Library Dependencies

In order to install the required packages one will need the "pip" package manager which is included into your Python distribution if you have the official version by python.org or an Anaconda Virtual distribution.

To access pip please head to your command line (if you are using Anaconda then you must use the Anaconda Command Line.

Install the following packages by entering these commands into the command line environment of your python installation:

- pip install numpy
- pip install pillow
- pip install opencv-python
- pip install imutils
- pip install psutil **(To force Python.Multiprocessing to recognize only Physical Cores)**

The rest of the utilized libraries (multiprocessing-physical, Tkinter, sys, and etc.) are internal Python modules.

Open_CV Colorization Model Files

The 2 files given below will be uploaded to the Github repository and are required to be stored in the same folder as the Python script:

- colorization_deploy_v2.prototxt
- pts_in_hull.npy

The following file containing the model parameters must be downloaded from the link below and saved in the same folder as the Python script as well:

- colorization_release_v2.caffemodel

  Download Link (Click "GO TO SITE" to download the file):
  https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbkt6SVEtR1pKdWFUUjNKYThwSUdTSFdNRW85Z3xBQ3Jtc0trb2xJeUJHbUFHUzQ5d3BXS2Y0V2l3MDBRRGE2NGJrdG1IUWhBOUpiTGpNblV5TllpS3F6NFdZVU50R3A5ZHdrN0xGX2Fwd0o1YWRsUjBlRkRFNFF2MGZNWHdYVEFuWlNDbEEyOF90ZkVpaUxZVnR5Zw&q=https%3A%2F%2Fwww.dropbox.com%2Fs%2Fdx0qvhhp5hbcx7z%2Fcolorization_release_v2.caffemodel%3Fdl%3D1&v=gAmskBNz_Vc

Command Line Arguments (important)

Entering these command line arguments are mandatory to launch the script. The application will take three arguments from the command line:

filename, square size and the processing mode

Example: yourprogram somefile.jpg 5 S

- **file name:** the name of the grayscale image file in jpg format (no size constraints)
- **square size**: the side of the square for the averaging (The bigger the better)
- **processing mode:** 'S' - single-threaded and 'M' - multi-threaded

Launch modes (important)

If you run the application with the "S"(single-threaded) key a Tkinter GUI window will be launched and by pressing the "START" button you'll be able to track the progress of colorization in real-time.

If you run the application with the "M" (multi-threaded) key then a command line interface will notify you of the state of the program. In order to track the progress of the parallel (multithreaded) program please go to the folder in which the .py script is stored and view the tempo_file_XX.jpg files.

Either way the final result will be stored in the "result.jpg" file located in the same folder as the Python script.