Complementary Approaches Built as Web Service for Arabic Handwriting OCR Systems via Amazon Elastic MapReduce (EMR) Model

Hassen Hamdi[1], Maher Khemakhem[2], and Aisha Zaidan[1]
[1]Department of Computer Science, Taibah University, Kingdom of Saudi Arabia
[2]Department of Computer Science, University of King Abdul-Aziz, Kingdom of Saudi Arabia

Abstract: *Arabic Optical Character Recognition (OCR) as Web Services represents a major challenge for handwritten document recognition. A variety of approaches, methods, algorithms and techniques have been proposed in order to build powerful Arabic OCR web services. Unfortunately, these methods could not succeed in achieving this mission in case of large quantity Arabic handwritten documents. Intensive experiments and observations revealed that some of the existing approaches and techniques are complementary and can be combined to improve the recognition rate. Designing and implementing these recent sophisticated complementary approaches and techniques as web services are commonly complex; they require strong computing power to reach an acceptable recognition speed especially in case of large quantity documents. One of the possible solutions to overcome this problem is to benefit from distributed computing architectures such as cloud computing. This paper describes the design and implementation of Arabic Handwriting Recognition as a web service (AHRweb service) based on the complementary approach K-Nearest Neighbor (KNN) /Support Vector Machine (SVM) (K-NN/SVM) via Amazon Elastic Map Reduce (EMR) model. The experiments were conducted on a cloud computing environment with a real large scale handwriting dataset from the Institut Für Nachrichtentechnik (IFN)/ Ecole Nationale d'Ingénieur de Tunis (ENIT) IFN/ENIT database. The J-Sim (Java Simulator) was used as a tool to generate and analyze statistical results. Experimental results show that Amazon Elastic Map Reduce (EMR) model constitutes a very promising framework for enhancing large Arabic Handwriting Recognition (AHR) web service performances.*

Keywords: *Arabic handwriting, complementary approaches and techniques, K-NN/SVM, web service, amazon elastic mapreduce.*

Received April 25, 2015; accepted January 3, 2016

1. Introduction

Web services are client and server applications that communicate over the Web. Web services provide a standard means of interoperating between software applications running on a variety of platforms and frameworks [29].

Optical Character Recognition (OCR) refers to the branch of computer science that allows the translation of optically scanned image of a printed or written text into manageable and editable text. Handwriting Recognition (HR) is a sub-domain of OCR, dealing only with the handwriting part. Arabic writing is one of the most used writings in the world. It is over 1500 years old, written from right to left and more like a drawing than writing [11, 23]. The collaboration of the network and OCR fields is expected to be useful for making lightweight applications called OCRweb service [29, 30].

OCRweb service transforms images into editable text which can be saved, stored, edited or sent via email, SMS or web service. Applications based on the OCRweb service can transform notes, business cards, newspaper clippings, menus and other texts captured via a mobile imaging device or scanned into electronic and editable data that can in turn be easily exported and used into other applications.

Therefore, Arabic Handwriting Recognition (AHR) can be useful to digitalize old historical documents and handwritings of all kinds of texts. Arabic Handwriting Recognition as a web service (AHRweb service) cannot be implemented easily due to the complex morphology of the Arabic script. Recently, we can find more approaches for the Arabic script, but most of the work on Arabic recognition today has been focused on: first small and medium quantity documents and second, on printed documents.

The design and implementation of AHRweb service for large scale Arabic handwriting datasets is an open issue which needs to be addressed. Different approache have been proposed for the pertinent OCR step such us the classification.

As each classification technique has its advantages and shortcomings, we can deduce that the complementary approaches and techniques can provide very good performances and interesting properties as a classifier and present promising tools in building accurate AHRS. These approaches and techniques share the same problem-their high algorithmic

complexity that requires massive storage and processing capacities.One of the proposed solutions for the design and implementation of large-scale AHRweb service is to distribute the web service on advanced distributed architecture such us Cloud computing technology.

In this paper, we are gowing to demonstrate the potential of the Elastic Map Reduce (EMR) model via amazon cloud computing on the big data analytic by providing distributed processing. We are also gowing to discuss how this study can contribute in building accurate, scalable and efficient AHRweb service especially when one needs to combine (collaboration) several strong complementary approaches and techniques to reach a customized recognition rate and speed.

This paper is structured into five sections. First, we are gowing to present an overview of the problem statement wherein the performance of complementary approaches (K-NN, SVM) and its complexity are analyzed. Second, other motivations of our approach are explained. The design and implementation of Arabic Handwriting Recognition as a web service (AHRweb service) based on the complementary approach K-NN/SVM via Amazon EMR model is proposed in section 3. The design of the experiments, experimental results and discussions are presented in section 4. The conclusion and some perspectives are presented in the last section.

2. Problem Statement and Motivation

Easy use, low cost, elasticity, reliability, availability, accuracy, speedup, efficiency are still the major research challenges in AHRweb services when taking into account the quantity and quality of the data. In this section, we will present some issues and motivations behind the design and implementation of an efficient and accurate large scale AHRweb service.

2.1. Complementary K-NN/SVM Classifiers: Performances and Complexity

Large Arabic character recognition system cannot be fulfilled without using suitable classification methods to solve the problem of Arabic handwriting recognition system. Different iterative techniques and algorithms such as data reduction [4], active learning [8], K-means [1], SVM [6, 7], Dynamic Time Warping (DTW) [14, 19], Gaussian Mixture Mode (GMM)[4] have been adopted in classifying step of the AHR process as complementary approaches and techniques. Speeding up the response time of these approaches and techniques constitute their major problem in Arabic handwriting OCR application [18, 24, 25].

Based on some existing research [1] and as in extension of our previous research [9, 15], we have chosen the K-NN and SVM classifiers for several reasons. They have a large capacity of classification and showed their effectiveness in Arabic handwriting recognition.

In this section, we will present the complementary K-NN/SVM classifiers performances and its complexity [2].

K-Nearest Neighbor (K-NN) is an instance based classification method for pattern recognition. Many OCR researchers have found that the K-NN technique achieves a very good performance for Arabic handwriting character recognition in their OCR system on different Arabic datasets [15].

The idea behind k-Nearest Neighbor method is quite simple. To classify a new character, the system finds the k nearest neighbors among the training datasets, and uses the categories of the k-Nearest Neighbor to weight the category candidates.

SVM was applied successfully in different applications such as face detection, face recognition, object detection, object recognition, handwritten character and digit recognition. For more details on this two techniques, Hamdi and Khemakhem [16].

Now consider the potential of combining classifiers for Arabic handwriting [30]. We propose a complementary approach K-NN/SVM similar to Bellili *et al.* [5]. This approach consists on using SVM as a decision classifier to exceed the limits of K-NN (sensible to different classes with similar attributes). Thus, as shown in Figure 1, a handwritten OCR which combines the K-NN and SVM classifiers.

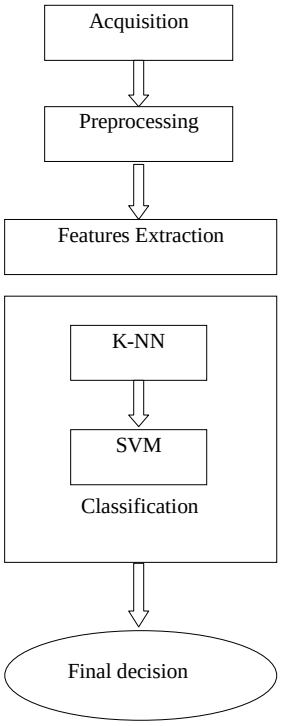

Figure 1. Overview of the proposed handwritten word recognition system.

Unfortunately, the design and implementation of AHR web service based on K-NN/SVM classifier present some problems like the requirement of a large

amount of local memory to store various coefficients and involves complex computations including matrix multiplications and exponential calculations that may limit its practical use in large scale dataset [20, 28, 30, 31].

Indeed in terms of empirical computational complexity, and as illustrated in Table 3. We found that the execution time of K-NN and SVM used individually is better than the hybrid one. This is already expected since the complexity of the hybrid technique as proposed is bounded by the summation of the complexity of both techniques (K-NN and SNM).

2.2. Other Motivations

In many national libraries, archive and research centers, several publications are in the form of books, journals, research papers, conference proceedings, dissertations, Islamic manuscripts and monographs. Most of these documents are still in their original form and only a small amount of it in digital form is available to society. To ease the use of such documents, archive them and make them readable by a bigger audience it is necessary to have them digitalized [3, 14, 22].

Making a Web service based on Arabic handwriting recognition application requires a lot of expertise about network programming and network security. Another motivation, many Arabic HRS developers and researchers are not so familiar with network programming.

Moreover, mobile devices based on mobile phones, PDAs and automobiles used for mobile OCR system are characterized by a limited computational power, memory size and battery life [27]. Thus, the massive and large amounts of data involved in these applications exacerbate the need for a computing framework that offers enough computing and storage capacities which can support large AHRS due to the exponential calculations and the large number of coefficients involved.

Former OCRWeb service provided a few computing frameworks for large Arabic Handwriting recognition system. Representative examples of distributed frameworks include ABBYY Mobile OCR[1], OCRGrid[2] and OCRopus[3].

3. Overview of the Proposed System

In the previous section, we have demonstrated that AHRWeb service based on K-NN/SVM classifier for large Arabic handwriting datasets of regular images cannot work properly on a single personal computer, for many reasons; they do not fit into the corresponding memory, and we cannot process them in a reasonable time.

Our idea is to take advantages of distributed infrastructures, where the limitations of a single computer are overcome by integrating the resources of many computers to perform a large scale AHR process. This approach is not new, since distributed architectures and computing clusters have already existed for several years and have been used in many research activities. Nevertheless, using new technology based on virtual data center such as cloud computing technologies for distributed OCR have recently gained popularity [17].

The basic idea here is, totally, different from those already proposed for Arabic handwriting documents.

Since this paper is focused on using the Amazon EMR model for performing large scale AHR processing, we will describe the proposed system that is based on dividing and merging dataset and processing. That is, instead of combining physical resources like processing power, memory and hard drive storage to allow the processing software to see these combined devices as one entity. In this case, the problem is divided into independent parts which are then processed separately in different physical or virtual node and later joined together to form the output.

To a growing extent, a massive dataset processing, a repetitive and iterative resource intensive IT tasks that need enough power computing and storage capacity, can be out sourced to Service Providers (SP) [16]. These SPs run the task and often provide the results at a lower cost and in short time. A new concept in which computing is offered as a service by third parties where the user is billed only for consumption and use of resource is being introduced. This new service oriented approach from organizations producing a large and massive portfolio of services can be scalable and flexible [3, 26]. Cloud computing technology is the new distributed computing architecture based on the service oriented approach.

The new cloud computing technology is a large pool of systems which are connected in private or public networks, to provide dynamically, scalable infrastructure for application, data and file storage. The growth of this technology affects the cost of computation. Application hosting, content storage and delivery are reduced significantly [13]. Cloud Providers (CP) offer services in three categories. First, Software as a Service (SaaS) where a complete software application is offered to the user on demand. Second, Platform as a Service (PaaS): in this category the users build their own applications, which run on the CPs infrastructure. Finally, the Infrastructure as a Service (IaaS) where CPs provide basic storage and computing capabilities as standardized services over the network. The user would typically deploy his own software on the infrastructure [12].

[1] http://www.abbyy.com
[2] http://www.ocrgrid.org
[3] https://code.google.com/p/ocropus

Although, trying to counter this by running the AHRweb service based on K-NN/SVM classifier on a distributed computing architecture is a solution, it creates a need for task management, mechanisms for data distribution and means to ensure that the AHR process completes even when first, some computers nodes fail during work, second, the network churns and finally, eliminates the need for central administrations and by continuous resource availability. This is the problem that Map Reduce model was designed to solve.

Map Reduce is a programming model developed by Google society for processing, monitoring, managing and generating large datasets used in practice for many real world jobs [23].

This model or framework is based on two main operations Map and Reduce which are applied to the data. These two operations have given the name of the Map-Reduce framework (MR) [10, 24].

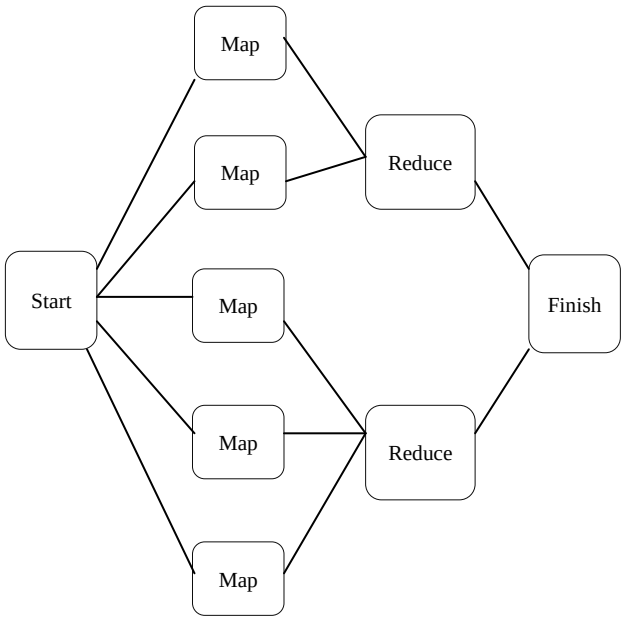

Figure 2. Map-reduce mechanism.

In the Map operation, the Master node takes large problem input and slices it into smaller sub problems then distributes these to worker nodes that may do this and again leads to a multilevel tree structure to process small problems and hands back to master node. In the Reduce operation, the master node takes the answers to the sub problems and combines them in a predefined way to get the output/answer to the original problem. Figure 2 describes the MR framework mechanism.

The suggested approach consists of deploying the AHRweb service based on the complementary approach K-NN/SVM via Amazon EMR model to be run on the amazon cloud computing technologies. Figure 3 describes the suggested system.

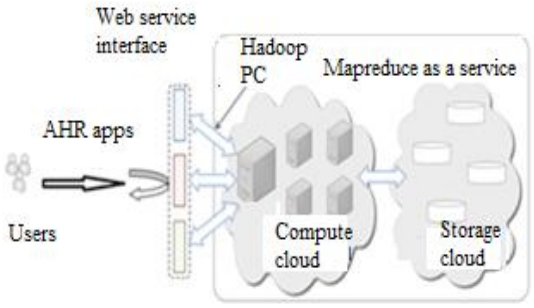

Figure 3. AHRweb service based on the complementary approach K- NN/SVM via Amazon EMR model.

4. Implementation Details

According to the last two IDG Enterprise Cloud Computing providers surveys[4], Amazon Elastic Computing Cloud is selected for the design and implementation of the suggested distributed AHRweb service.

Our developments are implemented with different tools to execute the AHRweb service on Amazon EMR model[5]. First, MapReduce is used as a programming model and as an associated implementation for the processing of massive datasets. Second, we have used Cascading[6]a Java application framework to quickly and easily develop rich Data Analytics and Data Management applications deployed and managed across a cloud computing environment. Third, the Hadoop framework[7] is used to distribute the processing of large datasets across clusters and nodes of computers using simple programming models. This framework processes the input data files by splitting the files in the chunks to process them in parallel. Finally, Amazon Simple Storage Service (Amazon S3)[8] is designed to make web scale computing easier for developers. It is used as a simple web services interface that enables storing and retrieving any amount of data in a pervasive environment (at any time, from anywhere on the web).

In this section, we will present the experimental evaluation of the distributed scheme for a large scale AHRS. First, we will describe the AHRS environment. Second, we will present the experimental setup. And finally, we will present and analyze the experimental results.

4.1. The AHR Process Environment

In this subsection, we will present our AHRS parameters adopted for testing the proposed approach such as the data collection for learning and testing, the different techniques used in the hole AHR process.

[4] http://www.idgenterprise.com
[5] http://aws.amazon.com/elasticmapreduce
[6] http://www.cascading.org
[7] http://hadoop.apache.org
[8] http://aws.amazon.com/s3/

- *Dataset used for learning and testing steps*: The grapheme samples have been generated by a handwritten cursive word segmentation performed on Tunisian handwritten words (town/village names) provided by the Institute of Communications Technology Technical University Braunschweig, (IFN) Germany and the Ecole Nationale d'Ingénieurs de Tunis (ENIT), Tunisia. Therefore, the data base is called IFN/ENIT database [16]. Since graphemes are already preprocessed correspond to only one character (succeeded segmentation). The full test set contains 15000 pages. The 28 Arabic characters written with different scripter in different positions in the word represent all the classes used in our experiments.
- *Techniques used in the AHR process*: The analytical approach is adopted in our OCR system where the dataset IFN/ ENIT was already normalized [16], words are segmented manually and the invariant movement wavelet Transformation was used as a feature extraction technique [31]. Finally, we have used K-NN, SVM and the complementary approach K-NN/SVM as a classifier [29].

The hybrid K-NN/SVM classifer based on Map and Reduce model is illustrated by the pseudo-codes presented below.

Algorithm 1: The hybrid K-NN/SVM classifer based on Map and Reduce model algorithm

Inputs : digit images :Learning Database L D,Testing Database TD, K: Number ok nearest number, SVM.
Output: Intermediary Class Labels KNN, Error Class Labels KNN: ECL-KNN, Intermediary Class Labels SVM: ICL -SVM, Final Predict Class Labels: FPCL.
Begin
1. *Mapper (KNN) Read LD and TD from Hadoop Distributed File System (HDFS).*
2. *Compute the distance between each ldi ∈ LD and tdj ∈ TD.*
3. *Mapper (KNN) output: key-value pairs with key as the test instance ID and the value of the train instance ID and the Euclidean distance is applied netween them.*
4. *Reducer (KNN) : sort the distance and take fisrt K as nearest .*
5. *Reducer (KNN) take majority voting of class labels of K.*
6. *Reducer output ICL-KNN and ECL-KNN.*
7. *Mapper (SVM) Read new TD from ECL-KNN.*
8. *Find out support vectors .*
9. *Merge all calculated SVs and save the resul t to the global SVs*
10. *Mapper (SVM) (apply the function of the golobal SVM).*
11. *Reducer output : and ICL -SVM.*
12. *Merge ICL-KNN and ICL -SVM.*
13. *Reducer output FPCL.*

End

4.2. The Execution Environment

This section describes the basic execution environment of a distributed large scale AHRweb service.
Before describing the execution environment, it is necessary to introduce the concept of jobs. A job is an executable combined with some data and described by a job description. Each job has a numerical identifier, analogous to the Process IDentification (PID). This value is called the Job IDentifier (JID) for short nomination. If the job belongs to an array job, it will also have an Array IDentifier (AID).

The Java based Hadoop Distributed File System (HDFS) is used to provides a scalable and reliable data storage and is designed to span large clusters of commodity servers.

Since the Arabic handwriting data set consists of characters much smaller in size than the standard HDFS block size, the big amount of document to recognize is better to split it into small parts (D1, D2, D3 …Dn) and assign each one to a slave node number to achieve the recognition task. The AHR application based on the complementary approach K-NN/SVM will be implemented by a jobs flow for each node.

The master–slave model and Single Process, Multiple Data (SPMD) technique on distributed memory multiprocessor system is applied to the K-NN/SVM technique as the parallelization technique. In this approach, each copy of the single program runs on processors independently and communication is provided by Hadoup.

For our needs, the free tier of the Amazon Elastic Compute Cloud (Amazon EC2) is enough to deploy the suggested large AHR application. We have allocated 100 cores using the three EC2 Standard Instances with the features described in Table 1.

Table 1. EC2 Standard instances.

Instance type	**m1.small**
1.7 GB of memory, 160 GB of instance storage, and 32-bit platform	
Instance type	**m1.large**
7.5 GB of memory, 850 GB of instance storage, 64-bit platform	
m1.extra large	**m1.extra large**
15 GB of memory, 1690 GB of instance storage and 64-bit platform	

Experimental environment setup is based on two steps: the first step is the setup and configuration of the experimental environment while the second is the submission of the application in a real cloud such as Amazon EC2.

In the first step, we setup the mentioned development environment to develop the application, and then we implement our OCR application and build the executable jar file with Eclipse. Finally, we Run and debug our executable jar file in Cygwin with Hadoop. In the second step, when our application was successfully running in Hadoop, it's time to submit it to Amazon cloud EC2. First we should Sign Up for Amazon S3 to create a bucket using the AWS Management Console were every object in Amazon S3 will be stored. Different techniques of distributed security are presented in the review [21].

Then we upload the data base and the OCR application in the created bucked using client tools of the CloudBerry Explorer for Amazon S3 installed on

the local machine. The third sub step is to create a job flow based on the queueing model applied on Mapreduce model using Cascading by specifying the input and output data, the processing application, the number of EC2 instances and finally, we launch the job flow [11].

Amazon Elastic MapReduce automatically spins up a Hadoop implementation of the AHR framework on Amazon EC2 instances, subdividing the large amount of documents in a job flow into smaller chunks so that they can be processed in parallel, then eventually merging the processed data into the final solution. The Amazon S3 serves as the source for the data being processed, and as the output destination for the end results.

Different running Jobs flow were created on the Amazon Elastic Computing Cloud service. Finally, when the job flow finishes processing the data, we pick up the results from our S3 bucket.

The Figure 4 is from the user guide book Amazon cloud computing, describes the steps of using Amazon Elastic MapReduce[9] to execute the suggested AHRweb service.

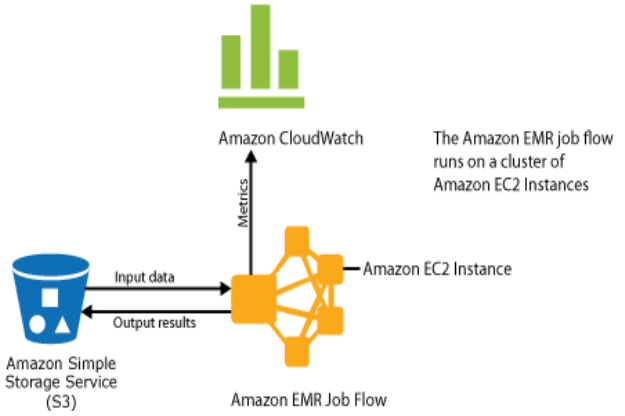

Figure 4. Job flow execution in Amazon MapReduce model.

4.3. Result Evaluation

In this section, first, we will present the K-NN, SVM, and the complementary K-NN/SVM accuracy performance and then we will describe two examples use cases inspired by real large AHR application processing issue. The first one deals with the non distributed (sequential) approach in a local Intel Core i7 CPU 4300 @ 2.70GHz x 2 PC with 8GiB of memory and 64 bit windows operating system, and the second one is the divide-and-merge approach used in splitting the large input dataset into manageable regulars images, to be processed in a distributed manner with the Amazon EMR model. And finally, we will analyze the effects of the Amazon EMR model on the AHRweb service performances. The evaluation result is shown in Tables 2,3,4,5 and Figures 6 and 7.

Table 2. KNN, SVM, KNN/SVM recognition rate (%).

Classes	K-NN (%)	SVM (%)	KNN/SVM(%)
ا	96.15	97.15	97.30
ب	96.20	97.15	97.80
ت	95.10	96.15	97.79
ث	95.40	96.47	96.95
ج	95.20	96.00	97.40
ح	95.30	96.10	97.90
خ	96.50	96.88	97.65
د	96.50	95.10	97.15
ذ	95.10	96.15	97.83
ر	96,80	96,90	97.40
ز	96.17	96.56	97.47
س	95.76	95.95	97.52
ش	94.80	95.10	97.45
ص	94.00	95.10	96.70
ض	95.29	95.80	97.45
ط	95.10	95.45	97.86
ظ	96.10	96.19	97.81
ع	94.90	95.20	97.63
غ	94.00	95.10	97.20
ف	94.90	95.20	97.35
ق	95.18	95.32	96.96
ك	95.48	95.52	96.80
ل	96.00	96.15	97.10
م	95.89	95.98	97.04
ن	96.18	96.67	97.62
ه	96.04	96.29	97.35
و	96.10	96.80	97.14
ي	94.90	94.80	96.15
Average (%)	**95.49**	**95.93**	**97.43**

Table 2 shows that the hybrid K-NN-SVM classifier improves the performance in terms of recognition rate compared with a single K-NN, SVM techniques for Arabic handwriting recognition application (this rate is: 97.43 for complementary approach K-NN/SVM compared to K-NN and SVM with 95.49 and 95.93 respectively). One can also observe that the results obtained using K-NN/SVM for Arabic handwriting recognition is better than existing products [31].

Roughly, the total execution time using K-NN, SVM and K-NN/SVM are approximately 66 hours (16 seconds per image), 69 hours (19 seconds per image), around 3 days and 90 hours, around 4 days to process all 15000 images of the dataset on the non distributed (sequential) approach in a local Intel Core i7 CPU 4300 @ 2.70GHz x 2 PC with 8GiB of memory and 64 bit windows operating system.

This confirms our motivation of distributing the complementary approach K-NN/SVM via Amazon EMR model.

The Table 3 presents the AHRweb service processing time obtained by Amazon EMR model using the large EC2 Standard Instances that make use of 20 to 100 computer nodes.

[9]http://aws.amazon.com/elasticmapreduce/book/

Table 3. Processing time (h) using of K-NN, SVM, and K-NN/SVM on Amazon EMR model using the large EC2 Standard Instances that make use of 20 to 100 computer nodes.

Cores numbers	K-NN (h)					SVM(h)					K-NN/SVM (h)				
	The document size					The document size					The document size				
	3000	6000	9000	12000	15000	3000	6000	9000	12000	15000	3000	6000	9000	12000	15000
20	0.066	0.133	0.199	0.265	0.331	0.073	0.161	0.211	0.265	0.356	0.082	0.172	0.287	0.311	0.360
40	0.055	0.111	0.166	0.221	0.282	0.062	0.135	0.191	0.221	0.311	0.071	0.142	0.211	0.244	0.320
60	0.051	0.101	0.152	0.202	0.251	0.059	0.125	0.162	0.202	0.271	0.060	0.152	0.175	0.211	0.290
80	0.032	0.063	0.095	0.126	0.161	0.042	0.075	0.121	0.126	0.170	0.050	0.086	0.134	0.150	0.211
100	0.021	0.041	0.062	0.082	0.131	0.031	0.054	0.061	0.082	0.145	0.042	0.063	0.076	0.090	0.161

The measured times represent the processing time reported by the Cloud computing technology. By looking at Table 2 which illustrates the accuracy and Table 3 which illustrates the response time, we can conclude that the accuracy of the proposed AHRS is improved and its processing time is reduced significantly by the distributed computing technologies used. We note that this time includes the data replication, task scheduling, hosting machine failures, facilitating communications between machine and consequently everything.

Table 3 shows that our proposed system for large scale AHRS is able to process theoretically more than 40 images in a second. We note that our sequential system takes 22 seconds to process an image, this confirms that our approach decreases in exponential manner the amount of time required to complete the tests.

Before making a web service available for commercial usage, it is important to evaluate the corresponding performances.

As web service involves real services which are difficult to analyze, simulation constitutes an alternate technique for analyzing the efficiency, the scalability of the suggested AHRweb service. The Java based simulation JSIM[10] tools that contain several features was used as a simulation technique to evaluate the AHRweb service performances. We have measured the performance under two factors:

1. Changing the size of the corpus to recognize.
2. Varying number of nodes.

To explain such improvement, we use three factors:

1. The service time (S).
2. The message delay time (M).
3. the waiting time (w) that represent the execution time (T) for each web service (w) that can be described by Equation (1) [11].

$$T_w = S_w + M_w + W_w \quad (1)$$

First, the service time (S) is defined as the time needed to perform an AHR task. Second, the message delay time (M) is the delay taken by the Simple Object Access Protocol (SOAP) messages, in being sent/received by the request. Finally, the waiting time (w) that represents the delay caused by the load on the system where the Web service is deployed on Amazon EMR model with the different number of cores. The evaluation results are shown in Figures 5 and 6.

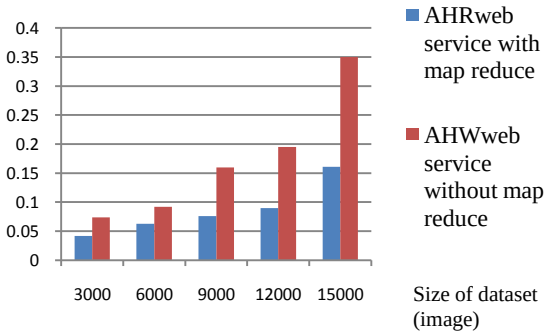

Figure 5. AHRweb service performances response time with and without mapreduce model for diffrent document size.

Figure 5 shows that the response time for AHRweb service without Mapreduce model will increase gradually with the increase in the document size to recognize. But, with Mapreduce, there won't be much increase in the response time when the size of the document increases.

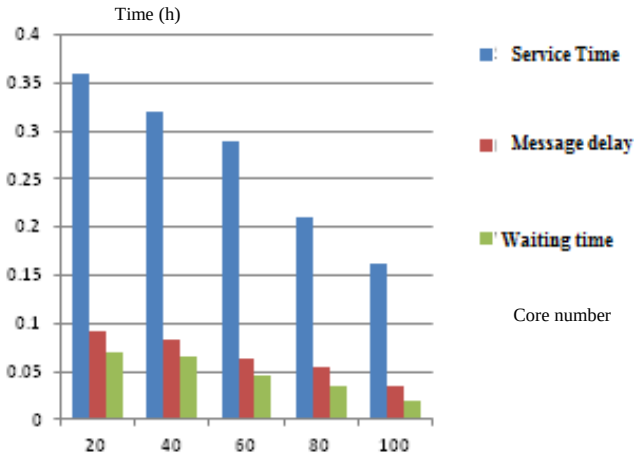

Figure 6. AHRweb service performances using the large EC2 Standard Instances that make use of 20 to 100 computer nodes for a dataset of 15000 images.

[10]http://j-sim.cs.uiuc.edu/

Figure 6 confirms that the execution time of the AHRweb service decreases when the number of cores increases using Amazon EMR. For thus, Amazon EMR model represents an efficient infrastructure in building powerful large AHRS based on the collaboration of complementary approaches since it provides enough computing and storage powers.

In addition to scalability, Availability, reliability, manageability, Cost efficiency are others specific requirements for robust AHRweb service.

The cloud computing paradigm is responsible for the data replication, task scheduling, hosting machine failures, facilitating communication between machine and other tasks. In this paradigm, users have transparent access to a wide variety of distributed infrastructures and platforms. In this distributed environment, computing and data storage necessities are accomplished in different and unanticipated ways to give the user the illusion that the amount of resources is unrestricted. From our tests, using the AHRweb service on MapReduce model via cloud computing technology simplifies the manageability of AHR tasks.

To improve the availability of the suggested AHRweb service, we created different jobs flow (6000 jobs) launched respectively on hadoop node and on amazon mapreduce model. We categorized the completion status of AHRweb service as:

1. Successful web service.
2. Failed web service.
3. Canceled web service which were aborted by the user (Table 4).

Table 4. AHRweb service status with and without Elastic Map Reduce (EMR).

Jobs	Without Map- Reduce model	With Amazon elastic Map- Reduce model
Number of jobs	6000	6000
Successful	4080 (68%)	5850 (97.5%)
Failed	1800 (30 %)	114 (1.9 %)
Canceled	120 (2 %)	36 (0.6 %)

Table 4 shows that there were 1800 and 114 failed jobs respectively without and with Mapreduce model. These success accounted for 97.5 of the total jobs run on Mapreduce model proves that the suggested mapreduce model guaranties the availability for the proposed AHRweb service. Another question to answer is: does the suggested system process a large amounts of distributed data quickly with good response times at minimum cost?

In Amazon cloud computing technology, we pay only for what we use by the hour. This technology allows users to rent resources in this pay-as-you-go environment. This feature offers a cost effective service to process large amounts of distributed data quickly at minimum and efficiency cost, especially when the size of dataset is too huge. In terms of cost, we run a cost efficient Map-Reduce AWS cloud, which can be dynamically extended by renting more instances. The Table 5 presents pricing model of different Amazon EC2 instances.

Table 5. Amazon EC2 instances pricing.

Instance type	Memory (GB)	Storage (GB)	Price per Hour($)
Standard small	1.7	160	0.115
Standard medium	3.7	410	0.23
Standard large	7.5	850	0.46
Standard extra large	15	1690	0.92

As observed in Table 5, by improving the EC2 instances, the price is slightly rinsing.

As discussed above, the Amazon EMR model performs data processing with the available working machine at Amazon clouds computing with an extremely low budget. For thus, the combination of the MapReduce model and the cloud computing is an attractive propositioning for the suggested large AHRweb service so that MapReduce systems can take advantage of massive parallelization in a cloud computing architecture. From our analysis deploying large AHRweb service on a distributed architecture using cloud computing and MapReduce together improve the speed of processing, the availability, the reliability, and decrease consuming cost.

5. Conclusions and Perspectives

In this paper, we have introduced a novel design model to build a powerful Arabic handwriting OCR for large amount of documents. This approach is based on the collaboration of some complementary approaches which can reach a customized recognition rate on building them as web services. Due to the high complexity of these complementary approaches, we have distributed the corresponding web services on a distributed infrastructure in order to speed up the recognition process. For this purpose, we have implemented an Arabic handwriting OCR based on K-NN/SVM over Amazon EMR model.

The obtained results confirm the viability of our proposed approach. Moreover, these results confirm that Amazon EMR model can be adopted and used by other applications that require large amounts of dataset.

The proposed design model requires further investigations. In particular, the most obvious starting point is looking into ways of processing large Arabic dataset. The issues could be solved by using Amazon EMR model that are more suitable for iterative processing.

References

[1] Alhutaish R. and Omar N., "Arabic Text Classification using K-Nearest Neighbour Algorithm," *The International Arab Journal of Information Technology*, vol. 12, no. 2, pp. 190-195, 2015.

[2] AlKhateeb J., Khelifi F., Jiang J., and Ipson S., "A New Approach for Off Line Handwritten Arabic Word Recognition Using K-NN Classifier," *in Proceedings of IEEE International Conference on Signal and Image Processing Applications*, Kuala Lumbour, pp. 191-194, 2009.

[3] Armbrust M., Fox A., Griffith R., Joseph A., Katz R., Konwinski A., Lee G., Patterson D., Rabkin A., Stoica I., and Zaharia M., "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB/EECS-2009-28, 2009.

[4] Belhouari S., Bermak A., Shi M., and Chan P., "Fast and Robust Gas Identication System Using an Integrated Gas Sensor Technology and Gaussian Mixture Models," *IEEE Sensors Journal*, vol. 5, no. 6, pp. 1433-1444, 2005.

[5] Bellili A., Gilloux M., and Gallinari P., "An Hybrid MLP-SVM Handwritten Digit Recognizer," *in Proceedings of 6th International Conference on Document Analysis and Recognition*, Seattle, pp. 28-32, 2001.

[6] Boser B., Guyon I., and Vapnik V., "A Training Algorithm for Optimal Margin Classifiers," *in Proceedings of 5th Annual Workshop on Computational Learning Theory*, Pittsburg, pp. 144-152,1992.

[7] Byun H. and Lee S., "A Survey on Pattern Recognition Applications of Support Vector Machines," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 3, pp. 459-486, 2010.

[8] Cheng J. and Wang K., "Active Learning for Image Retrieval with CoSVM," *Pattern Recognition*, vol. 40, no. 1, pp. 330-334, 2007.

[9] Chow T. and Huang D., Data Reduction for Pattern Recognition and Data Analysis, *Computational Intelligence: A Compendium*, pp. 81-109, Springer, 2008.

[10] Dean J. and Ghemawat S., "Mapreduce: Simplified Data Processing on Large Clusters," *Communications of the ACM-50th Anniversary Issue: 1958*, vol. 51, no. 1, pp. 107-113, 2008.

[11] Eken S. and Sayar A., "Big Data Frameworks for Efficient Range Queries to Extract Interested Rectangular Sub Regions," *International Journal of Computer Applications*, vol. 119, no. 22, pp. 36-39, 2015.

[12] Foster I., Zhao Y., Raicu I., and Lu S., "Cloud Computing and Grid Computing 360- Degree Compared," *in Proceedings of Grid Computing Environments Workshop*, Austin, pp. 1-10, 2008.

[13] Goto H., "An Overview of the WeOCR System and a Survey of its Use," *in Proceedings of Image and Vision Computing*, Hamilton, pp. 121-125, 2007.

[14] Ha K., Chen Z., Hu W., Richter W., Pillaiy P., and Satyanarayanan M., "Towards Wearable Cognitive Assistance," *in Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, Bretton Woods, pp. 68-81, 2013.

[15] Hamdi H. and Khemakhem M., "Arabic Islamic Manuscripts Digitization Based on Hybrid K-NN/SVM Approach and Cloud Computing Technologies," *in Proceedings of International Conference on Advances in Information Technology for the Holy Quran and Its Sciences*, Medina, pp. 366-371, 2013.

[16] Hamdi H. and Khemakhem M., "A Comparative Study of Arabic Handwritten Characters Invariant Feature," *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 12, pp. 62-68, 2011.

[17] Jain A., Duin R., and Mao J., "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence Transactions. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, 2000.

[18] Khemakhem M. and Belghith A., "Towards A Distributed Arabic OCR Based on the DTW Algorithm: Performance Analysis," *The International Arab Journal of Information Technology*, vol. 6, no. 2, pp. 153-161, 2009.

[19] Khemakhem M. and Belghith A., Towards Distributed Cursive Writing OCR Systems Based on the Combination of Complementary Approaches, *Guide to OCR for Arabic Scripts*, Springer London, pp. 351-371, 2012.

[20] Milgram J., Cheriet M., and Sabourin R., "Speeding Up the Decision Making of Support Vector Classifiers," *in Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, Kokubunji, pp. 57-62, 2004.

[21] Rashvand H., Salah K., Calero J., and Harn L., "Distributed Security for Multiagent Systems Review and Applications," *IET Information Security*, vol. 4, no. 4, pp.188-201, 2010.

[22] Roper G., *World Survey of Islamic Manuscripts*, Al-Furqān Islamic Heritage Foundation, 1992.

[23] Sala K. and Calero J., "Achieving Elasticity for Cloud MapReduce Jobs," *in Proceeding of 2nd IEEE International Conference on Cloud Networking*, San Francisco, pp. 195-199, 2013.

[24] Sergios T. and Koutroumbas K., *Pattern Recognition*, Elsevier, 2006.

[25] Shi M., Bermak A., Chandrasekaran S., and Amira A., "An Eficient FPGA Implementation of Gaussian Mixture Models Based Classifer Using Distributed Arithmetic," *in Proceedings of 13th IEEE International Conference on Electronics, Circuits and Systems*, Nice, pp. 1276-1279, 2006.

[26] Singh S., "Current Trends in Cloud Computing A Survey of Cloud Computing Systems," *International Journal of Electronics and Computer Science Engineering*, vol. 1, no. 3, pp.

1214-1219, 2011.
[27] Srihari S. and Ball G., "Statistical Characterization of Handwriting Characteristics using Automated Tools," *in Proceedings of SPIE -The International Society for Optical Engineering*, San Jose, pp.1-10, 2011.
[28] Srihari S., "Handwriting Address Interpretation: a Task of Many Pattern Recognition Problem," *International Journal of Pattern Recognition and Articial Intelligence*, vol. 14, no. 5, pp. 663-674, 2000.
[29] Verma B., "A Contour Code Feature Based Segmentation For Handwriting Recognition," *in Proceedings of the 7th International Conference on Document Analysis and Recognition*, Edinburgh, pp. 1203-1207, 2003.
[30] Zanchettin C., Bezerra B ., and Andrade V., "AK-NN-SVM Hybrid Model for Cursive Handwriting Recognition," *in Proceedings of WCCI 2012 IEEE World Congress on Computational Intelligence*, Brisbane, pp. 1-8, 2012.
[31] Zangeneh I., Moradi M., and Mokhtarbaf A., "The Comparison of Data Replication in Distributed Systems," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 5, no. 11, pp. 1183-1185, 2011.

Hassen Hamdi received in 2008 Masters Degree in Computer Science from the University of Sfax, Tunisia. He is currently Lecturer of Computer Science at the Faculty of Computing and Information Technology at Taibah University, Saudi Arabia and doing his PhD research at the Multimedia, InfoRmation Systems and Advanced Computing Laboratory University of Sfax, Tunisia. His research interests include Arabic OCR, distributed systems, performance analysis, and networks security.

Maher Khemakhem received his Master of Science, his Ph.D. and Habilitation accreditation degrees, respectively, from the University of Paris11 (Paris Sud, Orsay), France, in 1984, 1987 and the University of Sfax, Tunisia, in 2008. He is currently Associate Professor of Computer Science at the Faculty of Computing and Information Technology at King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include Arabic OCR, distributed systems, performance analysis, and networks security.

Aisha Zaidan received her BSc in Computer Information Systems from Zarqa Private University in 2004, and her MSc in Computer Science from Jordan University of Science and Technology in 2012. She is currently working as a lecturer at Taibah University. Her area of interests includes web applications, data mining, and virtual reality.