Toward a Distributed Benchmarking Tool For Biometrics

Julien Mahier, Mohamad El-Abed, Baptiste Hemery, Christophe Rosenberger

Toward a Distributed Benchmarking Tool For Biometrics

Julien Mahier, Mohamad El-Abed, Baptiste Hemery, and ChristopheRosenberger
Université de CaenBasse-Normandie, UMR 6072 GREYC, F-14032 Caen, France
ENSICAEN, UMR 6072 GREYC, F-14050 Caen, France
CNRS, UMR 6072 GREYC, F-14032 Caen, France
{*julien.mahier, mohamad.elabed, baptiste.hemery, christophe.rosenberger*}*@ensicaen.fr*

ABSTRACT

Research in computer science evolves very quickly. In order to prove the efficiency of a new algorithm, it is generally necessary to show some results on a large and significant benchmark used in the state of the art. This fact has for consequence the need of a large computation capability in a research laboratory. We address in this paper the performance evaluation of biometric systems through distributed computing. In this domain, we need to realize with the same algorithm many computations on different data (generally corresponding to images). We propose a solution based on a software we developed for facilitating this task. The proposed architecture is composed of a server distributing computation tasks on all the available clients. Experimental results show the benefit of the proposed software.

KEYWORDS: Benchmark, Distributed Computing, Performance Evaluation, Biometrics.

1. INTRODUCTION

Biometrics is an emerging technology in the general field of computer security for user authentication. Despite the obvious advantages of this technology in enhancing and facilitating the authentication process, its proliferation is still not as much as attended [1], [2]. By contrast to traditional methods, biometric systems do not provide a 100% reliable answer, and it is quite impossible to obtain such a response. This uncertainty is due from the variations of human characteristics (e.g., occlusions in Iris-based recognition systems), environment factors (i.e., variation of acquisition conditions such as illuminations in facial-based recognition systems) and cross-device matching [3].

The performance evaluation of such systems is very important as the associated security depends on the possible errors that could appear. In general, a biometric system is characterized by two performance metrics: False Acceptation rate (FAR), which corresponds to the number of impostors that are authenticated by the system, and the False Rejection Rate (FRR), which measures the number of genuine users that are rejected [4]. These two evaluation metrics are computed given a threshold value set by the system administrator; This threshold defines if the user's identity based on the provided biometric information is verified. The performance evaluation of a biometric system is generally defined by a Receiver Operating Curve (ROC) while plotting FAR versus FRR for different values of the threshold.

In order to have a reliable judgment of the performance of a biometric system, we need to use a benchmark database containing some biometric features for a large number of users [5], [6]. If we have a database of N individuals and M samples for each user, the definition of the FRR value requires (M-1).N computations and for the FAR value N.(N-1)*M ones (for a single enrollment biometric system). Actually, in the biometric field, the objective is to maximize at the same time N and M in order to make a reliable comparison of matching algorithms. For an industrial use, N could be 1 million of users (i.e. border control application). For a research lab, the need of a large computation capability for testing biometric algorithms is extremely important. A researcher who wants to propose a new matching algorithm must compute many results on a large benchmark but also has to compare them with other methods from the state of the art.

As the financial capabilities of research labs are limited and the access to clusters not necessary easy and possible, some solutions based on distributed computing must be

developed. There is a strong need of a simple solution for distributing computational tasks on classical computers. In the biometrics field, this distribution is easier as the computation tasks can be intrinsically parallelized. Indeed, in the authentication case, we have to compute for each pair (reference, capture) a matching result to generate FAR, and FRR values for plotting the ROC curve.

The outline of this paper is described below. We make an overview of existing methods for benchmarking in the field of biometrics in the next section. Section III describes the proposed solution in order to solve this problem. We highlight some validation elements of the proposed system in section IV. We then conclude and give some perspectives of this study.

2. RELATED WORKS

The evaluation of biometric systems is now carefully considered in the research in biometrics. Nowadays, many efforts have been done to evaluate such systems. We present in this section an overview of the performance metrics, the research platforms and benchmarks in biometrics as an illustration of the evaluation methodologies used in the literature for the comparison of biometric systems.

2.1. Biometric Performance Metrics

The comparison result between the acquired biometric sample and the corresponding stored reference (also called Template) is a similarity score. If the score is higher than the predefined decision threshold, then the system accepts the claim user, otherwise the claim is rejected. This threshold is defined according to the security level required by the application. Figure 1 illustrates the distribution of the genuine users and impostor scores.

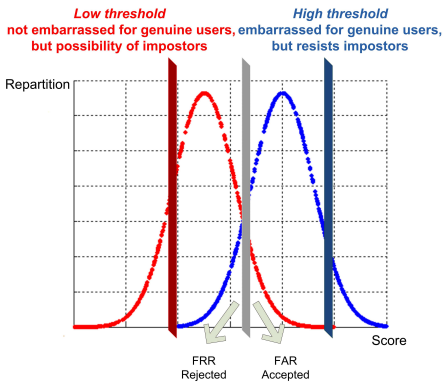


Figure 1. Distribution of Genuine and Impostor Scores

Here are different biometric common terms defined by the International Organization for Standardization ISO/IEC 19795-1 [6]. They mainly concern evaluation metrics that are used to characterize the performance of a biometric system:

- Failure-to-enroll rate (FTE): proportion of the user population for whom the biometric system fails to capture or extract usable information from biometric sample;
- Failure-to-acquire rate (FTA): proportion of verification or identification attempts for which a biometric system is unable to capture a sample or locate an image or signal of sufficient quality;
- False acceptation rate (FAR): proportion of impostors that are accepted by the biometric system.
- False rejection rate (FRR): proportion of authentic users that are incorrectly denied.
- False-match-rate (FMR): the rate for incorrect positive matches by the matching algorithm for single template comparison attempts. FMR equals FAR when the biometric system uses one attempt by a user to match its own stored template;
- False-non-match rate (FNMR): the rate for incorrect negative matches by the matching algorithm for single template comparison attempts. FNMR equals FRR when the biometric system uses one attempt by a user to match its own stored template;
- Equal error rate (EER): it is the value where both errors rates, FAR and FRR, are equals (i.e., FAR = FRR). It constitutes a good indicator, and the most used, to evaluate and compare biometric systems. In other words, lower the EER, higher the accuracy of the system.

All those rates may be drawn to graphically visualize the accuracy of a biometric system. For example, the ROC (Receiver Operating Characteristics) curve plots the FRR versus the FAR as illustrated in Figure 2. It is mainly used to evaluate and compare the performance of biometric systems.

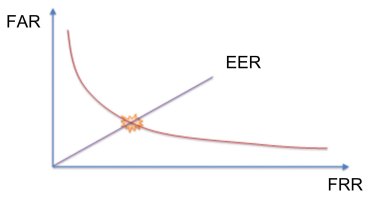


Figure 2. ROC curve

These evaluation metrics are generally computed based on data from benchmarks.

2.2. Benchmarks

A benchmark allows researchers to test their algorithm and compare them with those from the state of the art. It takes a lot of time and energy to build a large and significant benchmark. It is very convenient to download one for research purposes. We present in this section an overview of several benchmarks in the state of the art.

- GREYC alpha [7] is a keystroke dynamics database that we have collected using GREYC keystroke software developed in our research laboratory. The database is composed of 133 individuals, by typing between 5 and 107 times the password "greyc laboratory" between 03/18/2009 and 07/05/2009. We have 7555 available captures, and the average number of acquisitions per user is 51 with 100 of them having more than 60 templates. Most of the individuals have participated at least to 5 sessions. Both the software and the collected database are publicly available on our website.
- FERET database [8], [9] is a facial database composed of 725 individuals with from 5 to 91 samples per individual (the average value is 11). Each sample corresponds to a pose angle, illumination and expression.
- BioSecure database [10] is a set of databases collected by 11 university institutes across Europe in the framework of the BioSecure Network of Excellence. It contains data for face, voice, iris, fingerprint, hand and signature modalities, within the framework of three datasets corresponding to real multi-modal, multi-session and multi-environment situations. The databases are requested through the BioSecure website.

To help researchers to process these data and to quantify the performance evaluation of their algorithms, some platforms have been created.

2.3. Platforms

Different platforms have been established for enhancing the widespread of use of biometric systems. All the evaluations have been done using a predefined database and protocol.

- FVC-onGoing, On-Line Evaluation of Fingerprint Recognition Algorithms: FVC-onGoing is the evolution of the FVC international competitions held on 2000, 2002, 2004 and 2006. It is a web-based automated evaluation for fingerprint recognition algorithms available on https://biolab.csr.unibo.it/FVCOnGoing/. It uses a set a set of sequestered datasets using well known performance indicators and metrics (such as EER).
- BioEVA Tool: BioEVA [11] is a tool that allows testing and evaluating the performance of biometric algorithms. It contains three modules: enrollment, authentication and evaluation. The tool receives a biometric algorithm as a "black-box", and uses some metrics called quality parameters (such as ROC curves) to quantify its performance. Two algorithms based on static signatures and one based on keystroke dynamics were evaluated using bioEVA.
- BioSecure Reference and Evaluation framework: BioSecure (http://biosecure.it-sudparis.eu/) is a project of the 6th Framework Programme of the European Community. Its main objective is to build and provide a common evaluation framework, which investigates and compares the biometrics-based identity authentication methods. It provides twelve benchmarking reference systems (available on: http://share.int-evry.fr/svnview-eph/): 2D face, 3D face, Fingerprint, Hand, Iris, Signature, Speech and Talking-face reference systems. These reference systems are made of replaceable modules (preprocessing, feature extraction, model building and matching) which allow developers and researchers to investigate the improvement of a specific part of the system. In this case, a researcher can evaluate and compare its matching algorithm just by replacing the corresponding module in the reference system. An example of the used performance metrics are the distributions of intra and inter scores, resulting from genuine and impostor comparisons, respectively.

2.4. Discussion

As shown in the previous subsections, many laboratories have proposed biometric platforms whose objective is mainly to compare enrollment and verification/identification algorithms in the state of the art. Multiple metrics are used within this context [6]. These statistical measures allow in general a precise performance characterization of a biometric system. As argued in the introduction section, the computation complexity in terms of time and required materials, limit researchers contributions in this research field. In order to resolve such a problematic, we propose in this paper a software-based solution aiming to parallelize computation tasks. Such kind of solution would facilitate and decrease the time consumption of biometric performance metrics.

We present in the next section our contribution.

3. DEVELOPED METHOD

Before presenting the proposed solution, we define the objectives and constraints of the work.

3.1. Objectives and Constraints

Biometric systems use a wide range of different technologies, platforms and algorithms. An unbiased and reliable evaluation process of such systems should be applied for each algorithm. The more accurate the evaluation is, the more computation needs to be done.

Furthermore, as long as the different ISO standards are not applied, dealing with different kinds of biometric systems implies dealing with different technologies, different databases and different evaluation process. This fact increases the researcher's work in terms of technology integration instead of evaluation. The proposed computation platform focuses on the work of the researcher and proposes a way to ease the integration process in a pool of computation machines.

The constraints of the proposed solution are: low cost, ease of use, ease of maintenance, language independence (C, C++, C#, Java, Python ...), scalability.

3.2. Architectures

The solution is based on a client-server architecture. The server centralizes all the different computation requests coming from the user and some client machines (named hosts) which compute the different tasks (computation requests). Moreover, the architecture relies on an interface for managing the system (named managing client), a data repository (shared directories) and configuration files in XML format. Figure 3 presents the global architecture of the solution.

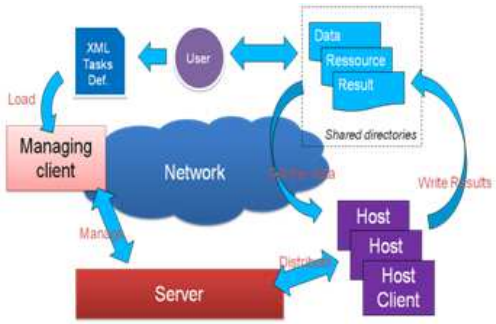


Figure 3. Principle of the Solution

The core of the system lies on the computation requests. Each request is represented with 3 parameters:

- The first one is the data source on which the whole system has to deal with. The data are stored on the network within a shared directory. This way, the application does not have manage and deploy the data to the client.
- The second is the binary application, developed by the researcher which can execute one task. The application is launched by a program called "HostClient" deployed on some production machine (hosts) in the laboratory (server or workstation). The server manages the execution of the application on the HostClient. The algorithm executed on the server is a simple FIFO. HostClients and server communicate asynchronously.
- The third one is a shared repository where the results can be stored. Below is the XML representation of a computation request.

```
<TaskPools>
 <TaskPool>
  <Task>
   <ExecutionCommand>...</ExecutionCommand>
   <ParameterList>
     <Parameter>...<Parameter>
   </ParameterList>
  </Task>
 </TaskPool>
</TaskPools>
```

A task represents an atomic action to apply on the data source. The execution parameters and the execution command set up the application before computation on a client host. TaskPool represents all the computations to realize on a data set. TaskPools represents all the different works to compute on the distributed system.

3.3. Use Case

When a user wants to benchmark a solution or an algorithm, he/she first has to implement an atomic operation. This development is aimed by three simple rules. First, it must be a stand alone program, without any software installation. Second, the application and the data must be deployed on the repository. Last but not least, the application should get its parameters from command line options. A sample of parameters could be the name of the image source and the destination of the results. After developing the application, the user has to complete the XML representation of the computation. He/She then first has to be sure that the server application is running. We can see the server running in Figure 4.
The user connects to the server using the Managing Client Application and sends the XML file containing the whole task pools. The tasks are then queued on the server waiting for a client to execute them. Figure 5 shows the management client and the state of a computation task pool. Each user has access to the Managing Client Application, so they can see if there is computation work in progress.
When a client executes a task, it informs the server of

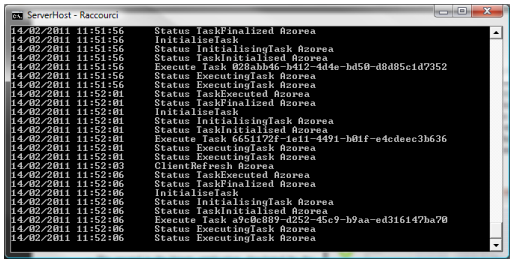


Figure 4. Server

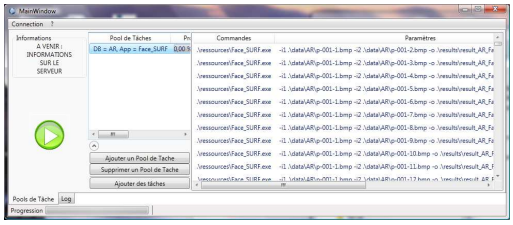


Figure 5. Managing Client

the success or the fail of the task. Each client can see the current task being executed on the computer. We can see the client application in Fig. 6.

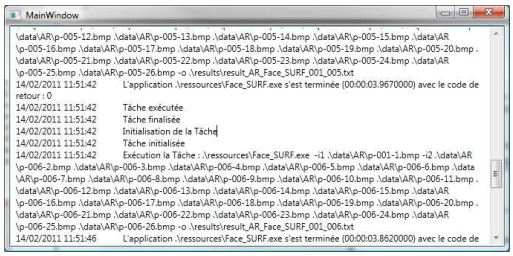


Figure 6. Client

3.4. Performance Consideration

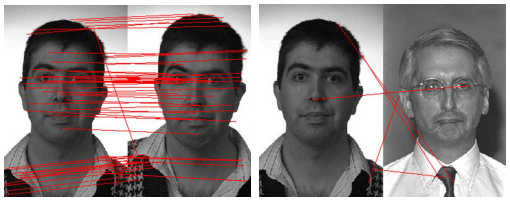
The whole system is aimed to be hosted on a production network which is not dedicated to computation. In consequence, the client and server could not be assigned on specific machines but on some existing ones. Thus, depending on the work activity of the server, on the network and on the laboratory, data computation results time may vary. Furthermore, some of the workstations which host the application could be offline or deactivated.

4. EXPERIMENTAL RESULTS

We tested our benchmarking solution with a face recognition algorithm. The protocol and obtained results are presented below.

4.1. Protocol

In order to make a decision on the identity of a user, we compare a reference image with a captured one. The used algorithm for face recognition uses SURF keypoints [12] detected on an image. We first compute these keypoints on the two images that we want to compare. Each SURF keypoint is characterized by a vector in 64 dimensions. We then compute the similarity of the two individual by looking at similar keypoints in the two images. The matching process is the same as the one presented in [13]. The more keypoints are associated between the two images, the more confident we are that the two images correspond to the same individual. An example of a genuine user can be seen in Figure 7(a) and an impostor user can be seen in Figure 7(b). Red lines correspond to a match between two associated keypoints.



(a) Genuine User (b) Impostor User

Figure 7. Two Face Authentication

In order to evaluate our face recognition algorithm, we use a part of the AR face database [14] containing 30 individuals. Each individual has 26 images, the first one corresponds to the reference face whereas the 25 remaining ones correspond to test images. In order to evaluate the algorithm on this part of the database, we need to compare each test image to each reference image. This leads to 25*30 = 750 genuine comparisons and 25*29*30 = 21.750 impostor comparisons. The total number of comparisons is then equal to 22.500.

In order to evaluate the benchmarking tool, we made two experiments. The first one aims to check the influence of the task size. We defined two different XML files: for the first one, each comparison is assigned to a task. This leads to the definition of an XML file that contains 22.500 tasks. For the second one, each task contains a comparison for one individual to one reference image, that is to say 25 comparisons per task and 900 tasks. For the second experiment, we try to evaluate the performance regarding the number of clients that are used.

4.2. Results

For the first experiment, we run the Server, the Managing client and the client on the same computer. For the first

XML file, the computation time for each task is around 0.3 second long. The total experiment lasts 355 minutes. For the second XML file, each task required 3.5 seconds and the total experiment lasts 57 minutes. We conclude that each task must be long enough to compensate the time used for the communication between the server and the client. We use the second XML file for the second experiment. We then try to change the number of client that are considered. The results of this second experiment are presented in Table 1.

Table 1. Result for the Second Experiment

Number of client	Time
1 (Local)	57m39s
1+1 (Local)	30m01s
1	70m27s
2	51m07s
3	32m19s
4	25m48s
4+4	14m07s

First, we can see that using two clients (line 1+1) on the same computer increases the performance. This is due to the fact that the computer used is a dual core processor and each task is single threaded. Second, we can see that, when using only one client, it is faster when the client is on the same computer as the server. This small lost is due to the network communication latency. However, we can use many more computers as clients. We can see that using 3 or more clients brings significant improvements concerning the execution time.

5. CONCLUSIONS AND PERSPECTIVES

Benchmarking is an important issue in research in computer science and especially in biometrics. We propose in this paper a simple solution to distribute some computation tasks on different clients by using an ergonomic software. We showed that this kind of solution has many advantages for researchers. The proposed solution permits to use any computers in a research laboratory to be used for the performance evaluation of new biometric systems. We plan to provide in the future this platform for the research community in the biometrics domain.

Considering the current state of the application, several improvements could be done. Assuming we are in a private network and we have high privileges, the first one is to bypass the HostClient application and to execute the researcher application directly on the client machine. The second one is to add Linux compatibility. The existing one is based on Microsoft.net and is not compatible with Linux. The researcher could develop his algorithm for Windows or Linux and tell the server which machine to use. The third one is a web-based management client. Last, we also would like to test the proposed platform on a large cluster to quantify its benefit on very large benchmarks.

REFERENCES

[1] A. K. Jain, L. Hong, and S. Pankanti, "Biometrics: Promising frontiers for emerging identification market," Department of Computer Science, Michigan State University, Tech. Rep., 2000.

[2] A. K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross, "Biometrics: A grand challenge," *Pattern Recognition, International Conference*, 2004.

[3] N. Poh, J. Kittler, and T. Bourlai, "Quality-based score normalization with device qualitative information for multimodal biometric fusion," *SMC-A*, 2010.

[4] F. Cherifi, B. Hemery, R. Giot, M. Pasquet, and C. Rosenberger, "Performance evaluation of behavioral biometric systems," in *Behavioral Biometrics for Human Identification: Intelligent Applications*, 2009.

[5] A. J. Mansfield and J. Wayman, "Best practices in testing and reporting performance of biometric devices," BWG, Tech. Rep., 2002.

[6] "Information technology biometric performance testing and reporting," ISO/IEC 19795-1, Tech. Rep., 2006.

[7] R. Giot, M. El-Abed, and C. Rosenberger, "Greyc keystroke : a benchmark for keystroke dynamics biometric systems," in *IEEE Third International Conference on Biometrics : Theory, Applications and Systems (BTAS)*, 2009.

[8] P. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Journal of Image and Vision Computing*, 1998.

[9] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face recognition algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000.

[10] "Biosecure Multimodal Biometric Database," http://www.biosecure.info/, 2008.

[11] L. Sucupira, L. Araujo, M. Lizarraga, and L. Ling, "Bio-EVA: An evaluation tool for biometric algorithms," in *ICBA04*, 2004, pp. 716–723.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, pp. 346–359, 2008.

[13] B. Hemery, J.-J. Schwartzman, and C. Rosenberger, "Study on color spaces for single image enrolment face authentication," in *IAPR International Conference on Pattern Recognition (ICPR)*, 2010.

[14] A. Martinez and R. Benavente, "The AR face database," *CVC Tech. Report*, 1998.