DeepImageJ: Bridging Deep Learning to ImageJ

Author(s): Estibaliz Gomez-de-Mariscal [1, 2], C. Garcia-Lopez-de-Haro [1, 2], A. Munoz-Barrrutia [1, 2], D. Sage [3]

Affiliation(s):
[1] Bioengineering and Aerospace Engineering Department, Universidad Carlos III de Madrid
[2] Instituto de Investigación Sanitaria Gregorio Marañón, Spain
[3] Biomedical Imaging Group, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Oral/abstract: Talk otherwise poster
Session to submit: Image Data Analysis, Management, and Visualisation

Abstract (Abstracts are to be limited to a minimum of 100 words and a maximum of 1000 words including spaces.)

In the last decade, the use of Deep Learning (DL) methodologies has made a vast improvement in the solution of several bioimage analysis tasks such as denoising, super-resolution, segmentation, detection, tracking, response prediction, or computer-aided diagnosis [2]. These techniques support automatic image processing workflows and have demonstrated potential to surpass human-level performance in common tasks. Consequently, they have a profound impact on the way life-science researchers conduct their bioimage data analysis [1]. Nonetheless, the integration of this breakthrough technology into life-science research pipelines remains still a challenge for the scientific community. Training and evaluating DL models requires previous programming expertise and technical knowledge about Machine Learning. Therefore, the transfer of this technology to the daily practice of life sciences researchers remains a bottleneck. Aware of the latter situation, there are already some pioneer works that target the very need of making DL solutions accessible through user-friendly software [3-7]. Moreover, there is an increasing interest in the bioimage analysis community to teach and learn some general knowledge about DL and democratize it.

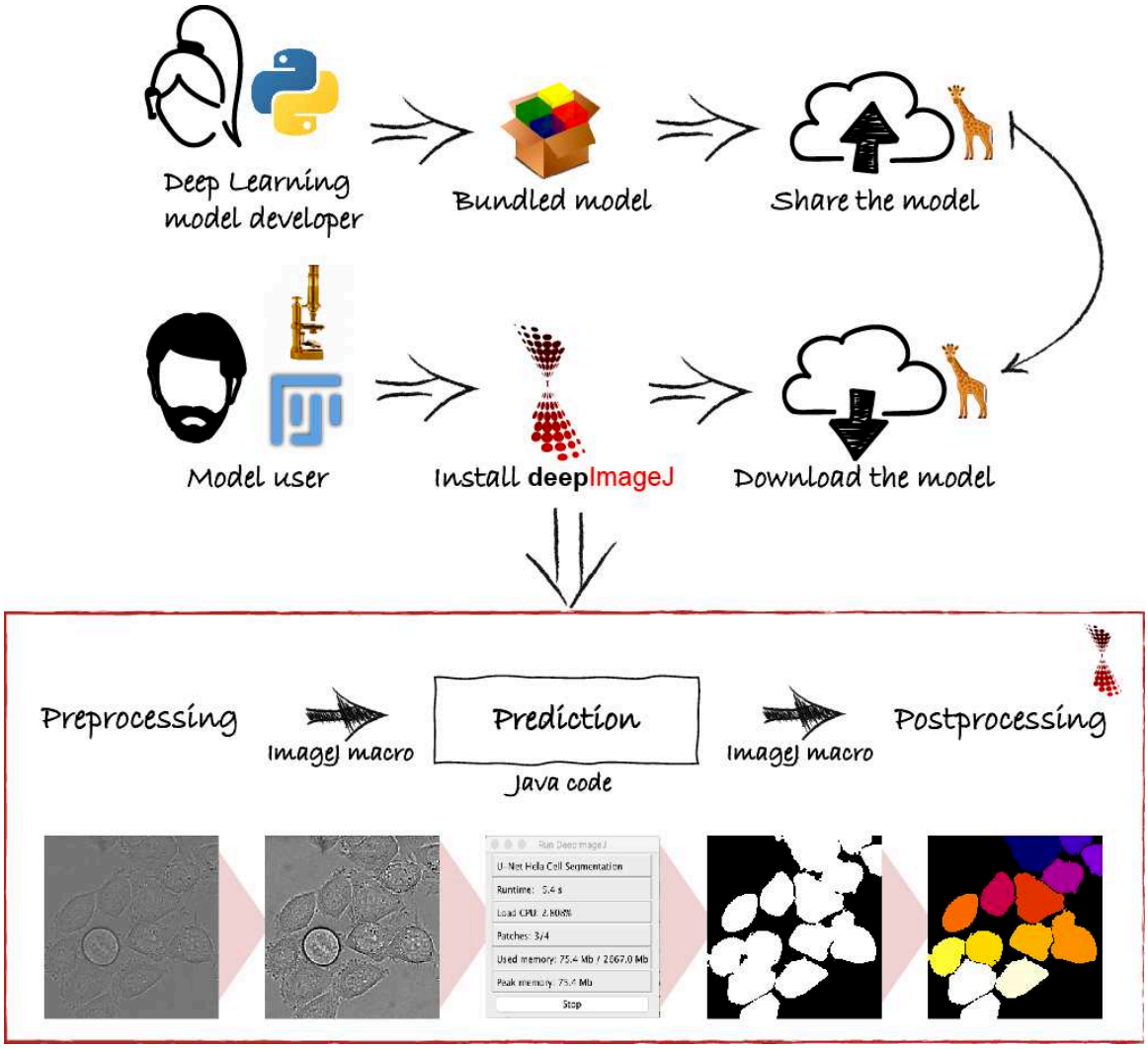

Figure 1: DeepImageJ bridges the gap between Deep Learning (DL) model developers and bioimage analysts. The model developers train their DL models for bioimage processing tasks and export them as bundled models. The format of the bundled model allows uploading it to the BioImage Model Zoo. ImageJ/Fiji users can download any deepImageJ compatible DL model at the BioImage Model Zoo and use it for their image processing tasks. DeepImageJ Run, the ImageJ/Fiji plugin, guides the user in a zero-code fashion through the DL bioimage processing workflow: image pre-processing, prediction, and post-processing are employed as specified by the model developer. The image pre-and post-processing routines are written in ImageJ macros and the model inference, the main functionality of DeepImageJ Run, is programmed in Java. However, the user will process the image with a single click.

We present deepImageJ [8], a user-friendly plugin of ImageJ/Fiji [9, 10] to run trained DL models in a one-click. This is regardless of the DL models architecture or the task for which they were trained. DeepImageJ is designed as a standard ImageJ plugin to deploy DL models through a user-friendly interface. The deepImageJ bundled model format is subject to the specifications defined in the BioImage Model Zoo (https://bioimage.io/) which seeks to define DL models in a standard manner, and hence, contribute to the democratization of DL in the bioimage analysis. This format

gathers all the technical information provided by the model developer in terms of image format, shape, pre-and post-processing steps, or tiling strategies. Moreover, it brings a plain description of the task that the model was trained for and an example image to test its performance. The deepImageJ interface renders all this information in a user-friendly fashion so it can guide the user through the application of trained DL models, Figure 1.

In most cases, the proper use of DL techniques relies on image pre-and post-processing steps. Indeed, the potential of trained DL models is often exploited by additional steps such as in StarDist [5], SplineDist [11], DeepSTORM [12], or DeepWater [13] methods. In deepImageJ, the pre-and post-processing can be written in ImageJ macros and Java. Furthermore, deepImageJ is macro recordable. Hence it can be directly connected to any of the standard and sophisticated bioimage analysis methods available in the ImageJ/Fiji ecosystem, which allows the integration of DL methodologies into more complex bioimage analysis pipelines. A straightforward example of the latter is the recent integration of DeepSTORM [12] for full stacks, DeepWater [13], or Usiigaci [14].

Thanks to the collaboration with the BioImage Model Zoo, life-scientists have access to a wide range of trained models which can be also fine-tuned with existing solutions such as ZeroCostDL4Mic [3] or ImJoy [7], and then, used locally with deepImageJ.

Noteworthiness, deepImageJ has been developed to import TensorFlow (Keras) and PyTorch models, covering the most extended DL environments among bioimage processing developers. To the best of our knowledge, deepImageJ is the only existing solution for the use of generic DL models. Hence, it has the potential to make available many of the powerful algorithms for microscopy image processing that are continuously being developed and published, enhancing research. Overall, we think that deepImageJ fosters a better integration and full exploitation of DL models.

REFERENCES

[1] Meijering E. A bird's-eye view of deep learning in bioimage analysis, Comput Struct Biotechnol J. 2020, https://doi.org/10.1016/j.csbj.2020.08.003

[2] Xing F., Xie Y., Su H., Liu F., Yang L. Deep learning in microscopy image analysis: a survey. IEEE Trans Neural Networks Learn Syst. 2018, https://doi.org/10.1109/TNNLS.2017.2766168

[3] von Chamier L., Laine R.F., Henriques R. Artificial intelligence for microscopy: what you should know. Biochem Soc Trans. 2019;47:1029–1040. https://doi.org/10.1042/BST20180391

[4] Weigert M., Schmidt U., Boothe T., Müller A., Dibrov A., Jain A., Wilhelm B., Schmidt D., Broaddus C., Culley S., Rocha-Martins M., Segovia-Miranda F., Norden C., Henriques R., Zerial M., Solimena M., Rink J., Tomancak P., Royer L., Jug F., Myers E.W. Content-aware image restoration: pushing the limits of fluorescence microscopy. Nat Methods. 2018;15:1090–1097. https://doi.org/10.1038/s41592-018-0216-7

[5] Schmidt, Uwe, et al. "Cell detection with star-convex polygons." International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, 2018

[6] Stringer, C., Wang, T., Michaelos, M. et al. Cellpose: a generalist algorithm for cellular segmentation. Nat Methods 2021, https://doi.org/10.1038/s41592-020-01018-x

[7] Ouyang, W., Mueller, F., Hjelmare, M. et al. ImJoy: an open-source computational platform for the deep learning era. Nat Methods 2019, https://doi.org/10.1038/s41592-019-0627-0

[8] E. Gomez-de-Mariscal, C. Garcia-Lopez-de-Haro, L. Donati, M. Unser, A. Munoz-Barrutia, D. Sage, DeepImageJ: A user-friendly plugin to run deep learning models in ImageJ, bioRxiv 799270, 2019, https://doi.org/10.1101/799270

[9] Schneider, C., Rasband, W. & Eliceiri, K. NIH Image to ImageJ: 25 years of image analysis. Nat Methods 2012, https://doi.org/10.1038/nmeth.2089

[10] Schindelin, J., Arganda-Carreras, I., Frise, E. et al. Fiji: an open-source platform for biological-image analysis. Nat Methods 2012, https://doi.org/10.1038/nmeth.2019

[11] Mandal, S. and Uhlmann, V., SplineDist: Automated Cell Segmentation with Spline Curves, bioRxiv 2020, https://doi.org/10.1101/2020.10.27.357640

[12] Nehme, E., Weiss, L.E., Michaeli, T., and Shechtman, Y., Deep-STORM: Super-resolution single-molecule microscopy by deep learning, Optica 2018, https://doi.org/10.1364/OPTICA.5.000458

[13] Tsai, H.F., Gajda,J., FW Sloan, T., Rares, A. and Shen, A. Q., Usiigaci: Instance-aware cell tracking in stain-free phase contrast microscopy enabled by machine learning. SoftwareX, 2019, https://doi.org/10.1016/j.softx.2019.02.007

[14] Lux, F. and Matula, I. Cell Segmentation by Combining Marker-Controlled Watershed and Deep Learning. ArXiv 2020, https://arxiv.org/abs/2004.01607