Real Time Web-based Toolbox for Computer Vision

Sidi Ahmed Mahmoudi
Computer Science Department,
University of Mons,
Belgium
-----
sidi.mahmoudi@umons.ac.be

Mohammed Amin Belarbi
Faculty of Exact Sciences and Computer Science,
University of Mostaganem,
Algeria
-----
belarbi_mohammed_amin@yahoo.fr

Mohammed El Adoui
Computer Science Department,
University of Mons,
Belgium
-----
mohammed.eladoui@umons.ac.be

Mohammed Amine Larhmam
Computer Science Department,
University of Mons,
Belgium
-----
mohamedamine.larhmam@umons.ac.be

Fabian Lecron
Computer Science Department,
University of Mons,
Belgium
-----
fabian.lecron@umons.ac.be

ABSTRACT

The last few years have been strongly marked by the presence of multimedia data (images and videos) in our everyday lives. These data are characterized by a fast frequency of creation and sharing since images and videos can come from different devices such as cameras, smartphones or drones. The latter are generally used to illustrate objects in different situations (airports, hospitals, public areas, sport games, etc.). As result, image and video processing algorithms have got increasing importance for several computer vision applications such as motion tracking, event detection and recognition, multimedia indexation and medical computer-aided diagnosis methods. In this paper, we propose a real time cloud-based toolbox (platform) for computer vision applications. This platform integrates a toolbox of image and video processing algorithms that can be run in real time and in a secure way. The related libraries and hardware drivers are automatically integrated and configured in order to offer to users an access to the different algorithms without the need to download, install and configure software or hardware. Moreover, the platform offers the access to the integrated applications from multiple users thanks to the use of Docker (Merkel, 2014) containers and images.

Experimentations were conducted within three kinds of algorithms: 1. image processing toolbox. 2. Video processing toolbox. 3. 3D medical methods such as computer-aided diagnosis for scoliosis and osteoporosis. These experimentations demonstrated the interest of our platform for sharing our scientific contributions related to computer vision domain. The scientific researchers could be able to develop and share easily their applications fastly and in a safe way.

KEYWORDS

Cloud computing; real time; GPU; heterogeneous architectures; image and video processing; medical Imaging.

ARTICLE INFO

Received: 19 February 2018
Accepted: 12 June 2018
Published: 30 June 2018

https://dx.doi.org/10.7559/citarj.v10i2.494

1 | INTRODUCTION

In the last few years, a great evolution of processors has been noticed due to the change of central processing units (CPUs) architectures, which offer actually a high computation power. The latter is obtained as a result of increasing the computing units' number within processors. This evolution is present in different kinds of super calculators: grid, cluster, GPU, Multi-CPU/Multi-GPU machines, etc.

In this context, several computer vision algorithms and applications tend to benefit from this high computing power by the development of parallel implementations that exploits the high number of processing units. Notice that image and video processing algorithms are well adapted for parallelization within Multi-CPU or/and Multi-GPU platforms (Mahmoudi S. A., 2014) since they consist mainly of a common computation over many pixels. Several parallel solutions that exploit the above-mentioned hardware have been developed recently. Although they offer a great potential of processors (Multi-CPU or/and Multi-GPU, cluster, grid, etc.), the use, configuration and exploitation of these solutions in not so easy.

Indeed, users must have the required hardware and need to download, install and configure the related CPU or/and GPU libraries. Therefore, we propose a cloud-based platform that integrates image and video processing toolbox on the one hand and 3D medical computer-aided diagnosis for osteoporosis and scoliosis diseases on the other hand. Each connected guest or user (to our platform) can select the required algorithm (from the toolbox) or the medical application, load its data and retrieve results with an environment similar to desktop either if the required application exploit parallel (GPU) or heterogeneous (Multi-CPU/Multi-GPU) platforms.

The integrated algorithms and applications can be executed in real time and in a secure way. The related libraries and hardware drivers are automatically integrated and configured in order to offer to users an access to the different algorithms without the need to download, install and configure software and hardware. Moreover, the platform offers the access to the integrated application from multiple users thanks to the use of Docker (Merkel, 2014) containers and images. In terms of security, each connection is secured within SSL protocol, the exchanged data are secured within SFTP protocol and encrypted within an efficient method of data compression. Another key protects the encryption key, and data are duplicated in two virtual machines within Hadoop (Shvachko, 2010) framework in order to keep a backup of our data in case of crash.

The remainder of the paper is organized as follows: Section 2 presents the related works, while the third Section is devoted to detail our real time cloud-based platform and toolbox for computer vision. Experimental results, that present several use cases and applications, integrated in our platform, are given in Section 4. Finally, conclusions and future works are discussed in the last section.

2 | RELATED WORKS

In literature, we can cite several works related to our work of applying computer vision algorithms and applications within cloud platforms in real time. These related works can be presented with three subsections: 1. Computer vision cloud platforms. 2. Web development tools. 3. Security within cloud platforms.

2.1 COMPUTER VISION CLOUD PLATFORMS

Recent development of computer vision platforms have been significantly influenced by the emergence of a growing number and accessible cloud computing platforms hosted by large-scale IT-companies (AWS, GCP, Azure). They enabled the development of a variety of cloud interfaces, which makes abstraction on the complexity behind computer vision application. The latter use a specific workflow for cloud architectures (Calasanz, 2017) which gives access to a high computing power without the need of a low-level software programming or any hardware adaptation.

CloudCV (Agrawal, 2015) is an example of a cloud-based and distributed computer vision platform composed of three parts; an AI-as-a-service platform that enables researchers to easily convert their deep learning models to web service and call them by a simple API, a drag and drop collaborative platform for building models, an evaluation server for comparing different AI and computer vision algorithms (for example for challenges). CloudCV provides access to two API (Python and Matlab), and englobes multiple modern components for its backend architecture such as OpenCV, Caffe, Turi (GraphLab).

Image processing On Line (IPOL) (Limare, 2011) platform provides image processing algorithms and descriptions, source code and a handy web interface to check results from new input images. This initiative intended to promote reproducible research related to image processing.

Other cloud commercial applications such as Face API (by Microsoft Azure), Amazon Rekognition, Watson Visual Recognition (by IBM) or Clarifai, deliver APIs, which focus on specific computer vision tasks for image and video understanding.

With the exponential grow of image data, solving big data challenge became an important task managed on the cloud. Yan et al. (Yan, 2014) proposed a cloud architecture dedicated to large scale image processing based on Hadoop. They evaluated the performance of the platform using different image processing algorithms. They reported some issues with data distribution and cluster resource related to the use of Hadoop.

Dealing with massive data that requires an efficient exploitation of high-performance computing (HPC) resources. When it comes to cloud architecture this task becomes more challenging. In a recent work, Netto et al (Netto, 2017) addressed in a survey the challenges of HPC cloud for Scientific and business applications. They reported different aspects to take in account for performance optimization of HPC in the cloud including; Scheduler planning which affects the performance of jobs, Platform selector, Spot instance related to the pricing model, Elasticity for dynamic resource adding and finally Predictors for future resource consumption.

2.2 WEB DEVELOPMENT TOOLS

In the early days of internet, web development encompasses many different challenges, including graphic design, database interaction, etc.

Thanks to its stability, PHP presents one of the most popular programming languages used by the web developers in the world. However, to make development faster, it is recommended to use PHP frameworks. For example, using one of these frameworks, we do not need to write complex queries to retrieve data from database. In fact, the frameworks provide a prepared operation such as 'Create', 'Read', 'Update', and 'Delete'. On the other hand, the maintenance of framework codes is easier compared to using own PHP code.

In this context, Symfony presents a powerful PHP framework that enables to create scalable and high-performance applications based essentially in PHP (Zaninotto François, 2007). One of the advantages of Symfony APIs is the easy integration with popular front-end frameworks such as AngularJS, used in our platform to visualize 3D medical images. Thanks to its secured methods of user registration and connection, Symfony offers better security at the level of the users' connection and the protection of their data uploaded on the platform (RĂDESCU, 2010). Symfony has been exploited rigorously in concrete projects, especially in e-activities projects such as in (PUPEZESCU, 2015) in which the authors developed a platform of easy learning that include databases, graphical design and high-tech devices. According to the authors, Symfony framework simplifies many of their repetitive tasks and provide to them the possibility of automatic entity generation, which reached the performance of their platform. Recently M. Benmoussa et al. (BENMOUSSA, 2017) developed a platform for the Governance of University Cooperation to ensure the continuation of constructive and productive collaborations between universities using Symfony platform, according to their experience.

In October 2017, the NASA and Arizona University teams (STONE, 2017) conducted an open data repository to assess the needs of diverse research groups and software developing, which allow them to create and publish databases and other related data.

In this project, Symfony was used for creating extensible object-oriented software in PHP. In more researches, Symfony has proven to be a powerful tool to develop some cloud platforms (Samuel, 2016) (YING, 2016) (TOMLINSON, 2015).

The main factor of our choice of Symfony was its flexibility and adaption with our project needs. Indeed, Symfony is compatible with most of the database models, such as MySQL, Oracle, and Microsoft SQL Server (MAASWINKEL, 2015). In addition to that, Symfony can be installed and learned in a short time thanks to its big community. All these useful properties can significantly increase the development speed and web platform maintenance and improvements. This allows programmers to focus more on logic of the work than implementing the code itself.

2.3 SECURITY WITHIN CLOUD PLATFORMS

In case of medical applications, we need to secure our platform within serval protocols. Authors in (Gupta, 2015) explain that the best way to communicate between containers is the use of the secured protocols such as SFTP (Xia, 2010) , SSH (Alshammari, 2007), FTPS (Bhimani, 1996), and SCP.

In (Callegati, 2009), authors demonstrated that the protocol HTTPS presents actually the best solution when we have a web server, since it combines between the HTTP and a connection encrypted by Transport Layer Security, or its predecessor, Secure Sockets Layer.

In our case, we have chosen the famous protocol HTTPS for our web server, and SFTP protocol to ensure secure transfer of data within our platform. Notice that SFTP is provided with SSH protocol that allows to execute commands and run the applications.

Our contribution consists mainly on providing a cloud-based platform, which offers an interactive real-time image and video processing taking advantage of a high-performance computing with GPUs. It also handles multi user connection based on containers orchestration architecture. Our platform is among the few platforms which propose to experiment specialized medical application based on proven computer research studies and tackled security in data transfer and storage.

3 THE CLOUD PLATFORM

Our cloud platform can be described within four parts:

- Platform architecture
- Webserver and clients management
- Data encryption
- Applications management

3.1 PLATFORM ARCHITECTURE

To ensure good performance of our cloud-based applications, we used three Virtual Machines (VMs). The first one is used as a web server that communicates with the other virtual machines using SFTP and SSH protocols. SFTP is used to send data between machines, and SHH is used to run the application or the encryption of data.

The second VM is used to encrypt the data related to medical applications. In this case, users prefer generally to secure their data, which is done with Hadoop architecture. The third VM is used to launch and execute the different applications. For this aim, we have used the Docker framework in order provide a multi-user exploitation of our applications in real time. Different users can run the same application simultaneously. The general architecture of our cloud-based platform is described in Figure 1.

Notice that Docker container is an open source software development platform. Its main benefit is to package applications in “containers”, allowing them to be portable among any system running the Linux operating system (OS). (Merkel, 2014)

Figure 1 shows that our platform is composed of three principal virtual machines. The heart of our platform is the virtual machine running the web server, because this machine interacts with the client and with the other virtual machines.

3.2 WEBSERVER AND CLIENTS MANAGEMENT

In our case, we have developed the website with the Bootstrap framework that allows to have a multi-platform website that can be run even on mobile devices (smartphone, tablet, etc.). On the other hand, the framework Symfony is used to offer more security to our website. The protocol HTTPS is based on certificate of Let’s encrypt [1], which represents a free, automated, and open certificate authority brought by the non-profit Internet Security Research Group (ISRG).

3.3 DATA ENCRYPTION

Since our cloud platform should integrate medical applications, the task of securing data is so important. Therefore, all the data are encrypted by using the protocol RSA (Kalpana, 2012), and to ensure this operation, we have used the framework Hadoop. The latter allows to have a replication of data with the HDFS system. RSA is a cryptosystem for public-key encryption, and is widely used for securing sensitive data, particularly when being sent

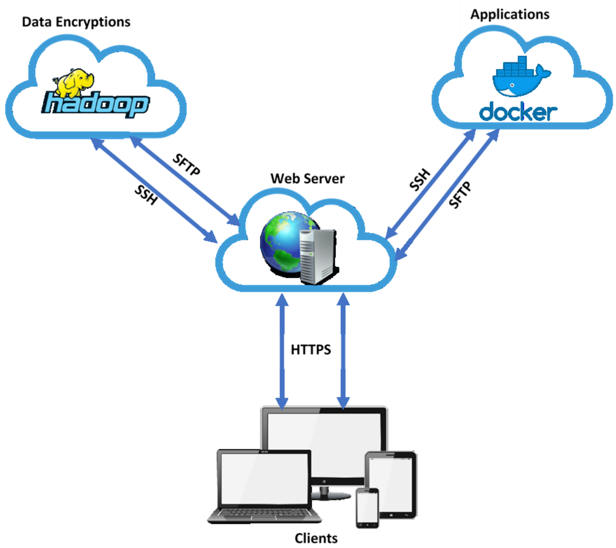


Figure 1 | The general architecture of our platform.

over an insecure network such as the Internet.

The general architecture of encryption protocol is shown in Figure 2. This figure illustrates three VMs, one as master and the two others as slave. The operation of encryption is executed in the master machine when the user accepts to store his data in the platform. In this case, we use Hadoop and HDFS systems to replicate the data in the two other machines (slave). Our use of Hadoop is due to its efficiency and availability since Hadoop is provided, by default, by Google Cloud machines, used for our cloud-based platform. Thanks to this implementation, the data can be recovered either if we have a crash in the slave machine. Notice that HDFS is a distributed file system that provides high-performance access to data across Hadoop clusters. Like other Hadoop-related technologies, HDFS has become a key tool for managing pools of big data and supporting big data analytics applications.

3.4 APPLICATIONS MANAGEMENT

In order to have a multi-user access and exploitation of our applications, we have used the Docker framework, which allows to generate and configure an image including all the libraries that are required from our algorithms and applications such as OpenCV [2], Java [3], ITK [4], etc. This image is called "basic-docker-image" in our case. Within Docker, we generated a second image "nvidia-docker" since we have also some applications that require the exploitation of GPU cards (Mahmoudi S. A., 2016).

Once the user selects an application, the web server receives the data and choose the appropriate VM. This machine will create a container based from the "basic-docker-image" generated previously. Thereafter, the container will launch the application and store the result in a folder shared between the VM and the container. Finally, the VM can delete the created container.

In the case where we have many users that want to execute the same application simultaneously, the related VM can create for each user a container, which ensures the simultaneous multi-user access in real time. This process is summarized in Figure 3.

4 EXPERIMENTAL RESULTS

In order to test and validate the above-mentioned cloud platform, we have integrated four kinds of applications and algorithms:

- Image processing toolbox.
- Video processing toolbox.
- Computer aided diagnosis for scoliosis
- Spine detection and segmentation in MRI

4.1 IMAGE PROCESSING TOOLBOX

First, we integrated an image processing toolbox in our platform. This toolbox allows users and developers to test, exploit and combine several image processing algorithms such as image denoising, features extraction, SIFT and SURF descriptors, etc. Within our cloud platform, these algorithms can be turned in real time, either if we change the input parameters. The integrated algorithms are compiled in a Docker image in order to offer to users the possibility to group the application parameters (input parameters, algorithm type and hardware type) in a container. With this, the container can execute the algorithm and send the result to the website.

Thanks to this configuration, the change of material does not affect our configuration since we have just

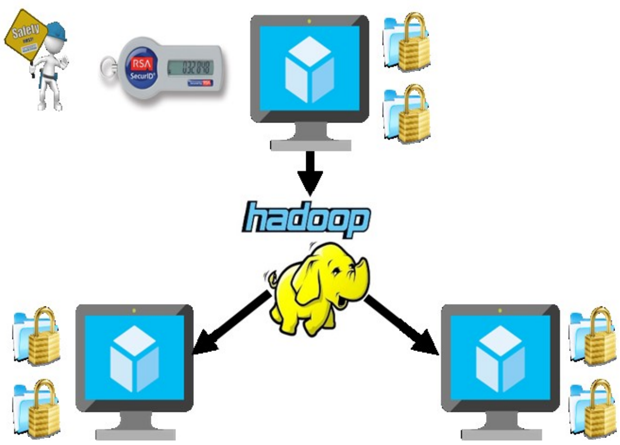


Figure 2 | Data encryption of our platform.

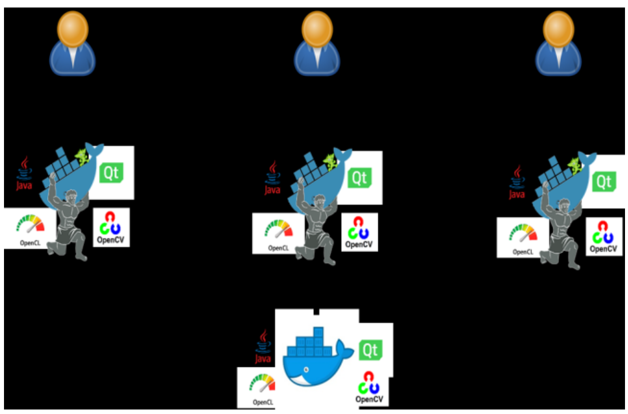


Figure 3 | Architecture of our cloud-based applications.

Figure 4 | Illustration of our cloud-based SURF descriptor on CPU and GPU.

to install Docker and download the appropriate images. Moreover, this configuration allows us to have a multiple user access simultaneously. Notice that the integrated algorithms can exploit CPU or GPU depending on the users preferences.

4.2 VIDEO PROCESSING TOOLBOX

The video processing toolbox provides algorithms related to video processing and motion tracking algorithms such as background subtraction, face, people and cars detection and tracking in real time, etc. The above-mentioned algorithms are also integrated and adapted for processing videos instead of images. In this case also, users can exploit CPU or GPU for treatment. Moreover, Docker containers are used in order to offer multi-user access and real-time processing.

Figure 4 illustrates an example of SURF descriptor extraction, launched within our platform. As shown in the figure, users can select the material: CPU or GPU. We note that the use of GPU offers betters performance as a result of the parallel exploitation of GPU's computing units. The address the toolbox platform is: https://toolbox.media-process.com/. Users are invited to register, connect and test our toolbox algorithms.

4.3. COMPUTER AIDED DIAGNOSIS FOR OSTEOPOROSIS

In order to offer a 3D reconstruction of the bone density and microarchitecture from 2D radiographs, we developed a multiplatform application, implemented within Java. The latter allows extracting information about bone density, trabecular microarchitecture, and cortical microarchitecture.

Bone density: represented by several metrics such as the BMD (Bone Mineral Density), the volume fraction (fraction of the bone in the total volume) and the T-score that allows to detect bone fragility and osteoporosis.

Trabecular microarchitecture: represented by serval metric related to the trabecular microarchitecture of the bone such as: the connectivity (between trabecules), the trabecular thickness and the thickness separation (between trabecules).

Cortical microarchitecture: represented by parameters related to the cortical microarchitecture.

In this section, we present three different versions of our web-based application of 3D bone analysis: basic, advanced and personalized. For each version, the user should provide three parameters:

The 3D volume: represented by the file containing the DICOM (Digital Imaging and Communications in Medicine) slices of the bone that we want to analyze.

Phantom values: represented by some specific points values (in the Phantom) that we know in advance their density values. These pairs of values (point value in HU and density value in mg/cm3) are used to apply a calibration for estimating the density of each point within the 3D volume.

BMD population: represented by a list of BMD values related to some healthy patients in order to compute the Mean BMD and standard deviation. The latter are used to estimate the T-score that allows to detect osteoporosis.

4.3.1. Basic version

In terms of computation, the basic version allows extracting the density and microarchitecture (trabecular) values. The density values are represented by several parameters such as:

Volume Fraction: represented by the fraction between the bone volume and the total volume.

BMC: the bone mineral content that represents the density of mineral within the whole bone.

BMD: the bone mineral density that measures the mean density of minerals in bones. In our case, we use the Phantom values in order to apply a calibration for getting the density of each bone.

Fractal dimension: provides a statistical index about the pattern complexity of the bone.

T-score: represents the bone density compared with normally expected bones from healthy young adult (BMD population). The T-score is ranged between -2,5 and 2,5. With this, we can have three main diagnosis of bone:

- Normal : if the T-score is above that -1.0
- Osteopenia: if the T-score ranged between -1 and -2.5, (osteopenia that may leads to osteoporosis).
- Osteoporosis : if the T-score less than -2,5

Otherwise, the microarchitecture (trabecular) parameters are represented by three values: the trabecular thickness, the trabecular separation and the connectivity.

Trabecular thickness: represents the mean, standard deviation and maximum values of thickness of the trabecular bone.

Trabecular separation: represents the mean, standard deviation and maximum values of separation between trabecular bones.

Connectivity: represents the trabecular number per mm3 of the bone.

Notice that the values of volume fraction, fractal dimension and trabecular microarchitecture are calculated within BoneJ library (Doube M, 2010).

4.3.2. Advanced version

This version provides the 3D cortical analysis in addition to the density and trabecular analysis, already provided within the basic version.

The cortical analysis consists of measuring parameters related to the cortical bone such as:

Inertia moments: by computing the moments of inertia of the structure, such as a whole bone.

Slice geometry: by calculating the cross-sectional parameters such as second moment of area and section modulus.

4.3.3. Personalized version

This version offers to users the possibility to select their preference for bone analysis. Indeed, they can select one or several options from:

- 3D bone trabecular analysis
- 3D bone cortical analysis
- 3D-based Anisotropy computation
- 3D visualization and reconstruction

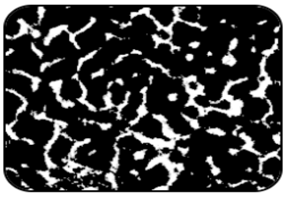
Notice that the anisotropy computation allows measuring, counting and classifying the skeleton branches and junctions. The 3D visualization allows to display the slices of the input and output volumes. The output volume illustrates with colors the trabecular thickness and separation as shown in Figure 5. Otherwise, we provide algorithms for DICOM visualization and reconstruction. These tools allow users to have a complete idea of the bone before its analysis.

4.4. SPINE SEGMENTATION IN MRI

In this part, we present three different scenarios of a medical image analysis application as a web service. This is an extended version of our cloud-based spine detection and segmentation application in T1 weighted MR images (Mahmoudi S. A., 2017).

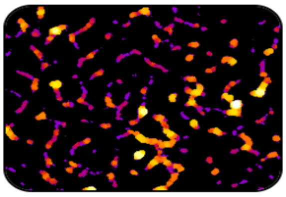
The application is composed of three web services, namely basic, advanced and personalized. In each service, the user uploads an input image, a central sagittal MRI of the spine converted from DICOM, and can interactively tune the input parameters of the algorithm. The basic service is a demonstration of the functionality of the application, with an early version of the algorithm behind. The advanced service offers a full version of the algorithm and enables to use more tuning parameters compared to

Input volume



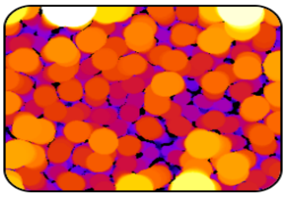
Download the volume

Thickness volume



Download the volume

Thickness separation volume



Download the volume

Density (BMD) Analysis

Volume Fraction			Fractal Dimension		Density values		
Bone Volume	Total Volume	Volume Fraction	Fractal Dim	R2	BMC	BMD	T-Score
1.216	14.825	0.082	1.582	0.991	0.12	8.37	-33.11

Microarchitecture (Trabecular) Analysis

Trabecular Thickness			Connectiviy		Trabecular Thickness Separation		
Mean	Std Dev	Max	Connectivity	Trabecular. N	Mean	Std Dev	Max
0.146	0.058	0.313	38.500	2.597	0.850	0.287	1.689

Microarchitecture (Cortical) Analysis

Inertia Moments										
Bone code	CSA (mm^2)	Xcent (mm)	Ycent	Density	Theta	Feret Min	Feret Max	Feret Angle	Peremiter	Mean Thick
0	6.346	4.921	4.189	1.800	0.219	0.166	0.636	0.742	1.441	0.151
Slice Geometry										
Xc (mm)	Yc (mm)	Zc (mm)	Mass (g)	Vol (mm^3)	Icxx ($Kg.m^2$)	Icyy	Iczz	I1 ($Kg.m^2$)	I2	I3
4.916	4.163	0.116	1.216	0.002	0.010	0.015	0.025	0.025	0.015	0.009

Anisotropy computation

DA	tDA
0.693	3.254

Download the report

Save results on your MBP server

Figure 5 | Personalized version of our Cloud-based Computer Aided Diagnosis for Osteoporosis.

the basic one. The advanced service (cf. Figure 6) offers a full tuning of the algorithm and a GPU implementation, which takes advantage from the OpenCV GPU ready functions. The latter, offer an in-depth analysis and experimentation of the different steps of the algorithm, it can also be generalized to different use cases related to clustering and blob detection in images. The application performs vertebra detection and segmentation as well as spine curve extraction. The personalized version allows to analyze gray profile of the spine curve, a useful tool for preliminary anomaly detection.

The main steps of the algorithm behind are as follow:

- Preprocessing: to reduce noise and enhance contrast in the input image
- Clustering: to find homogenous regions in the preprocessed image
- Shape recognition: based on an iterative polynomial regression to find vertebra centers and spine curve.

For more details on the underlying steps of this method refer to (Larhmam, 2014). The three web services communicate with a remote virtual machine, which hosts the container of the core application, as explained in Section 3.1. The adopted architecture seamlessly automates the management of multi-user connections to the different services, thanks to containers orchestration. The above-mentioned applications and experimentations demonstrated the interest of our platform for sharing scientific contributions related to computer vision and medical imaging domains. The scientific researchers could be able to develop and share easily their applications fast and in a safe way. The address of our platform is: https://bone.media-process.com/. Users are invited to register, connect and test our applications.

CONCLUSION

In this paper, we have presented a cloud-based toolbox able to run in real time computer vision algorithms. To that end, three VMs were used: one for the web server, one for the encryption of data and one executing the applications. In order to ensure a multi-user access to the applications in the toolbox, the Docker framework were used, allowing to generate and configure all the required libraries, and to create separate containers executing the applications. The toolbox was evaluated through the execution of four kinds of applications and algorithms: image processing, video processing,

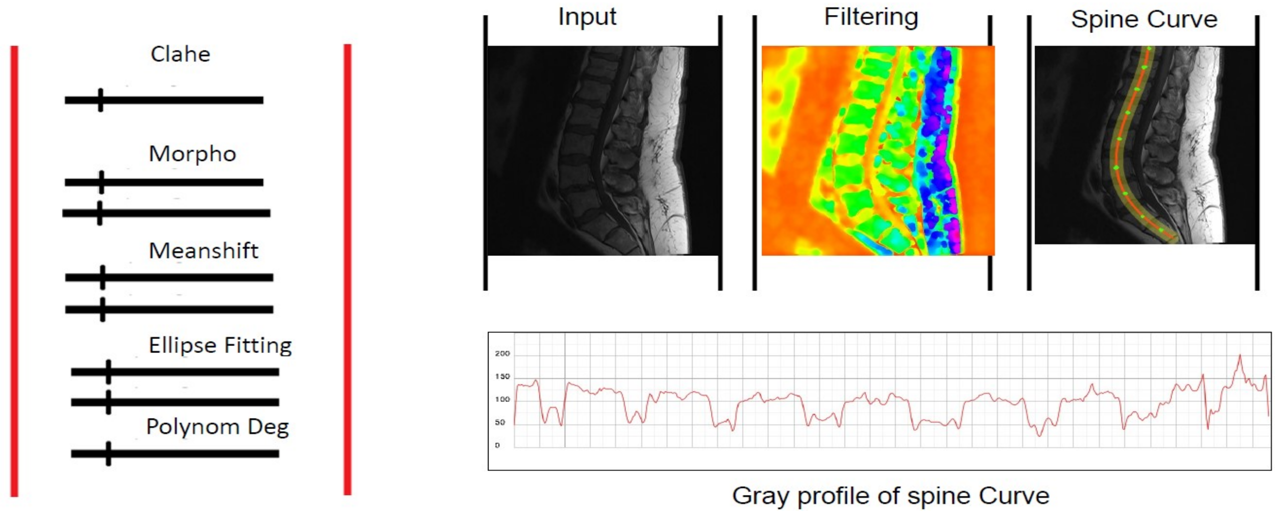


Figure 6 | Illustration of the functionalities of the advanced services with the expected output.

computer-aided diagnosis, and spine detection and segmentation in MRI. This evaluation showed that our platform can deal with heterogeneous data (image, video, etc.) and is suitable for various applications. The security was also a concern of the platform. In the future, we hope that other scientific researchers will develop and share their applications in a fast way and easily thanks to our platform.

ACKOWLEDGEMENTS

Authors would like to thank the support of eNTERFACE workshops organizers. The last workshop allowed us to finalize our cloud-platform.

ENDNOTES

[1] Let’s encrypt: https://certbot.eff.org/.

[2] https://opencv.org/

[3] https://www.java.com

[4] https://itk.org/

REFERENCES

Agrawal, H. M. (2015). Cloudcv: Large-scale distributed computer vision as a cloud service. In Mobile cloud visual media computing, pp. 265-290.

Alshammari, R. &.-H. (2007). A flow based approach for SSH traffic detection. Systems, Man and Cybernetics. ISIC. IEEE International Conference, 296-301.

BENMOUSSA, K. L. (2017). SIMACoop: a Framework Application for the Governance of University Cooperation. Transactions on Machine Learning and Artificial Intelligence, vol. 5, no 4.

Bhimani, A. (1996). Securing the commercial Internet. Communications of the ACM, 29-35.

Calasanz, R. B.-I. (2017). Towards the Scientific Cloud Workflow Architecture. 5th International Workshop on ADVANCEs in ICT Infraestructures and Services (ADVANCE'2017.

Callegati, F. C. (2009). Man-in-the-Middle Attack to the HTTPS Protocol. IEEE Security & Privacy, 78-81.

Doube M, K. M.-C. (2010). BoneJ: free and extensible bone image analysis in ImageJ. Bone 47:1076-9.

Gupta, U. (2015). Comparison between security majors in virtual machine and linux containers. Comput. Res. Repos., 4.

Kalpana, P. &. (2012). Data security in cloud computing using RSA algorithm. IJRCCT, 143-146.

Larhmam, M. A. (2014). A portable multi-cpu/multi-gpu based vertebra localization in sagittal MR images. In International Conference Image Analysis and Recognition, Springer, pp. pp. 209-218.

Limare, N. a.-M. (2011). The IPOL initiative: Publishing and testing algorithms on line for reproducible research in image processing. Procedia Computer Science , pp. 4:716-725.

MAASWINKEL, T. (2015). Practical Symfony 2. Apress.

Mahmoudi, S. A. (2014). Taking Advantage of Heterogeneous Platforms in Image and Video Processing. High-Performance Computing on Complex Environments. Wiley.

Mahmoudi, S. A. (2016). Real time GPU-based segmentation and tracking of the left ventricle on 2D echocardiography. 4th International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO 2016, pp. 602-614.

Mahmoudi, S. A. (2017). Cloud-based platform for computer vision applications. In Proceedings of the 2017 International Conference on Smart Digital Environment, ACM, pp. pp. 195-200.

Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. Linux Journal, 239.

Netto, M. C. (2017, Oct 24). HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges. arXiv preprint arXiv:1710.08731.

PUPEZESCU, V. e. (2015). Enhanced protection level by database replication in the easy-learning online platform. Advanced Topics in Electrical Engineering (ATEE), 2015 9th In-ternational Symposium on. IEEE, pp. 929-932.

RĂDESCU, R. D. (2010). Security And Confidentiality In The Easy Learning On-line Plat-form. Proceedings of the 5th International Conference on Virtual Learning (ICVL-2010), pp. 449-452.

Samuel, D. (2016). Monitor and Control Manager update. Haute Ecole de Gestion & Tourisme.: Thesis.

Shvachko, K. e. (2010). The hadoop distributed file system. 26th symposium on Mass storage systems and technologies (MSST). IEEE.

STONE, N. L. (2017). Collaborative Data Publication Utilizing the Open Data Reposi-tory's (ODR) Data Publisher. NASA: American Geophysical Union, Fall General Assembly 2016.

TOMLINSON, T. (2015). Anatomy of a Module . Beginning Drupal 8, 141-145.

Xia, L. C.-s. (2010). Design of secure FTP system. International Conference on Communications, Circuits and Systems , 270-273.

Yan, Y. a. (2014). Large-scale image processing research cloud. Cloud Computing, pp. 88-93.

YING, H. C. (2016). EARec: leveraging expertise and authority for pull-request re-viewer recommendation in GitHub. Proceedings of the 3rd International Workshop on CrowdSourcing in Software Engineering. ACM, pp. 29-35.

Zaninotto François, P. F. (2007). The definitive guide to Symfony. Apress.

BIOGRAPHICAL INFORMATION

Sidi Ahmed Mahmoudi received the graduate engineering degree in computer science from the University of Tlemcen, Algeria, the master degree in multimedia processing from the Faculty of Engineering in Tours, France, and the PhD degree from the University of Mons, Belgium, in 2006, 2008, and 2013, respectively. Currently, He is working as PhD research associate at the University of Mons, Belgium. His research interests are focused on the efficient exploitation of local and cloud computing resources (multi-CPU/ multi-GPU) architectures for multimedia processing and machine learning algorithms within Big Data volumes. Sidi Ahmed has participated in several national projects and European actions. He is author or co-author in more than 60 international publications.

Mohammed Amin BELARBI is a PhD Student from the Mathematics department, Faculty of Exact Sciences and Computer Science, University of Mostaganem, Algeria and from computer science department, Faculty of Engineering, University of Mons, He started his PhD the in University of Mostaganem, Algeria and University of Mons in 2014 (Co-Tutelle). He holds a master's degree in engineering information system in 2013 and Computer Science degree in 2011 from University of Mostaganem. His research interests are Multimedia Retrieval, Indexing videos and images.

Mohammed EL ADOUI received the Master degree in computer science (computer graphics and Image processing) from the University of Fez, Morocco, in 2015, and the basic license degree in Mathematics and Computer science from the Faculty of Science in Oujda, Morocco in 2013. Currently, he is a PhD student at the University of Mons, Belgium. In 2015, he started a PhD, is focused on breast cancer prediction and segmentation.

Mohammed Amine Larhmam received the graduate engineering degree in Modeling and Scientific Computing from Mohammadia School of Engineering in Rabat, Morocco in 2010. He worked as a project engineer for 9 months at MCINET, Rabat. In 2017, he obtained a PhD in engineering science in the same department. Currently, he works as teaching assistant in the computer science department at the faculty of Engineering, University of Mons. His researches focus on computer vision and machine learning with applications to Medical Image Analysis and Computer Aided Diagnosis.

Fabian Lecron received the computer science engineering degree from the Faculté Polytechnique de Mons (FPMs), Belgium, and the management sciences degree from the Facultés Universitaires Catholiques de Mons (FUCaM), Belgium, respectively in 2008 and 2011. He obtained a Ph.D. degree in applied sciences at the University of Mons (formerly Faculté Polytechnique de Mons) in 2013. He is now postdoctoral researcher at the University of Mons (UMONS), Belgium. His main research areas are computer vision, image processing, collaborative recommendation, and data mining.