Towards Integrating ImageJ with deep biomedical models*

Adrián Inés, César Domínguez, Jónathan Heras, Eloy Mata and Vico Pascual

Dpto. Matemáticas y Computación, Universidad de La Rioja,
Edificio CCT, Madre de Dios, 53, E-26006 Logroño (La Rioja, Spain)
{adrian.ines, cesar.dominguez, jonathan.heras, eloy.mata,
vico.pascual}@unirioja.es

Abstract. Nowadays, deep learning techniques are playing an important role in different areas due to the fast increase in both computer processing capacity and availability of large amount of data. Their applications are diverse in the field of bioimage analysis, e.g. for classifying and segmenting microscopy images, for automating the localization of proteins or for automating brain MRI segmentation. Our goal in this project consists in including these deep learning techniques in ImageJ – one of the most used image processing programs in this research community. To do this, we want to develop an ImageJ plugin from which to use the models and functionalities of the main deep learning frameworks (such as Caffe, Keras or Tensorflow). It would be feasible to test the suitability of different models to the problem that is being studied at each moment, avoiding the problems of interoperability among different frameworks. As a first step, we will define an API that allows the invocation of deep models for object classification from several frameworks; and, subsequently, we will develop an ImageJ plugin to make the use of such an API easier.

Keywords: Bioimage, Deep Learning, Image Processing, ImageJ, Interoperability, Object Classification.

1 Problem statement

Currently, computer vision and artificial intelligence have many different applications in diverse fields: security, biology or medicine for example. Also, deep learning is playing an important role due to both the fast increase in computer processing capacity and the availability of large amount of data. Specifically, in the field of bioimage analysis, several applications of these deep learning techniques are being applied for classifying and segmenting microscopy images [1], for automating the localization of proteins [2] or for automating the segmentation of brain MRI [3] for example.

* This work was partially supported by Ministerio de Economía, Industria y Competitividad [MTM2014-54151-P, MTM2017-88804-P], and Agencia de Desarrollo Económico de La Rioja [ADER-2017-I-IDD-00018].

Applying these deep learning techniques is not easy for a non-expert user. It being required to have basic knowledge of programming and to know different concepts of deep learning. Because of this, our idea is to make the use of these deep learning techniques easier integrating them in a processing program, in this case ImageJ [4] - one of the most used image processing programs in the biological research community.

In addition, there are a lot of different deep learning frameworks (such as Caffe [5], Keras [6], TensorFlow [7] or DeepLearning4J [8]) with different aims and functionalities. So, we consider to improve the interoperability among these frameworks by means of integrating all of them in ImageJ.

2 Related work

As we have indicated, the goal of this project is to improve the interoperability between different deep learning frameworks and image processing programs - like ImageJ for example. There are some other projects that try to connect ImageJ with deep learning or machine learning frameworks. One of these projects, *ImageJ - TensorFlow* [9], connects ImageJ with TensorFlow. Specifically, this project translates between ImageJ images and TensorFlow tensors. Our approach to integrate deep learning techniques with ImageJ is different. We want to include an intermediate layer between the ImageJ program and the deep learning frameworks. In this layer we want to abstract the characteristics of the different frameworks to classify an image. In addition, we do not focus only on one deep learning framework, our aim is to integrate the main deep learning frameworks and to define a way to include any other framework in an easy way.

Due to the fact that there are a lot of deep learning frameworks with different characteristics, there exist several active projects with the aim to transform a model from one deep learning framework into other. The project *Deep learning model converter* [10] deals with this problem, allowing to use models in different frameworks easily. Other project in the same line is the *Open Neural Network Exchange (ONNX)* [11]. There exists a lot of different frameworks with their own specification and proprietary representation of the graph of the models. In this project, a common representation of the computation graph is provided and different converters have been created. Our approach tries to be different. We want to provide the users several models and frameworks. In this way, the users can select the most suitable option for their problems.

3 Proposal

Our goal in this project consists in providing ImageJ with different deep learning techniques. To do this, we want to develop an ImageJ plugin that allows the users to have the models and functionalities of the main deep learning frameworks (such as Caffe, Keras or Tensorflow) available. In this way, it will be possible to test the suitability of different models to the problem that is studied at each moment, avoiding the problems of interoperability between different frameworks.

As a first step, we will define an API that allows the invocation of deep models for object classification from several frameworks, as we can see in Figure 1. This API, will gather the main frameworks of deep learning and will offer an abstraction layer that allows us to use these frameworks through a set of methods. Our idea is to integrate in this API the main deep learning frameworks for object classification as well as including other functionalities such as object detection, object localization or image segmentation. Also, we want to allow the users to integrate other frameworks in the API or integrate their own models in a simple way.

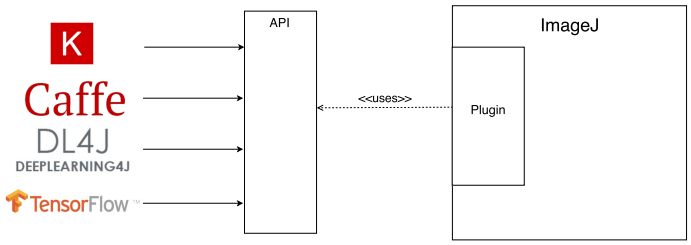

Fig. 1. Structure of the project.

In addition, due to ImageJ is one of the most used image processing programs in research community, specifically in biology, we want to develop an ImageJ plugin to make the use of such an API easier. The idea is that this plugin allows the user to select both the framework and the model used to classify an image and then communicates with the API to obtain the correct class.

4 Preliminary results

As a first step of our project, we have defined an API that integrates the main deep learning frameworks like Keras, Caffe and DeepLearning4J. This API, has one method called *predict* that allows the user to classify an image with one model included in one of these frameworks. Also, we have developed an ImageJ plugin that allows us to use this API in a simple way. An example of using this plugin is included in Figure 2, for classification of bioimages.In this case we have classified melanomas images.

With this plugin, the users only have to select the framework they want to use. Then the plugin shows the models included in the API for this framework. Once, the users choose the model, the plugin communicates with the API and shows the class that the image belongs.

5 Reflections

Today, deep learning techniques are very useful and powerful to solve different image processing problems like object classification, object detection, object

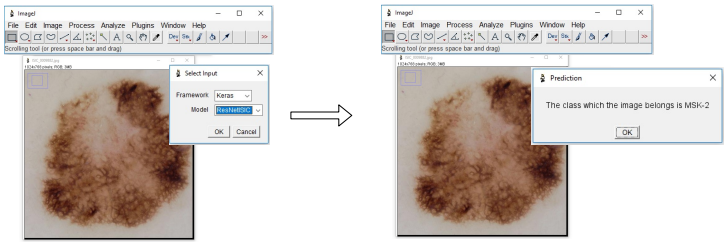

Fig. 2. Interface of the ImageJ plugin for select the input and show the output.

localization or image segmentation. In addition, the use of these techniques is becoming commonplace. Because of our experience we have detected that the use of these techniques is not easy for a common user, this project has been considered to make the use of these techniques easier.

With this project we try to provide deep learning techniques to the image processing users, abstracting deep learning techniques details. The users will only have to select the framework and the model that best suits their problem.

We think that this work will allow users to improve the results of some researches by applying the latest deep learning techniques in a simple way.

References

1. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (2015), http://arxiv.org/abs/1505.04597
2. Nauman, M., Ur Rehman, H., Politano, G., Benso, A.: Beyond Homology Transfer: Deep Learning for Automated Annotation of Proteins. bioRxiv (2017), https://www.biorxiv.org/content/early/2017/07/25/168120
3. Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D.L., Erickson, B.J.: Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions. Journal of Digital Imaging 30(4), 449–459 (2017)
4. Rueden, C.T., Eliceiri, K.W.: The ImageJ Ecosystem: An Open and Extensible Platform for Biomedical Image Analysis. Microscopy and Microanalysis 23(S1), 226–227 (2017)
5. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093 (2014)
6. Chollet, F., et al.: Keras. https://github.com/keras-team/keras (2015)
7. Abadi, M., et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2016), http://arxiv.org/abs/1603.04467
8. Eclipse Deeplearning4J Development Team: Deeplearning4j: Open-source, Distributed Deep Learning for the JVM (2018), https://deeplearning4j.org/
9. Google Brain team: ImageJ/TensorFlow integration library plugin (2018), https://imagej.net/TensorFlow
10. Shuai Yuan: Deep Learning Model Convertors (2018), https://github.com/ysh329/deep-learning-model-convertor
11. Microsoft, Facebook open source & AWS: ONNX: Open Neural Network Exchange (2018), http://onnx.ai/