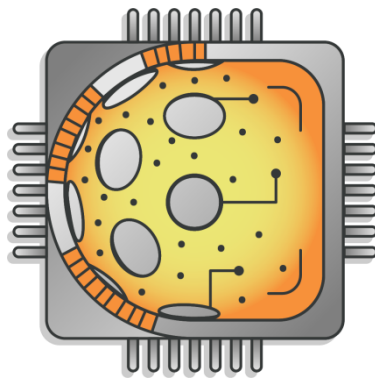


User manual



**Bourel B., Marchant R., de Garidel-Thoron T., Tetard M.,
Barboni D., Gally Y. and Beaufort L.**

1. Presentation of the files and scripts

With these instructions, you will find the files and scripts related to the article: Bourel B., Marchant R., de Garidel-Thoron T., Tetard M., Barboni D., Gally Y., Beaufort L., 2020. Automated recognition by multiple convolutional neural networks of modern, fossil, intact and damaged pollen grains. *Computers & Geoscience*, in submission. The files and scripts were created and tested on Windows 8, and that they are also compatible with Linux and MacOS.

All files and folders presented here are downloadable at <https://github.com/Beniofh/Multi-CNN>. We used the script `Multi-CNN_test_V3.5.2.py` (40.9 Ko) to recognize pollen types using multi-CNNs with augmentation. Multi-CNNs with augmentation is compared to single-CNN with augmentation (script `CNN_test_v2.1.0.py`, 16.5 Ko), single-CNN without augmentation (`CNN_test_v2.1.0 (wihout_augm).py`, 16.5 Ko), and multi-CNNs without augmentation (`Multi-CNN_test_V3.5.2(wihout_augm).py`, 40.9 Ko). `Multi-CNN_test_V3.5.2.py` and `Multi-CNN_test_V3.5.2(wihout_augm).py` are the same script but with different settings. `CNN_test_v2.1.0.py` and `CNN_test_v2.1.0 (wihout_augm).py` are the same script but with different settings.

In the `CNN_solo` folder, we have for each CNN model parameters, a folder with file of CNN model parameter used in .ckpt, histograms of accuracy, loss and learning rate during the training in .bmp and table of accuracy, loss and learning rate during the training in .csv. The folder "intact", "damaged" and "fossil" contain respectively the images for "training dataset"/"intact pollen dataset", "damaged dataset" and "fossil dataset". The part 2 and 3 explains how to use these scripts to reproduce our results.

These scripts have been developed by Benjamin BOUREL (chebenjamin@laposte.net) with the help of Ross MARCHANT. These scripts were first available on the date of publication of this article. They were created in the open-source Anaconda 3 v.5.2.0 (<https://www.anaconda.com>) with Spyder v.3.3.6 in python v.3.6. The details of the python libraries used (name and version) are indicated in part 2. All software and python libraries used are open sources. For optimal use, please use the indicated versions.



The 4 scripts and also, the "CNN_solo", "damaged", "fossil", and "intact" folders must be in the same folder.

2. Integrated development environment and installation

Our scripts were created in the Anaconda 3 v.5.2.0 navigator. We used Spyder v.3.3.6 with python v.3.6. The details of the python libraries used (name and version) are indicated in this part. All software and python libraries used are open sources. For optimal use, please use the indicated versions. Instruction for installation :

1. Install Anaconda 3 version 5.2.0

2. Open "Anaconda prompt" and use the commands :

a. `conda create --name tf-3.6 tensorflow python=3.6`

-> create new environments named "tf-3.6"

-> with all necessary modules for use Tensorflow

-> and with python version 3.6 (this the optimal version for our script)

b. `yes`

- >confirm the installation of the modules

3. Close "Anaconda prompt"

4. Open "Anaconda Navigator", load the environments "tf-3.6" check the presence and install (if not present) : - name in command "import"/"from" -> name in Anaconda Navigator

- glob -> glob2 (v. 0.7)

- itertools -> integrated into command-line interpreter

- Keras -> keras (v. 2.2.4)

- keras.preprocessing -> keras-preprocessing (1.1.0)

- math -> mpmath (v. 1.1.0)

- matplotlib -> matplotlib (**v. 2.2.2**)

- numpy -> numpy (v. 1.16.5)

- os -> integrated into command-line interpreter

- PIL -> pillow

- skimage -> scikit-image (v. 0.15.0)

- sklearn -> scikit-learn (v. 0.21.2)

- shutil -> integrated into command-line interpreter

- sys -> integrated into command-line interpreter

- tensorflow -> tensorflow (v. 1.14.0)

5. Launch Spyder v. 3.3.6 and drag-and-drop the scripts to use into the Spyder window

3. Use scripts to reproduce our results

For the 4 scripts (files in .py) proposed, we have just to select our dataset ("intact" or "damaged" or "fossil") in the part Editable Setting of script. To do this, change the variable switch1 for Multi-CNN_test_V3.5.2.py or Multi-CNN_test_V3.5.2(wihout_augm).py and the variable switch2 for CNN_test_v2.1.0.py or CNN_test_v2.1.0 (wihout_augm).py.



If you downloaded the files directly to <https://github.com/Beniofh/Multi-CNN> you will need to delete three .txt files named "Delete_file.txt". These three .txt files are respectively in :

- \fossil\al\Delete_file.txt
- \fossil\ca\Delete_file.txt
- \fossil\co\Delete_file.txt

It is because, in GitHub, we cannot commit empty folders (GitHub does not actually save folders, only files). But script needs three empty folders (\fossil\al, \fossil\ca and \fossil\co). We had to add a .txt file inside these three folders to get them into GitHub.

PS : Normally, these three files " Delete_file.txt" don't interfere with the execution of scripts but it's safer to delete them.