ConvexiPy

Laurence Lu

December 19, 2020

1 Prerequisites

Before installing, it is crucial that the Python version you install is exactly 3.8.5, this is available here. Later versions do not work with some of the packages, and I have not tested for any earlier versions, making 3.8.5 the safest.

1.1 Preliminary notes

As a side note, for editing or viewing the code, please use Visual Studio Code or just the built in text editor for your computer.

1.2 Python installation

When installing Python, we should simply select the "Install now" button from the installer, however if one does not wish to use that option, there are some options that are required.

On Windows, we should select the option to "Add Python to PATH" and the option to install pip if possible. We will also need to select the "tcl/tk and IDLE" option to install the tkinter library.

1.3 Python package installation

There are a number of required packages, which can be found and installed using the 'requirements.txt' file found inside of the ConvexiPy folder. To do this, on a Windows system, copy and paste the following commands into a command line (each line in the below listing is a single command):

```
cd C:/<path>/<to>/<dir>/ConvexiPy
pip install -r requirements.txt
```

and the 'pip' package manager will handle the download/installation process. If an error is returned, one may also try

```
cd C:/<path>/<to>/<dir>/ConvexiPy
pip3 install -r requirements.txt
```

which differs by using the 'pip3' command instead of 'pip'. If one still receives errors, you should try installing the packages manually. For this, we will refer to the following list of packages, which the program relies on:

- opencv-python
- pathlib
- Pillow
- scipy
- numpy
- matplotlib

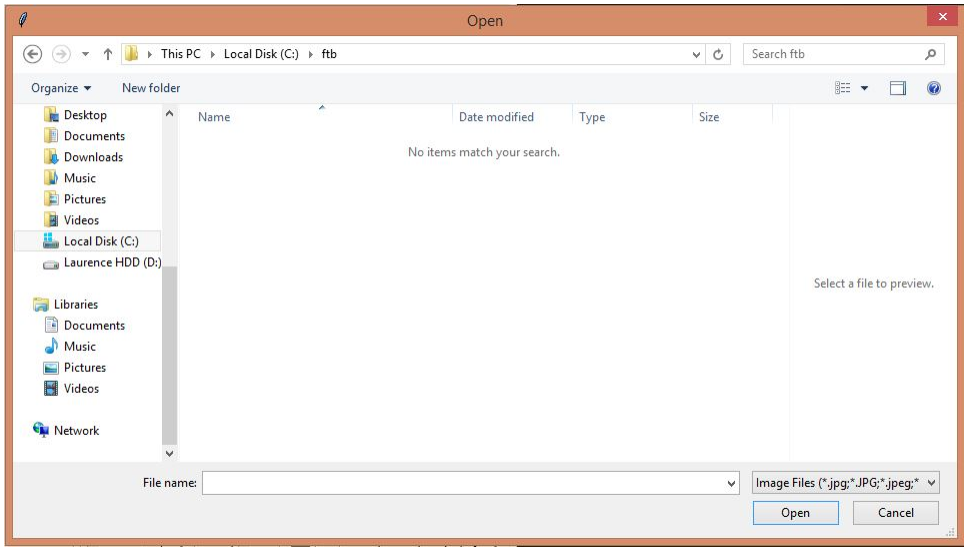
Each package may be installed using either 'pip3' or 'pip', depending on whichever worked earlier. If you continue to have any errors, please contact me via e-mail.

2 Using the program

In order to use the program, the 'main.py' file should be run using the command

```
python main.py
```

Once the program runs, you will see a prompt like this:

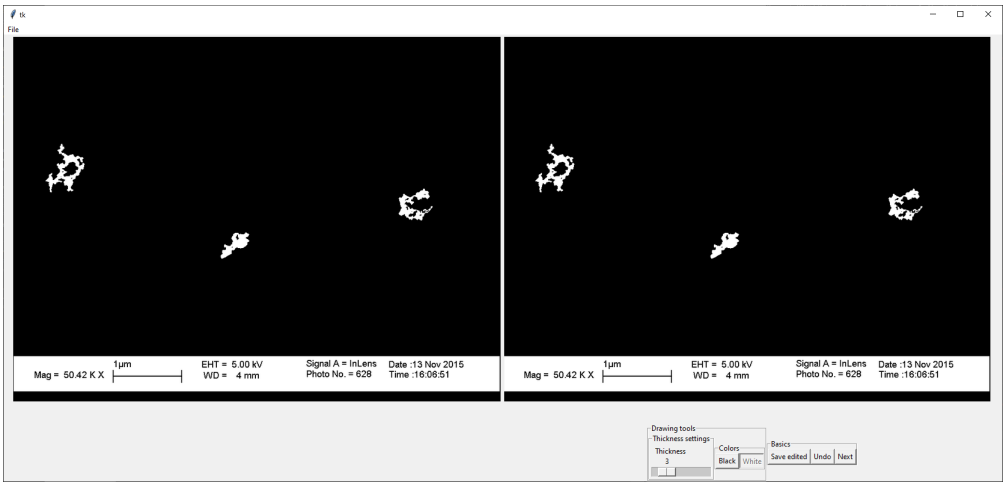


IMPORTANT NOTE: this dialog will allow you to select a file from *any* directory, you just need to select the file you want.

The first prompt after is the black/white conversion step.

You will have a slider on the bottom left, this will let you determine the particular threshold for the conversion. When you are satisfied, you may click the 'next' button.

The next dialog is the retouching step, where you will be able to draw in either white or black pixels (and change the thickness of the pen as you need it).



When you are satisfied, you may click the 'next' button.

Once you have done that, you will see the window below:

Now, we can crop. All you need to do is click and drag over the region that you want to crop, and you will see a red rectangle selected, like so:

You can crop **multiple regions**, you just need to click and drag over a different region.

If you wish to eliminate a region you have just cropped, you can only remove the last one you have cropped using the 'Undo' button. If you wish to undo *all* of the regions, you may click the 'Reset' button.

The rest of the buttons, I don't really know what they do exactly. However, the source code for the original library creating the cropping window may be found by clicking right here.

To complete the crops, click the 'Crops' button. The window should now close, and the program will generate a csv file containing the filename of the cropped images, the convexity value, the area (in pixels) of the projected aggregate, and the area (in pixels) of the convex hull thereof. There will also be the binarized (black/white) cropped images, and the cropped images alone (without modification other than cropping). A copy of the image with the convexity values superimposed, near the cropped areas, will also be generated to the output folder in the working directory.

3 To do

1. Allow user to find largest dimension of perpendicular width/length of a particle
2. Given a scale bar, deduce the real-life values for area/dimensions.
3. Implement roundness
4. Add lasso tool to work with images with substrates