Quickstart guide

The purpose of this guide is to allow you to get started with setting up the project and getting both the NST and Densisty statitics pipeline to work on your machine

In depth explanation of the inner workings of the pipeline are availiable on both the pipeline's README.Md and documentation present in the documentation folder

Table of Contents

Starting up

Tutorials regarding conda installation are availiable on the following path (V:/Studies/_tutorials/ coding enviroment). In order to avoid packages/dependency issues I *highly* reccomend you create a virtual enviroment

Required packages to run the pipeline are availaible in the environment file enviroment.yml These are the files that as of this document's writing (31-07-2025, handover between Gian and Alex) are needed to run both the CDxRLT pipeline (occasionally referred to as the stats pipeline) and James' computer vision pipeline (alternatively referred to as the NST pipeline) but you might find more packages are necessary: in case you need to confirm all the past enviroment files i have been given are in the oldenv folder.

if your enviroment has been created correctly, this, along with access to the V:/ and P:/ drives are about all you should need to strat running code

CDxRLT Pipeline

This paragraph represents a very quick rundown of the pipeline, how to use it as well as what you get out of the single steps (mainly figures, as i suspect you will be working quite a lot on those).

Again, refer to the PRxRLT guide for more details as into the workings of the pipeline. In the same folder you will find the READMES folder with info on the single steps of the pipeline, and the index.html file that acts as a sort of flowchart detailing the inputs/outputs of all the (main) components of the pipeline.

NOTE: in some of these files you will find lines such as

```
sys.path.append(r'U:\Repos\aoslo_pipeline')
```

These are to make sure the file can find the src folder when importing methods and functions form files that are in said folder: change it to point at the root folder of your project (wherever you cloned the repo essentially) if you get an

```
ImportError: package src not found
```

0: (Optional) Extract Layer Features:

Start by running the following notebook save_layers_feature.ipynb: this will extract from cohortbuilder the data regarding the layers and foveal parameters that you get from the oct scan

There is no need to rerun this every time, as the results are all already saved and ready for usage in the rest of the pipeline.

1: Run the pipeline proper:

Pipeline starts from the density_pipeline_manager.py file: running this is all you need to perform the analysis you need to get the correlations/figures you see in the paper The parameters of this pipeline, which can be tweaked in line 241

```
try:
    main(ses_path, do_montage = False, do_layer = True, from_csv = False,
from_csv_raw = True, to_csv_dens = True, subject = subject)
```

Of these parameters we have:

- do_montage: wheter or not to call the montage pipeline, to strat form the very beginning:takes a lot of time, you should not set it to true unles sabsolutely necessary
- do_layer: wheter or not to analyze the layer results obtained in step 0: if you remember the whole triangle thing, this is what we are talking about
- from_csv: wheter or not to extract the dneisties from the density.csv file pèresent in the density_new folder for each subject (and obtained from a previous run of the pipeline). if ste to True this skips both the Fourier transform cone counting process and the fitting of dernsities on our model
- from_csv_raw: wheter or not to extract Raw densities form the raw_densities.csv file located in the density_new folder: keeping this on True allows to skip the fourier coiunting of cones, also quite time consuming. Setting it to False necessarily requires setting do_montage to True, as the image sneed to be realigned with the reference frame before counting the cones again.
- to_csv: wheter or not to save the results of the pipeline to csv: useful to keep False if experimenting

2a: Correlation Notebook

This file is part of the pair of notebooks running the correlation analysis on the results of the pipeline run in the previous steps: in particular the focus is on obtaining correlations between cone densties at different ecectricities (for example figuring out if having more density in the center is correlated with having more density in the periphery) or between cone density and foveal parameters (for example if having a broader cone density peak is correlated to having a broader foveal pit)

2b: CdxRLT Notebook:

File n.2 of the notebooks, this one is focused ont correlations between the cone density and the thicknesses of specific retinal layers (for example: is the cone density at eccentricity 2 degree correlated with the thickness of the OS layer at the same eccentricity? does having more density there mean that we have a thicker OS layer?) as well as other generatory analysis of the cone density distribution

EXTRA: FIGURES MAP

As of this writing, these are the figures present on the paper and how to obtain them

- Figure 1 : Hand Drawn
- Figure 2 : correlation_analysis.ipynb Correlation analysis section
- Figure 3 : CDxRLT_analysis.ipynb ecc-wise correlation sections
- Figure S1 : Montaging step, images can be found in (P:\AOSLO_automation_PROCESSED\Photoreceptors\Healthy_Results\SubjectXX\SessionYYY\montaged_corrected)
- Figure S2 : Montage as above, for the ROI (red square) they can be found in (V:\Studies\AOSLO\DensityExtraction). I have removed the code to obtain these images for cleanliness since it was debug code, but i have an old copy on my poc, contact me if you need it
- Figure S3 : A mixture of screenshots form discovery, and images can you can obtain by uncommenting the debug code present in the os_layer_extraction and running the save_layers_feature.ipynb again
- Figure S4 : Can be obtained through the VISUALIZE_FOVEA script, who's gonna svae an html 3d represnetation fo the fovea as seen in the pciture
- Figure S5 : CDxRLT_analysis.ipynb Density Analysis and Comparison with Zhang 2015 sections

NST Pipeline

Again a very quick rundown of the pipeline, going from start to the figures present in the NST paper, more complete documentation can be found in here

I should have solved most problems that could stop the pipeline form running on your machine, but i probably missed some into merging James' branch into mine: feel free to contact me if any new ones come up

To run James' computer vision pipeline you need to follow two steps

0: Check the configuration

Before running the pipeline in the next step it's important to chekc everything is in order in the configuration file: in this file you can find the input and output folders, as well as the steps of the pipeline to perform, such as

```
- __do_mdrnn=True: wheter to perform mdrnn cone detection or not
- __do_nst = True: wheter to perform NST preprocessing BEFORE doing mdrnn cone
detection
- __do_density_analysis = False: I am keeping it false since the analysis pipeline
has been moved to the one me and Alexandre have developed, but might be useful if
one day you feel very brave and want to integrate everything into one giant
monster of a pipeline
```

Everytime you see a Parser call or other parameter that is imported, chances are it comes form this file

1: Start the pipeline

The pipeline entry point is the file Start_PostProc_Pipe.py which is by itself essentially a call to the pipeline runner which handles the various steps of the pipeline based on the parameters passed from the configuration file The pipeline engine calls the MDRNN Pipeline class which then calls in its preprocessing the NST Pipeline and NST Class.

The MDRNN pipeline works on patches (essentially crops of images) which can either be preprocessed with a number of ordinary methods or through NST, in which case the NST pipeline will get the patches as inputs and output the corrected patches back toi the mdrnn pipeline to perform cone detection

The results of the pipeline are saved, for each subject, in the folder postprocessed As far as i can tell reuslts form the previous run of the pipeline have been moved here

V:\Studies\AOSLO\AOSLO25\usb\black\data_hospital\images_resized_128\output_cone_detector_mdrnn\algorithmMarkers

Or the equivalent for the cunefare etc: you can find the path as the inputs of the next step

Note: several paths are hardcoded for James' machine such as:

```
sys.path.append(r'C:\Users\BardetJ\Documents\aoslo_pipeline')
```

I suggest you perform a search with the VSCode search function for BardetJ and change the path for your equivalent

2: Get the paper Images

All images can be obtained through the following notebook [all_paper_graphs.ipynb](src/cell/cell_detection/nst/paper/all_paper_graphs.ipynb)

The notebook can be run without having to run the pipeline again, as all results/ input files are already saved as csv in the AOSLO_2025 folder (located in V:\Studies\AOSLO\AOSLO25) or in the Reuslts folder (P:\AOSLO_automation_PROCESSED\Photoreceptors\Healthy_Results).

OTHER