Lung nodules segmentation in CT scans by DeepHealth toolkit

Marco Aldinucci, Barbara Cantalupo, Iacopo Colonnelli, Marco Grangetto, Riccardo Renzulli, Enzo Tartaglione
Computer Science Department, University of Turin, Torino, Italy
Marco Grosso, Giorgio Limerutti
Azienda Ospedaliera Città della Salute e della Scienza Presidio Molinette, Torino, Italy

***Abstract*—In this demo, we will show how a Deep Learning pipeline, whose goal is to train a model to recognize lung nodules from chest CT scans, can be executed on a hybrid HPC-Cloud infrastructure. Specifically, the OpenDeepHealth infrastructure available at the University of Torino will be presented, along with the usage of the Deep Learning libraries developed in the DeepHealth EU project. The AI pipeline is composed of both training and inference steps that are transparently executed on different target architectures, in order to better fit the specific characteristics of each computation without steepening the learning curve for the AI experts.**

I. Introduction

There are great expectations to unleash disruptive innovation across the health sector thanks to better exploitation of health data and Artificial Intelligence (AI)/Machine Learning (ML). The EU DeepHealth project [1] aims at contributing to such an epochal change exploiting in particular biomedical multidimensional images and state of the art AI based on Deep Learning (DL) methods. The DeepHealth project targets the development of a European Computer Vision Library[1] (ECVL) and a European Distributed Deep Learning Library[2] (EDDLL), coupled with advanced programming models optimized for parallel execution [2], [3]. On the one hand, the internal algorithms of EDDLL will be adapted to exploit the performance of advanced hardware accelerators (i.e. GPUs and FPGAs) and, on the other hand, the procedure for training predictive models will be efficiently distributed on hybrid and heterogeneous High Performance Computing (HPC) architectures. This demo is devoted to one of the 14 use cases defined in the project, namely automatic lung nodules segmentation in chest Computer Tomography (CT) scans. Lung nodules are small focal lesions in the lung parenchyma, can be solitary or multiple and in many cases are accidentally found in CT scans. Their identification is time consuming in the current clinical activity for the radiologist and, since these small lesions are difficult to spot, patients often need to perform follow-up CT scans in order to assess their benignity/malignancy, resulting in increased radiation exposure and anxiety for the patient and increased work amount for doctors. Lung nodules are quite common incidental findings in CT scans and can be defined as small focal lesions (ranging from 5 to 30 mm) that can be solitary or multiple. Some recently published studies show promising results about AI (artificial intelligence) applications for detection and characterization of lung nodular lesions [4], [5].

[1]https://github.com/deephealthproject/ecvl
[2]https://github.com/deephealthproject/eddl

II. DeepHealth toolkit for Lung segmentation

A. Neural model for lung segmentation

Deep learning models outperformed traditional computer vision techniques in various tasks. Typically, features of the input data are hand-crafted, but deep-learning features are learned in an end-to-end fashion. Convolutional Neural Networks (CNNs) are one of the most popular models that are also employed in medical imaging. More specifically, the U-Net [6] model is used to identify lung nodules in CT scans. This architecture consists of an encoder which takes as input an image (and encodes it in a low-dimensional tensor), and a decoder which reconstructs the image and outputs the segmentation map (where each pixel is classified as *background* or *lung nodule*).

The dataset used to train and evaluate the network (Table I) is provided by *Città della Salute e della Scienza di Torino*. In order to get preliminary results, splits have been created with only images with a ground truth mask, such that the training set contains 80% of images, val 10% and test 10%.

	Patients	Images
Train	247	13589
Validation	61	1699
Test	109	1708

TABLE I: Dataset used to train and test the U-Net model.

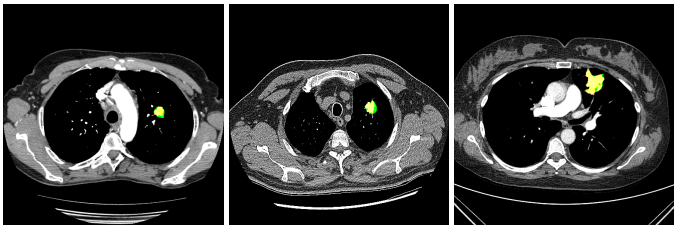

Fig. 1: Three U-Net segmentation maps. Ground truth: green, prediction: red, intersection: yellow.

CT scans are in DICOM format: before feeding the 2D slices to the model, they have been pre-processed according

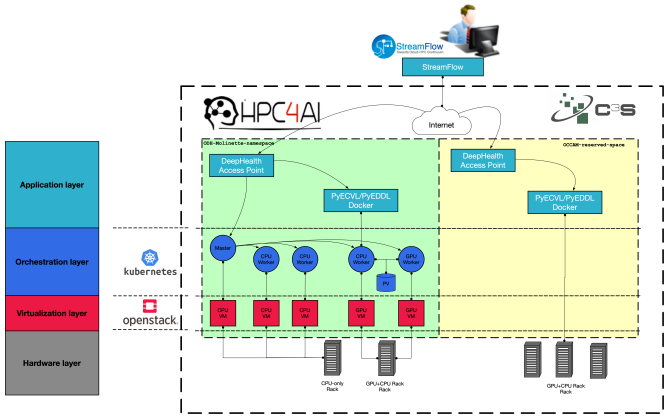

Fig. 2: OpenDeepHealth architecture

to a standard pre-processing pipeline[3]. As evaluation metrics, the Intersection Over Union (IoU) of the lung nodule class is computed. The model currently reaches an IoU of 0.62 with only rotations as data augmentation. Figure 1 shows three examples of the U-Net segmentation maps.

B. The OpenDeepHealth platform

The target infrastructure for this demo is OpenDeepHealth (ODH), a hybrid HPC-Cloud platform provided by the University of Torino (Fig. 2), integrating the DeepHealth libraries. The HPC part of the platform is the C3S OCCAM[4] supercomputer [7], which provides an elastic virtual farm of Docker containers running directly on top of the bare metal layer. The cloud part is instead hosted on the HPC4AI[5] [8] infrastructure, which is a federated OpenStack cloud serving multi-tenant private Kubernetes instances.

The OpenDeepHealth cluster is defined to provide Docker containers integrating both ECVL and EDDLL libraries. Moreover, the integrated StreamFlow[6] workflow manager [9], an orchestration layer on top of the Kubernetes cluster, allows users to easily model and seamlessly execute workflows in different, heterogeneous environments. Indeed, StreamFlow enhances Kubernetes' ability to deal with different computational steps (and their data dependencies) on potentially complex, multi-agent runtime environments and automatically handles data-transfers among different sites.

C. Putting it all together

The goal of the demo is to demonstrate not only how OpenDeepHealth and the DeepHealth libraries can be used to efficiently execute a real AI pipeline, but also the advantages of transparently performing the computation in a hybrid, potentially federated architecture. In this setting, an orchestration tool that masks the complexity of the underlying infrastructure to the AI expert becomes crucial.

[3]https://github.com/EIDOSlab/medical-image-preprocessing/blob/master/preprocessing_example.ipynb

[4]https://c3s.unito.it/

[5]https://hpc4ai.unito.it/

[6]https://streamflow.di.unito.it/

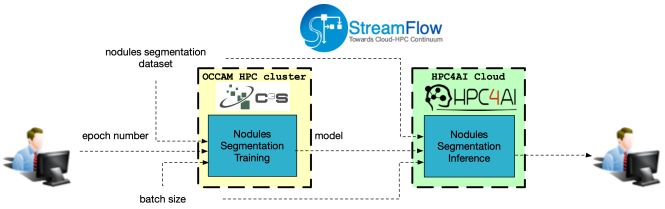

Fig. 3: Demo flow overview

The demo will showcase how the AI pipeline described in Sec. II-A can be executed on top of the OpenDeepHealth infrastructure introduced in II-B. As shown in Fig. 3, such pipeline can be modelled as a workflow with two tasks:

- A computational-heavy training step, which is executed on a GPU node on the OCCAM infrastructure;
- A subsequent, much more lightweight inference step, that can be offloaded to a CPU-equipped Kubernetes worker node in the HPC4AI cluster.

An AI expert can simply launch the pipeline directly from his computer using StreamFlow, which orchestrates the execution of the first step on OCCAM and the second one on HPC4AI and manages all the required data transfers in a fully transparent way.

From a practical point of view, the demo starts with the terminal of the AI expert, that launches StreamFlow. The execution of the first step of the pipeline starts on the GPU node and can be monitored in the GPU shell. When this step finishes, the validation starts on the HPC4AI cloud using data transparently transferred by StreamFlow. Eventually, the inference, step completes and the overall workflow execution terminates. StreamFlow handles the transfer of the output data into the AI expert's machine and the output can be visualized and analyzed on premise.

ACKNOWLEDGMENTS

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111, DeepHealth Project.

REFERENCES

[1] M. Caballero, J. Gomez, and A. Bantouna, "Deep-learning and HPC to boost biomedical applications for health (DeepHealth)," in *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2019, pp. 150–155.

[2] R. M. Badia, J. Conejero, C. Diaz, J. Ejarque, D. Lezzi, F. Lordan, C. Ramon-Cortes, and R. Sirvent, "COMP Superscalar, an interoperable programming framework," *SoftwareX*, vol. 3-4, pp. 32 – 36, 2015.

[3] E. Tejedor, Y. Becerra, G. Alomar, A. Queralt, R. M. Badia, J. Torres, T. Cortes, and J. Labarta, "PyCOMPSs: Parallel computational workflows in Python," *Int. J. High Perform. Comput. Appl.*, vol. 31, no. 1, pp. 66–82, 2017.

[4] J. G. Nam, S. Park, E. J. Hwang, J. H. Lee, K.-N. Jin, K. Y. Lim, T. H. Vu, J. H. Sohn, S. Hwang, J. M. Goo *et al.*, "Development and validation of deep learning–based automatic detection algorithm for malignant pulmonary nodules on chest radiographs," *Radiology*, vol. 290, no. 1, pp. 218–228, 2019.

[5] V. K. Raghu, W. Zhao, J. Pu, J. K. Leader, R. Wang, J. Herman, J.-M. Yuan, P. V. Benos, and D. O. Wilson, "Feasibility of lung cancer prediction from low-dose ct scan and smoking factors using causal models," *Thorax*, vol. 74, no. 7, pp. 643–649, 2019.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015, pp. 234–241.

[7] M. Aldinucci, S. Bagnasco, S. Lusso, P. Pasteris, and S. Rabellino, "OCCAM: a flexible, multi-purpose and extendable HPC cluster," in *Journal of Physics: Conf. Series (CHEP 2016)*, vol. 898, no. 8, San Francisco, USA, 2017, p. 082039.

[8] M. Aldinucci, S. Rabellino, M. Pironti, F. Spiga, P. Viviani, M. Drocco, M. Guerzoni, G. Boella, M. Mellia, P. Margara, I. Drago, R. Marturano, G. Marchetto, E. Piccolo, S. Bagnasco, S. Lusso, S. Vallero, G. Attardi, A. Barchiesi, A. Colla, and F. Galeazzi, "HPC4AI, an AI-on-demand federated platform endeavour," in *ACM Computing Frontiers*, Ischia, Italy, May 2018.

[9] I. Colonnelli, B. Cantalupo, I. Merelli, and M. Aldinucci, "StreamFlow: cross-breeding cloud with HPC," *IEEE Transactions on Emerging Topics in Computing*, August 2020.