make_montage Getting Started Guide

Table of Contents

1. Installing the Anaconda Environment (skip if you already have a python interpreter)

1.1. Install Anaconda

The make_montage.py script is written in python. A python interpreter is required to execute it. If you don't already have one, I recommend installing the Anaconda Platform, which is an open-source (free) package and environment management system (https://www.anaconda.com/products/distribution). It is available for Windows, Mac, and Linux. Follow the installation instruction provided on the following website: https://docs.anaconda.com/anaconda/install/.

1.2. Create the Anaconda Environment

Once you have Anaconda installed, open an Anaconda prompt (https://docs.anaconda.com/anaconda/install/verify-install/):

- **Windows:** Click Start, search for Anaconda Prompt, and click to open.
- **macOS:** Use Cmd+Space to open Spotlight Search and type "Navigator" to open the program.
- **Linux–CentOS:** Open Applications > System Tools > terminal.
- **Linux–Ubuntu:** Open the Dash by clicking the Ubuntu icon, then type "terminal".

To execute a python script, you must first install the required modules. I suggest installing the required modules in an isolated conda environment. On Windows, create the conda environment in one of the following ways:

Method 1 (create a conda environment from a yml file – easier):

1. On your computer, navigate into the folder where make_montage_env.yml is saved.
2. Type:

   conda env create -f image_analysis_py39_env.yml
3. For detailed instructions, please refer to: https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-from-an-environment-yml-file

Method 2:

1. First, create a new environment called, *e.g.*, image_analysis_py39 using the command:

conda create --name image_analysis_py39 python=3.9

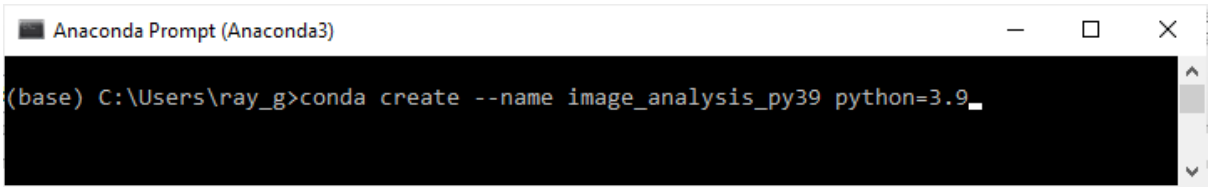

Figure 1. Create a conda environment.

2. Activate the environment using the command:

   conda activate image_analysis_py39

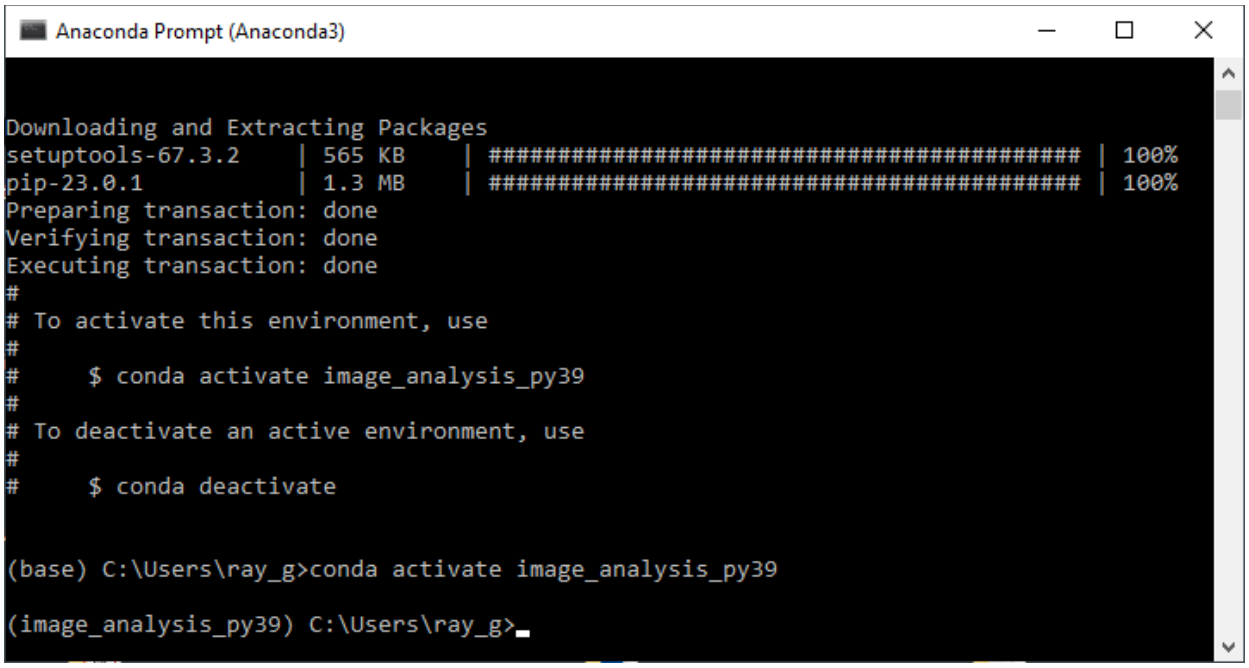

Figure 2. Activate the newly created conda environment.

3. To install the required modules, type:
   - pip install matplotlib
   - pip install numpy
   - pip install PyQt5
   - pip install easygui_qt
   - pip install argparse

4. To verify that the required modules have been successfully installed, type:

   conda list

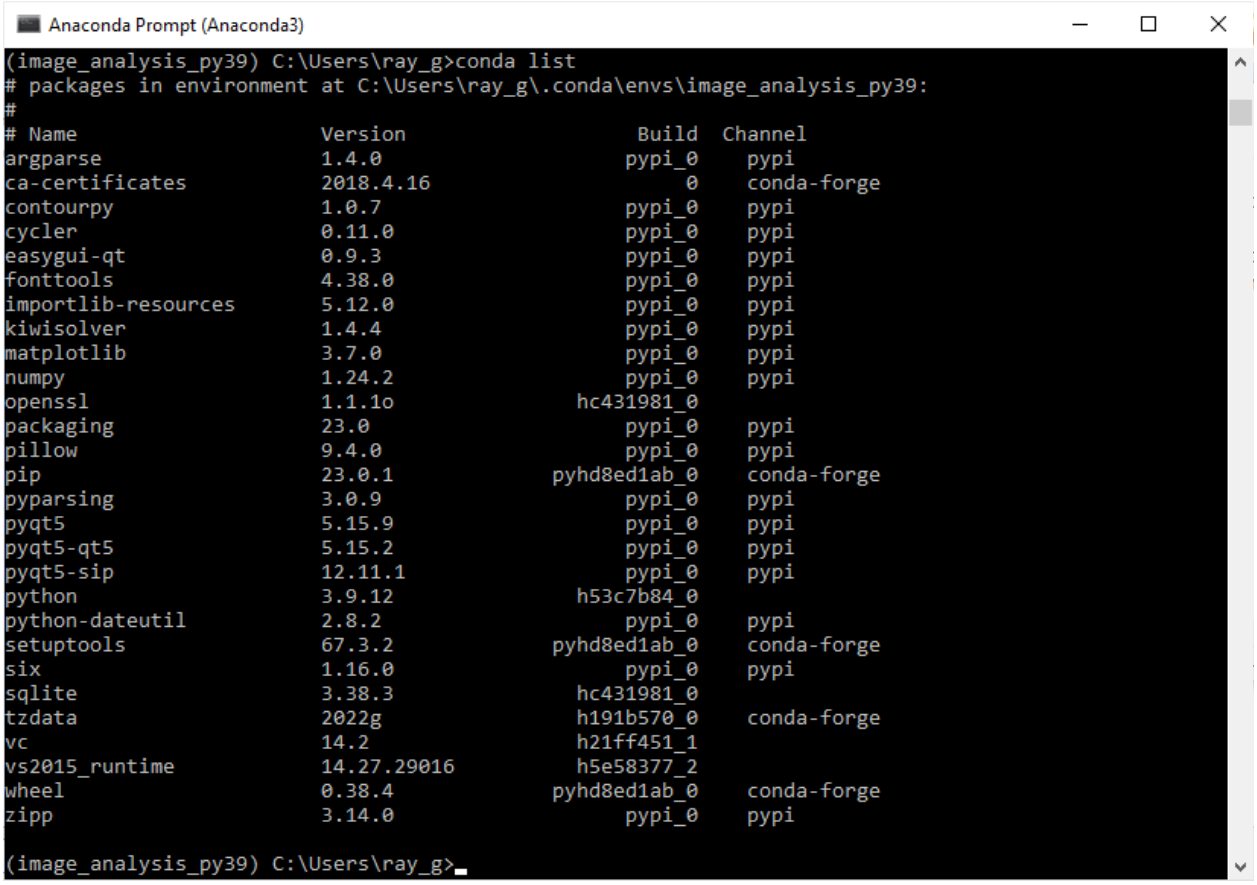

Figure 3. Review installed modules within the image_analysis_py39 environment.

5. To run make_montage:

   - First, go to the directory where the script is saved:

     cd C:\Users\ray_g\Desktop\make_montage_demo

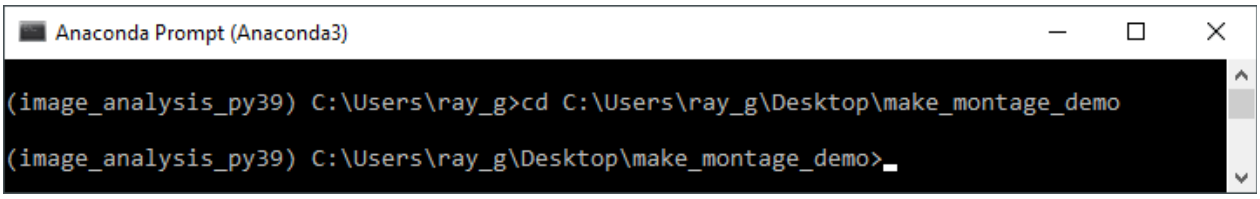

Figure 4. Navigate into the directory where make_montage is saved.

- To review the parameters for running the script, type:

  python make_montage.py –help

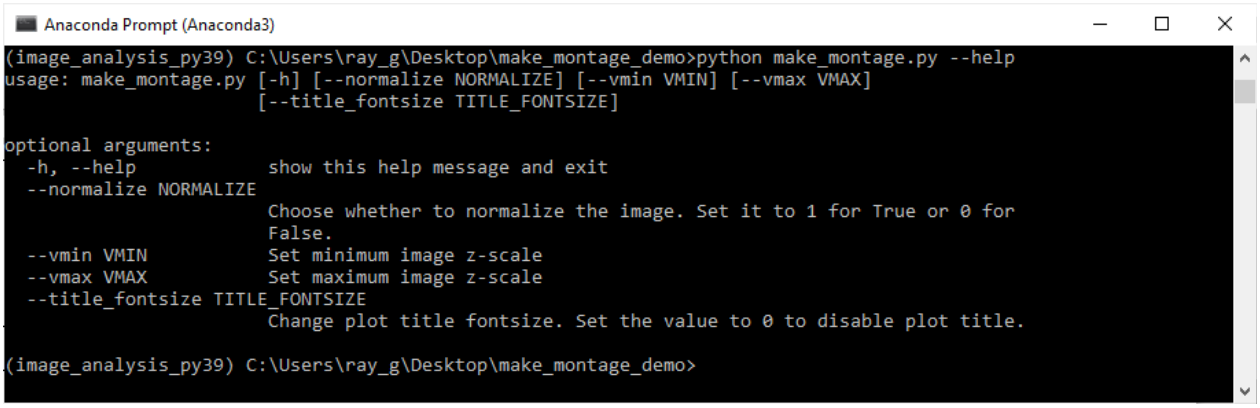

Figure 5. Review make_montage.py parameters.

6. (Optional) Having successfully created the image_analysis_py39 environment, you can save this environment into a yml file using the command:

   conda env export > image_analysis_py39_env.yml

   You may use this yml file to duplicate this environment on another computer as described above. You can also customize the name of the environment within the yml file.

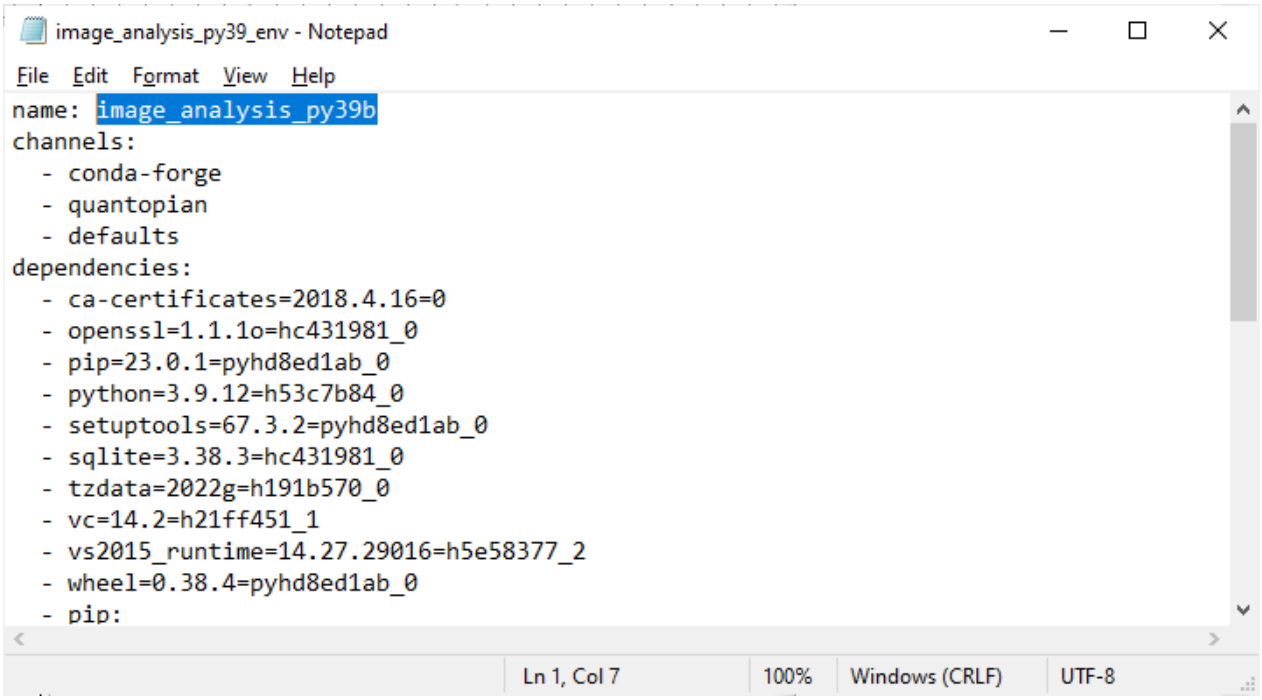

Figure 6. The content of the image_analysis_py39.yml file.

Removing a conda environment

If for any reasons you will have to remove a conda environment, use the command:

conda deactivate

Followed by:

conda env remove -n ENV_NAME

e.g.,

conda env remove -n image_analysis_py39

2. Usage Demonstration

The make_montage.py script sorts the sequence of images in one of two ways:

(a) Assumes a four 2-frame cameras which sorts the images as: C1-frame1, C2-frame1, C3-frame1, C4-frame1, C1-frame2, C2-frame2, C3-frame2, C4-frame2. The script is as follows:

```
if not i%2: #if odd
  plt.subplot(2, int(no_imgs/2), int((i+2)/2))
else: #if even
  plt.subplot(2, int(no_imgs/2), int(4+(i+2)/2))
```

(b) Assumes a set of sequential images which sorts the images in ascending order by date and time. The script is as follows:

```
plt.subplot(2, int(no_imgs/2), int(i+1))
```

By default, the script is configured to (a). To change the sorting sequence to (b), go to def make_montage() section of the code, comment (b), and uncomment (a).

2.1. Make a montage of the raw images

1. To make a montage of the raw images, type:

   python make_montage.py --normalize False --vmin 0 --vmax 100 --title_fontsize 8

2. When prompted, select the main folder containing the images saved within a subfolder called "Data" and click "Choose".

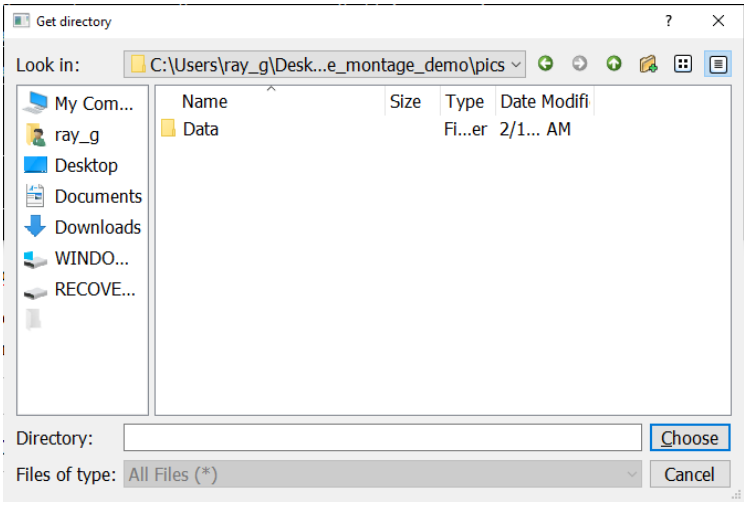

Figure 7. Select directory where pictures are saved.

3. Using the examples image provided in the Data folder, the result is:

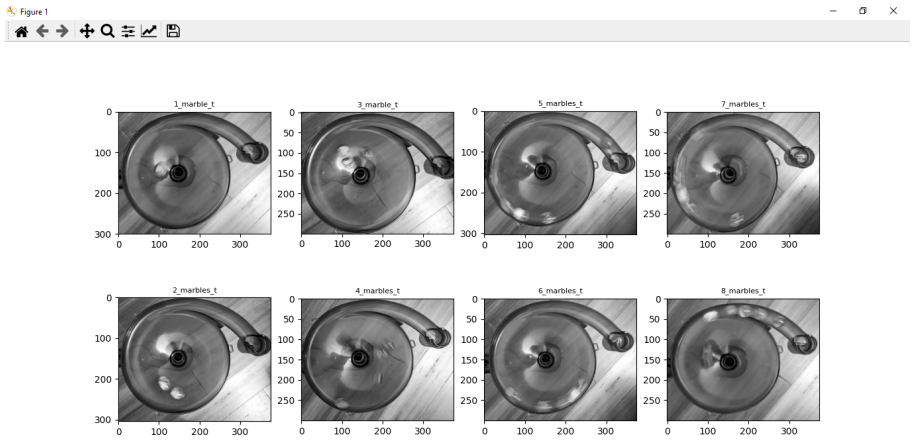

Figure 8. An example of make_montage application to a set of eight images without the flat-field normalization. Note that the image sequence is 1,3,5,7,2,4,6,8. In the script – within the make_montage() function – you have the option to change the sorting to sequential.

4. Close the figure window, restart the script and adjust the vmin and vmax values to vary the contrast.
5. If you like the montage, you can save it by clicking on the save button from the toolbar.

2.2. Make a montage of flat-field normalized images

1. Prepare three set of images: raw images, whitefield, and darkfield.
2. Place each set of images into three separate folders called:

   “Data”, “WF”, and “DF”, respectively.
3. To make a montage of the flat-field normalized images, type:

   python make_montage.py --normalize True --vmin 0 --vmax 1.5 --title_fontsize 8
4. When prompted, select the main folder containing the images saved within the subfolders called “Data”, “WF”, and “DF” and click "Choose" (not shown).
5. Close the figure window, restart the script and adjust the vmin and vmax values to vary the contrast.
6. If you like the montage, you can save it by clicking on the save button from the toolbar.

~