

Software for Automatic Treatment of Large Biomedical Images Databases

Mathieu Monziols¹, Julien Faixo², Elias Zahlan² and Gerard Daumas¹

¹ IFIP Institut du porc, Antenne Rennes - Le Rheu, La Motte au Vicomte, B.P. 35104, 35651 Le Rheu Cedex, FRANCE

² IFIP Institut du porc, Antenne Toulouse, 34 boulevard de la gare, 31500 Toulouse, FRANCE

`Mathieu.monziols@ifip.asso.fr`

Abstract. Measuring body composition of live animals or carcasses by CT involves the acquisition of large number of images. The software presented here is a simple and user friendly analysis tool for large images datasets. The software was developed in C# for windows. It permits to define instinctively by a graphical interface the different operations of an analysis scheme, and to apply the resulting treatment automatically to a large selection of images. Furthermore the software also easily allows the adding of new analysis operations. In the example presented here the software was able to apply a rather simple image analysis treatment for body composition measurement (threshold followed by mathematical morphology) on a dataset of more than 200 000 images in 838 minutes.

Keywords: Image Database, Software, Automatic, Dicom, Pig, CT.

1 Introduction

Imaging technologies are more and more used on farm animals and food products. High density phenotyping should increase in the future the number of studied animals and images. Automatic computation is thus a crucial issue. Furthermore, the huge number of images makes more necessary the processing by non-specialists in image analysis, implying simpler tools.

Our institute, which is mainly dedicated to pig and pork products, bought a CT in 2007. More than 1000 pig carcasses are yearly scanned. For an average of 1.5 meter length and a 3 mm slice thickness, which is our standard setting, each pig generates 450 images. This means about a half-million of images every year.

adfa, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011

As such a software was not available, we decided to develop a specific image analysis software.

After the presentation of the requirements our choices are explained. Then, the structure and the functionalities of the software are presented. Finally, an example of speed performances is given for the determination of the muscle volume in pig cuts.

2 Requirements for the software

Four main requirements were established before the development of this image analysis software. They deal with: type of images, kind of user, capability of evolution, computing time.

CT scanners produce Dicom images. Directly manipulating Dicom images was therefore an obvious requirement.

These images have to be managed either by advanced users or by non-specialists.

Another requirement was the possibility to add new operations in the software without touching its core design, like with some plug-ins. This would allow a continuing evolution.

We also wanted the software to be quite optimized in order to reduce the calculations time, even if the aim was to launch a large number of analysis without human intervention.

3 Development and structure choices

3.1 Development choices

Two choices had to be done: one for the development environment and the other one for the language.

The choice of the development environment was guided by the ability to directly manipulate DICOM images. A lot of frameworks had this ability. Clearcanvas (<http://www.clearcanvas.ca/dnn/>) was chosen because it was a maintained, open source framework that fulfilled all our requirements. As Clearcanvas was developed in C#, it seemed consistent to make the developments in C# too; Visualstudio 2008 with Microsoft.net 2.0 was chosen.

Nevertheless, some operations could need more advanced mathematical calculation possibilities than the ones offered by the C# language. In order to have access to a complete mathematical library, the possibility to use the Python language in the operation coding was added via the software pythonxy (<http://code.google.com/p/pythonxy/>), which permits to interface both languages and exchange data in both directions (C# to Python, then Python to C#).

3.2 Structure choices

The software is built in two parts: one for the selection of the successive steps of the image analysis algorithms, called workflows, and one for the execution of these workflows, applied on image sets.

The development of a workflow is done by a graphical interface, using some dataflow programming language (Nikhil, 1991). It consists in a succession of operations with inputs and outputs; outputs are generated from inputs transformations. The workflow is materialized on the screen by some “bricks” and some “threads”. The bricks represent the operations and the threads are linking them. A thread links the output of an operation to the input of another operation.

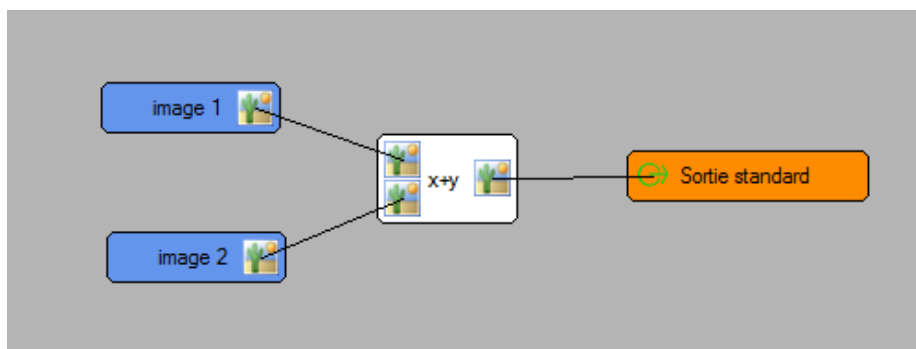


Fig. 1. Image of bricks and threads making an addition of two images

Operations are all stored in dynamic libraries (DLL) loaded by the software. New operations can be developed and then integrated in the software.

Operations can be iterated, if it makes sense. This possibility allows transforming each input and each output of an operation into input and output of the same type but of a superior set type.

For example, let's consider the operation calculating the mean of an image signal. Input is an image and output is a mean value. A list of images (like a complete patient) as input will give a list of data (the mean signal of all images) as output by using the iteration of this operation.

Two types of workflow execution were developed.

The first one is only dedicated to the test of the designed workflow. It is a step by step execution, each operation is executed one after the other and the execution can be paused in order to visualize and check the operation output.

The other execution type is a multi-agents execution inspired from the multi-agents system (Ferber, 1995). This execution type allows a highly parallelized execution and a better memory management. This is the execution type generally used in the software.

4 Simple presentation of the software

The software is mainly composed of two tabs: the treatment tab and the workflow tab. The treatment tab allows for launching analysis from already designed workflows. In this tab the user chooses the name of the analysis, the workflow and the different inputs. When the inputs are images or group of images (“patients”), a navigator can be opened to select them via drag and drop. The inputs list is iterative: if a workflow is designed for one image and if the user drags and drops N images, then the workflow will be applied to each image of the input list. Furthermore, it is possible to prepare in this tab the whole analysis the user wants. The analysis is defined as successive workflows; the software will automatically launch each workflow one after the other.

The second tab, the workflow tab, allows to more advanced users to graphically develop their workflows and manage them.

On the left of the window there is the list of operations which have already been developed. The more classic operations of image analysis can be used by selecting: ROI selection, statistical measurement, fixed or automatic threshold, semi automatic segmentation by grow cut, mathematical morphology with structural element, histogram, etc.

Each operation needs specific inputs (image, numerical value, list of images (patients)). The user designs his workflow by dragging and dropping the operations, which are needed in the workflow windows, and by linking them with threads.

Then there is the possibility to verify the workflow integrity (if inputs and outputs are correct for the different operations) and to launch the workflow step by step or by classic multi agent execution to test the workflow.

This tab allows also saving the workflow in order to use it in the treatment tab.

5 Example of application

Recently, we had to measure the muscle volume of the four main joints (shoulder, loin, ham and belly) of 300 pig carcasses by CT scanning. 1200 “patients” were therefore created. The 3 mm slice thickness, which we consider as a good compromise between cost and accuracy, has produced between 150 and 200 images per “patient” (joint). There was at the end more than 200 000 images to analyze.

The workflow was quite simple, consisting in 10 “bricks” and 9 “threads” (Fig. 2). Five operations, four inputs and one output formed the 10 “bricks”. Firstly, a ROI was made from the patient to check that the mean of the muscle is at about 60 Hounsfield Units (HU). Secondly, a simple threshold (“Seuillage standard” in French) was made by inputting the inferior limit (“seuil inf” = 0) and the superior limit (“seuil sup” = 120). Dumas and Monziols (2011) have shown the range 0-120 HU was efficient for

muscle segmentation. Nevertheless, the skin has a signal very close to the muscle one and cannot be thresholded. In order to remove it, a mathematical morphology operation, was done; this operation, an opening (“Ouverture”), needs a structural element (“struct”) as input to filter the thresholded images, outputting the filtered images. Then, the number of muscle pixels in each slice was calculated by summing up (“Somme”) the number of muscle pixels in the filtered images. Finally, the number of muscle pixels in each joint was calculated by summing up (“Somme”) the number of muscle pixels in each slice.

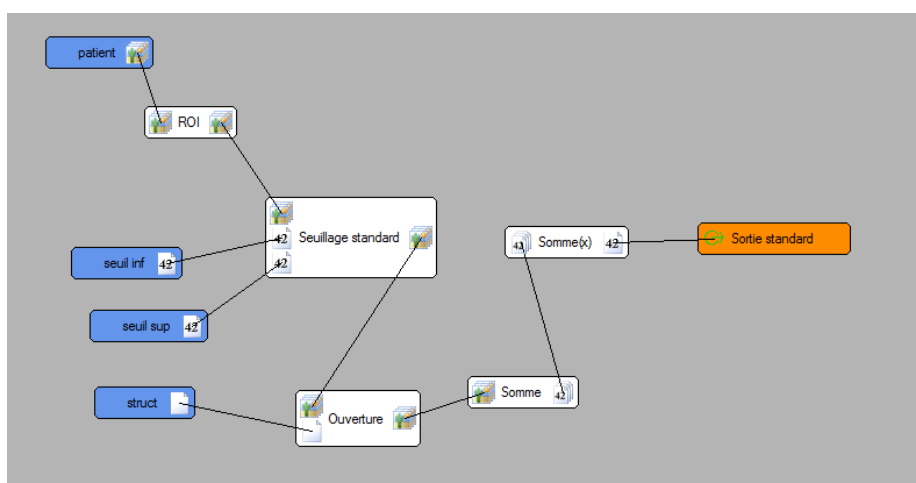


Fig. 2. Workflow used for measuring the pixels number of muscle on pig joints images

The result of the workflow is a number of muscle pixels for each joint which can easily be converted into a volume, by multiplying the number by the pixel size. An iteration of this workflow allows to easily calculate this volume for all the joints.

This analysis was launched on a computer equipped with an Intel-core I7-3610QM, 12 GB RAM, and with an SSD hard drive. Indeed, a lot of cache writing was needed, because of the limitation by Microsoft.net 2.0 to 2 GB Ram per application, which was quickly attained by the software. The cache was put on the SSD hard drive in order to gain calculation time.

With such a configuration the analysis was done in 838 min (about 14h), so approximately 4 images were analyzed per second. We consider that it is an acceptable performance result.

6 Conclusion

A software was developed to automatically deal with a large amount of Dicom images. Written in C# and authorizing Python language, this software allows simple workflow programming of image analysis by using a succession of operations already integrated in the software. Furthermore the software can evolve with development of new operations in separate DLLs. Automatic analysis can be done simply by multiplying the inputs for a same workflow.

This software is used in our institute since 2010 to process most of CT images. We plan to use it for MRI images too. This kind of software is really interesting for people needing to analyze a lot of images without user intervention.

Nevertheless, the software is still in beta state, some operations such as registration are not working yet, and other operations would be interesting to develop. A possible evolution towards Microsoft.Net 4.5 and a 64 bits version would remove the memory limitation.

References

Daumas, G., Monziols, M.: An Accurate and Simple Computed Tomography Approach for Measuring the Lean Meat Percentage of Pig Cuts. In: 57th ICoMST, paper 061. Gant (2011)

Ferber, J.: Multi-agents Systems: towards a Collective Intelligence (in French). InterEditions, Paris (1995)

Nikhil, R. S.: Id (Version 90.1) Reference Manual. Technical Report CSG Memo 284-2. MIT Laboratory for Computer Science, Cambridge (1991)