IJDAR (2008) 11:127–141
DOI 10.1007/s10032-008-0073-5

ORIGINAL PAPER

Arabic handwritten digit recognition

Sherif Abdleazeem · Ezzat El-Sherif

Received: 3 March 2008 / Revised: 27 September 2008 / Accepted: 5 October 2008 / Published online: 20 November 2008

Abstract In this paper, we fill a gap in the literature by studying the problem of Arabic handwritten digit recognition. The performances of different classification and feature extraction techniques on recognizing Arabic digits are going to be reported to serve as a benchmark for future work on the problem. The performance of well known classifiers and feature extraction techniques will be reported in addition to a novel feature extraction technique we present in this paper that gives a high accuracy and competes with the state-of-the-art techniques. A total of 54 different classifier/features combinations will be evaluated on Arabic digits in terms of accuracy and classification time. The results are analyzed and the problem of the digit '0' is identified with a proposed method to solve it. Moreover, we propose a strategy to select and design an optimal two-stage system out of our study and, hence, we suggest a fast two-stage classification system for Arabic digits which achieves as high accuracy as the highest classifier/features combination but with much less recognition time.

Keywords Benchmark · Arabic digits · Indian digits · Classifiers · Feature extraction · Two-stage

1 Introduction

Handwritten digit recognition problem can be seen as a subtask of the more general Optical Character Recognition (OCR) problem. However, there are some applications (e.g., postal code and bank checks reading) that are restricted to recognizing digits but require very high accuracy and speed. In addition, handwritten digit recognition problem is usually used as a benchmark for comparing different classification techniques [1].

While recognition of handwritten Latin digits has been extensively investigated using various techniques [1–8], little work has been done on Arabic handwritten digit recognition. Al-Omari et al. [9] proposed a system for recognizing Arabic digits from '1' to '9'. They used a scale-, translation-, rotation-invariant feature vector to train a probabilistic neural network (PNN). Their database was composed of 720 digits for training and 480 digits for testing written by 120 persons. They achieved 99.75% accuracy. Said et al. [10] used pixel values of the 16×20 size-normalized digit images as features. They fed these values to an Artificial Neural Network (ANN), where number of its hidden units is determined dynamically. They used a training set of 2400 digits and a testing set of 200 digits written by 20 persons to achieve 94% accuracy. In a previous paper [11], we introduced a large Arabic Digits dataBase (the ADBase—see Sect. 2.1 for more details) and devised a two-stage system for recognizing Arabic digits. The first stage is an ANN fed with a short but powerful feature vector to handle easy-to-classify digits. Ambiguous digits are rejected to the more powerful second stage which is an SVM fed with a long feature vector. The system had a good timing performance and achieved 99.15% accuracy on the ADBase. Note that results of different works cannot be compared because the used databases are not the same.

Naming conventions for different numeral systems may be confusing. Digits used in Europe and several other countries sometimes are called "Arabic Numbers"; and digits used in Arab world are sometimes called "Hindi Numbers". A different naming convention is used in this paper. Digits used in Europe will be referred to as "Latin Digits" and that

S. Abdleazeem · E. El-Sherif (✉)
Electronics Engineering Department,
American University in Cairo (AUC), Cairo, Egypt
e-mail: ezzatali@aucegypt.edu

S. Abdleazeem
e-mail: shazeem@aucegypt.edu

Table 1 Arabic printed and handwritten digits

Latin Equivalent	0	1	2	3	4	5	6	7	8	9
Printed	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Typical Handwritten	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Other Writing Style	--	--	--	٣	--	--	--	--	--	--

Table 2 Persian printed and handwritten digits

Latin Equivalent	0	1	2	3	4	5	6	7	8	9
Printed	٠	١	٢	٣	۴	۵	۶	٧	٨	٩
Typical Handwritten	٠	١	٢	٣	۴	۵	۶	٧	٨	٩
Other Writing Style	٠	--	٢	٣	۴	۵	٦	--	--	--

used in Arab world as "Arabic Digits". It is worthwhile to mention here that Arabic and Persian handwritten digits (digits used in Iran) are similar but not identical. However, there are some writing styles for Persian digits that are very similar to Arabic which leads some researchers to consider Arabic and Persian digits to be the same [12,13]. Tables 1 and 2 show Arabic and Persian handwritten digits with different writing styles as well as their printed versions.

In this paper, we are going to study the performance of various classifiers/features combinations on the Arabic digit recognition problem. Well-known feature extraction techniques besides one novel technique we introduce in this paper are considered. Results are then analyzed leading to the notice of the problem introduced by the Arabic digit '0'. A suggestion of how to alleviate this problem is then introduced. For the sake of comparison, the performances of the same classifier/features combinations are evaluated on Latin digits. Moreover, a selection process of a two-stage system is presented and an optimal two-stage system for Arabic digits is suggested.

The remaining of the paper is organized as follows. Section 2 is about the two Arabic digits databases used in this study: the ADBase and the MADBase. Sects. 3 and 4 introduce the classification and the feature extraction techniques used, respectively. Section 5 reports and discusses the results. Section 6 is about the selection process of the optimal two-stage classifier for Arabic digits. And in Sect. 7, we conclude.

2 Arabic digits databases

Both databases (ADBase and MADBase) are available for free online at http://datacenter.aucegypt.edu/shazeem.

2.1 The ADBase

The ADBase [11] is composed of 70,000 digits written by 700 participants. Each participant wrote each digit (from '0' to '9') ten times. The database is partitioned into two sets: a training set (60,000 digits to 6,000 images per class) and a test set (10,000 digits to 1,000 images per class). Writers of training set and test set are exclusive. Ordering of including writers to test set are randomized to make sure that writers of test set are not from single institution (to ensure variability of the test set).

2.2 The MADBase

The MADBase is a modified version of the ADBase that has the same format as MNIST [1]. This is done to ensure the validity of any comparison made between Latin and Arabic digit recognition problems (see Sect. 5 for a comparison between Arabic and Latin digits results). The MADBase is created from ADBase as follows. For each digit of ADBase, its height (h) and width (w) are calculated, and then size-normalized [20] to have a new height (h_{new}) and new width (w_{new}). The assigned values of h_{new} and w_{new} depend on whether h or w is greater. If $h > w$, then h_{new} is set to 20, and w_{new} to floor $(20 \times w/h)$. If $w > h$, then w_{new} is set to 20 and h_{new} to floor $(20 \times h/w)$. This procedure ensures that each digit of MADBase is confined in a 20×20 box, while its aspect ratio is preserved. Then each digit is placed in a 28×28 white background such that its center of gravity coincides with the center of the white background.

Note that the images of the ADBase are binary. When the ADBase images are down-sampled to form the MADBase, an antialiasing filter is applied to the images. This made the images of the MADBase gray-scaled.

Figure 1 shows samples of ADBase and their MADBase versions. In this paper, we evaluate the performance of different classification and feature extraction techniques on MADBase. The ADBase is used just for extracting size information required to alleviate the problem of the Arabic digit '0' as will be clear in Sect. 5.

3 Brief description of the used classification techniques

In this section, a brief description of each of the used classification techniques is going to be presented. In the results section (Sect. 5), the accuracy of each classifier/features combination and the timing performance of each classifier will be reported as well. Some of the used classification techniques have parameters that need to be specified (e.g. number of hidden neurons in the neural network). Such parameters are optimized using a validation set. The validation set is composed of 10,000 samples chosen randomly from the training