**Procedure for environments :**

* Install visual studio code
* Install anaconda or miniconda
* Open anacona prompt
* Write in the prompt: “conda env create -n ENVNAME --file ENV.yml” to create a conda environment from a .yml environment which contains the important libraries for the project (if this step fails, you need to install the libraries one by one with pip install LIBRARY\_NAME → see below)
* Write: “conda activate ENVNAME” (to see if the environment exists)
* You can also list all the environments with “conda env list”

**In visual studio code:**

1. **Open Visual Studio Code**: Launch VS Code on your machine.
2. **Open Your Project**: Open the folder containing your Python project by using the "File > Open Folder..." option.
3. **Select the Python Interpreter**: VS Code needs to know which Python interpreter to use for your project.
   * Open the Command Palette (you can use the shortcut **Ctrl+Shift+P** on Windows/Linux or **Cmd+Shift+P** on macOS).
   * Type **Python: Select Interpreter** and select it.

The environment that you created should appear and by clicking on it, it will activate it for all the files.

**Launching code with environments:**

* If everything worked, you should be able to run the different notebooks if you select the correct environments.
* If you still have a message like NameError: name 'LIBRARY\_NAME' is not defined, you need to install the missing library by opening anaconda prompt, activating the environment and type: pip install LIBRARY\_NAME

**For classic machine learning:**

I created a notebook called **“pipeline\_classic\_ML.ipynb”** which shows briefly how to use the different models (xgboost, catboost, randomforest). Each model has a “best parameter finder” and a “test model” part.

**For deep learning:**

There are several notebooks for deep learning which have overall the same structure but some small changes depending on the needs.

**List of the notebooks:**

* **Augment\_dataset**
  + Used to precreate the augmentation of the images (deprecated since the last versions creates the augmentation dynamically inside the deep learning pipeline)
* **Cellpose\_segmentation**
  + Used to test different segmentations from cellpose
* **Create\_gif**
  + Used to create the gifs to visualize the temporality of the images → E:\Laurent\New\_data\_Maxim\gifs
* **Crop\_maxim\_images**
  + Used to crop the images to eliminate the black borders
* **DL\_classifier\_CNN notebooks**
  + All these notebooks are different trials to enhance the performance of the deep learning pipeline: the most classic one is DL\_classifier\_CNN\_CV\_new\_data. In this notebook there is not many regulations to prevent regulation. Remember that the parameters to change are mostly:
* num\_epochs = 50
* learning\_rate = 5e-5
* batch\_size = 64
* k\_folds = 2
* weight\_decay = 1e-5

**Don’t hesitate to ask ChatGPT for advices.**

* **Extract\_features notebooks**
  + Used to extract the features from the images: takes as input the images, segment them and output a csv file with the features
* **Feature\_importance\_across\_models notebooks**
  + Used to analyze feature importance and UMAP and TSNE plots
* **LeNet**
  + Mariia’s pipeline
* **Pipeline\_classic\_ML**
  + Used to perform classic machine learning models (xgboost, catboost, random\_forest)
* **Read\_czi**
  + Used to read the czi files and extract the images
* **Tiff\_to\_png**
  + Used to transform set of images from tiff to png and also pipeline for YOLOv8 model (see roboflow documentation on the website)
* **Time\_line\_analysis**
  + Used for the new architecture of the deep learning pipeline: DL\_classifier\_CNN\_CV\_new\_architecture
* **Z-projection**
  + Used to create the z projection

**List of the python files:**

* **Classic\_ML**
  + Used for the pipeline\_classic\_ML notebook
* **Utility**
  + Contains a lot of functions used in the different notebooks