20% of the data.

In our previous work [23] by analyzing the confusions among the different Lampung character classes we realized that some of them are almost similar and just some minor strain distinction on the shape does the difference. In that sense we merged some character classes and we propose to separate those merged classes in our further work. 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.**.

Table 2: The Lampung dataset distribution

Class	Distr.	% Distr.	# Training	# Val.	# Testing
ka*	8077	22.95%	4847	808	2422
nga*	5352	15.21%	3212	536	1604
pa*	8629	24.52%	5178	863	2588
ta	3092	8.79%	1856	310	926
da	2157	6.13%	1295	216	646
na*	1756	4.99%	1054	176	526
ca*	1394	3.96%	837	140	417
nya	773	2.2%	464	78	231
ya	660	1.88%	396	66	198
wa	254	0.72%	153	26	75
ne.*	3049	8.66%	1830	305	914
Total	35193	100%	21122	3524	10547

The number of labeled data in our dataset is $35,193$. Since the document collection was generated by many participants writing text from Indonesian fairy-tales, the number of character samples of each class was not equally distributed. The character class distribution of the Lampung dataset is given in the Table 2. It can be observed from this table that based on the number of character, the biggest class is class *pa** with the size of $8,629$ elements, and the smallest class is class *wa* with the size of only 254 elements.

8.2 Results

For the experiments we trained and tested the Lampung data using our neural network. To provide robust results, we run the training three times with small changes in the network configuration, by modifying the training parameter and adapting the size of the hidden layer. For learning rate, we set the value in range 0.05 - 0.3. For the hidden layer size, we set different values between the input and output layer size. Although three experiments generate different level of accuracies, but their differences are relatively small. Therefore, we report the best results among those.

The first experiment involved the feature vector collected from branch points, end points, and pixel densities extracted from the grid of each CC. The recognition accuracy is 93.20%. See Table 3 for details.

The second experiment utilized the features from the water reservoir concept. As stated in Section 6, the dimension of this feature vector was only 30. Although the dimensionality is 2.5x smaller than the previous one, the performed accuracy is 91.32%. The high recognition scores achieved by the water reservoir based features proved the efficiency of the

Table 3: Summary of experiment result for training character recognition. [1]Branch point, end points, pixel density. [2]Water reservoir. [3]1 & 2

Features	#Training samp.	#Labels	#Test samp.	Class.	Rec (%)
BED[1]	21,122	21,122	10,547	MLP	93.20
WR[2]	21,122	21,122	10,547	MLP	91.32
BED-WR[3]	21,122	21,122	10,547	MLP	94.27

Table 4: Confusion results for branch points, end points, pixel density and water reservoirs

	ka*	nga*	pa*	ta	da	na*	ca*	nya	ya	wa	ne.*
ka*	2358	11	1	0	18	17	6	1	2	0	8
nga*	5	1528	5	13	2	16	1	9	0	4	21
pa*	0	3	2559	3	0	0	4	3	3	2	11
ta	1	26	4	854	6	0	0	1	1	2	31
da	20	4	0	7	598	2	0	1	1	1	12
na*	18	15	0	0	0	484	0	4	0	1	4
ca*	6	0	4	1	0	1	397	0	3	1	4
nya	0	12	2	0	0	6	0	198	6	0	7
ya	2	0	6	2	1	0	1	2	182	0	2
wa	5	8	0	1	5	0	1	1	0	47	7
ne.*	25	39	19	37	18	12	5	12	3	6	738

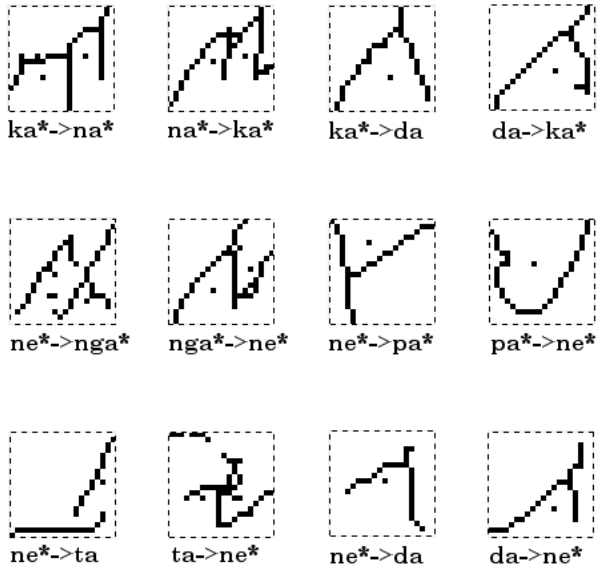


Figure 9: Misclassified samples

method and reduced considerably the dimensionality of the classification.

Finally, for the last experiment, we merged both features of the previous experiments into one bigger feature vector with 105 components. The result was the best among of the other two representations. The recognition rate was 94.27%. A detailed confusion matrix can be analyzed in Table 4. In Fig. 9 some common confusions between class samples can be observed. These confusions come from the fact that the different writers wrote the characters differently, and the different water reservoirs are located in different parts of the characters. As might be observed in Table 4 the class *ne.** is confused with the other classes. This is due to the fact that the *ne.** class gathers all the punctuation marks not discarded during the filtering and all other character artifacts coming from touching characters or characters connected to the different diacritics or punctuation not considered initially in this research.

9. CONCLUSION

In this paper a brand new database of handwritten separated Lampung characters has been provided in order to create a possible, more sophisticated benchmark data for the research community and focus the attention toward this special Indic script called "kaganga", originated from Bandar Lampung, Indonesia. Such a collection initiative could not just serve the scientific challenge, but also could be a possibility to avoid the extinction of this ancient script.

Beyond the data collection issues, the paper is also concerned about the labeling of such a large character dataset, involving the less possible human effort. The labeling was character-wise instead of line-wise due to the nature of the Lampung which is not a cursive script, hence each character is separated from other characters. In this work instead of labeling manually the characters, we applied our semi-automatic labeling strategy, hence only 20% of the data had to be relabeled by correcting the labels initially assigned by the method.

While other similar works just concentrate on the data collection initiatives, we proposed a new feature set, namely the water reservoir based measures to recognize Lampung characters. In our best knowledge, there is only one such work available for Malayalam handwritten digits [17], and our results achieved on Lampung characters are very promising. Instead of building a decision tree type classifier, we wanted to avoid the implication of human knowledge in the system, so we learned those separation rules by a neural network, creating a more general character recognition framework.

Combining the different structural features like end points, branch points and water reservoir based characteristics with a statistical feature, we achieved a recognition rate of 94.27%

(see Table 3), which is directly comparable with state-of-the-art methods [8, 9, 17, 25].

We are planning to make the dataset publicly available, preferable from the TC-11 website. This offline character dataset will definitely initiate new research initiatives to test existing methods and propose even more complex and suitable character recognition methods in the future.

10. ACKNOWLEDGEMENT

This work has been supported 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.

11. REFERENCES

[1] U. Bhattacharya and B. B. Chaudhuri. Databases for Research on Recognition of Handwritten Characters of indian Scripts. In *International Conference on Document Analysis and Recognition*, volume 2, pages 789 – 793, 2005.

[2] B. B. Chaudhuri and S. Ghosh. Orientation Detection of Major Indian Scripts. In *Proceedings of the International Workshop on Multilingual OCR*, MOCR '09, pages 8:1–8:7, New York, NY, USA, 2009. ACM.

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

[4] D. Ghosh, T. Dube, and A. Shivaprasad. Script Recognition: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:2142–2161, December 2010.

[5] G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006.

[6] M. S. Khorsheed. Recognising Handwritten Arabic Manuscripts Using a Single Hidden Markov Model. *Pattern Recogn. Lett.*, 24:2235–2242, October 2003.

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

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press, 2001.

[9] C.-L. Liu and C. Y. Suen. A New Benchmark on the Recognition of Handwritten Bangla and Farsi Numeral Characters. *Pattern Recognition*, 42:3287–3295, December 2009.

[10] L. M. Lorigo and V. Govindaraju. Offline Arabic Handwriting Recognition: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:712–724, May 2006.

[11] T. Mondal, U. Bhattacharya, S. K. Parui, K. Das, and V. Roy. Database Generation and Recognition of Online Handwritten Bangla Characters. In *Proceedings of the International Workshop on Multilingual OCR*, MOCR '09, pages 9:1–9:6, New York, NY, USA, 2009. ACM.

[12] S. Mozaffari, H. E. Abed, V. Märgner, K. Faez, and A. Amirshahi. IfN/Farsi-Database: a Database of Farsi Handwritten City Names. In *International Conference on Frontiers in Handwriting Recognition*, 2008.

[13] S. Mozaffari, K. Faez, F. Faradji, M. Ziaratban, and S. M. Golzan. A Comprehensive Isolated Farsi/Arabic Character Database for Handwritten OCR Research. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France), 2006.

[14] W. Niblack. *An Introduction to Digital Image Processing*. Strandberg Publishing Company, Birkeroed, Denmark, 1985.

[15] U. Pal, A. Belaïd, and C. Choisy. Touching Numeral Segmentation using Water Reservoir Concept. *Pattern Recognition Letters*, 24(1-3):261–272, 2003.

[16] U. Pal and S. Datta. Segmentation of Bangla Unconstrained Handwritten Text. In *International Conference on Document Analysis and Recognition*, pages 1128–1132, 2003.

[17] U. Pal, S. Kundu, Y. Ali, H. Islam, and N. Tripathy. Recognition of Unconstrained Malayalam Handwritten Numeral. In *ICVGIP*, pages 423–428, 2004.

[18] U. Pal, R. K. Roy, K. Roy, and F. Kimura. Indian Multi-Script Full Pin-code String Recognition for Postal Automation. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ICDAR '09, pages 456–460, Washington, DC, USA, 2009. IEEE Computer Society.

[19] T. Pudjiastuti. *The Lampung Ancient Script and Manuscript in Perspective of the Recent Contemporary Lampung Society (Indonesian)*. Cultural and Education Department, Republik of Indonesia, Jakarta, 1997.

[20] P. P. Roy, U. Pal, and J. Lladós. Morphology Based Handwritten Line Segmentation Using Foreground and Background Information. In *International Conference on Frontiers in Handwriting Recognition*, 2008.

[21] 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*, pages 226–231, Washington, DC, USA, 2010. IEEE Computer Society.

[22] S. Vajda and G. Fink. Exploring Pattern Selection Strategies for Fast Neural Network Training. In *International Conference on Pattern Recognition*, pages 2913 –2916, 2010.

[23] S. Vajda, A. Junaidi, and G. A. Fink. A Semi-Supervised Ensemble Learning Approach for Character Labeling with Minimal Human Effort. In *International Conference on Document Analysis and Recognition*, 2011. (in press).

[24] S. Vajda, T. Plötz, and G. A. Fink. Layout Analysis for Camera-Based Whiteboard Notes. *Journal of Universal Computer Science*, 15(18):3307–3324, 2009.

[25] S. Vajda, K. Roy, U. Pal, B. B. Chaudhuri, and A. Belaid. Automation of Indian Postal Documents Written in Bangla and English,. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(8):1599–1632, December 2009.