Last update: 1/30/2020

User Download Guide for Automated Parenchyma Analysis on CT data

Acil Applied Test Imaging Laboratory

Overview

The following guide outlines the process for downloading DockerHub and setting up an automated version of CIP Slicer for carrying out parenchyma analysis on lung CT scans. Setup should take 30 minutes.

Setup Steps

1. First navigate to the Docker Hub website at the following address: https://hub.docker.com/ and click the “Get Started With Docker Desktop Button”, and then “Download Docker Desktop for [Your Operating System]”.
2. After this download runs, click the installer and follow the steps to install docker.
3. From here you should be able to run docker from your command line.
4. To test this navigate to your command line (you can search for this as cmd in your computer's search bar), and type `docker` . If docker is properly installed a long list of commands will appear in the terminal. If docker is not installed properly your terminal will return “ 'docker' is not recognized as an internal or external command, operable program or batch file.” Refer to the docker documentation to troubleshoot problems.
5. Navigate to your system’s command line and type the following command to download the Chest Imaging Platform: `docker pull acilbwh/chestimagingplatform` (don’t close the command line, this may take a minute)
6. After this, it is time to create a Docker container for the Chest Imaging Platform (CIP) software.

Note: A container is essentially a Virtual Machine that runs on your computer independently of your system, and it is the easiest way to run the CIP software without having to download all of the different library requirements. Since the container is running independently of your system, you will need a means by which to exchange files with it. To do so we will need to create a mount which docker refers to as a volume. This is a file that can be accessed by your machine and the container being used.

a. Create a folder (mount) anywhere on your system where you wish to store the files being used by CIP. This can be any folder, anywhere.

b. To create the container and link the folder run the following command: `docker run -ti --name Slicer -v your_local_folder_path:/host -d` `acilbwh/chestimagingplatform` replacing your_local_folder, with whichever folder you chose in the previous step. An example may be "C:/Users/JohnDoe/Documents" (in this case the computers documents folder). You will need to include the entire folder path to properly mount the system. Now whenever you put a file into that folder the container will be able to see and use it.

7. Now that the container is running, place the Dicom files which you wish to run analysis on into a folder titled "DicomDataFiles" (case sensitive) inside of the folder created in step 5a.
8. Next download the **autogen.py** file from the following link (click on the clone and download link then the Download zip button to do so): https://github.com/NthW/automatedslicer, and place it into the same folder from 5a. Make sure you unzip the file before running.
9. Now you're all set to begin running your analysis! To do so begin by entering `docker exec -ti Slicer python autogen.py` into your terminals command line.
10. From here you will be asked which functionality of the script you wish to use. Typing 1 will start the full analysis, 2 will restart the analysis, 3 allows you to start the analysis from the scan you left off at (if the computer shut down), and 4 allows you to combine all of the result files into a single csv file.
11. Type 1 if this is your first time running analysis then hit enter.
12. From here the code will begin to run through each scan and run the analysis. If you need to stop the code for some reason exit the terminal.
13. Once the script has finished running, to combine all of your output files (since they were all separately generated for each CT scan) run the same command from step 8, but when prompted type 4. This will create a csv file with all of your results, which will pop up in the folder you created in 5a.

Note: The script will take 3 to 8 minutes per scan depending on the system being used to run the scans. For reference, the code will take 2 days to process 500 scans.

File Placement/Structure

In order to ensure the script runs correctly the following file format should be used. Here the Folder names (Scan1Folder) should correspond to your patient identifiers. Make sure there are no spaces in your file names.

```
your_local_folder_path
        autogen.py
        DicomDataFiles
                Scan1Folder
                        (all Dicom Files)
                Scan2Folder
```

(all Dicom Files)
....

All of the patient scans should be placed in the DicomDataFiles folder and have all the Dicom files exist in that folder. The script will look for where your scans are within the folders, and if you are not careful it may end up running a different series than you expected. *Be sure to understand how your files are being given to you from PACS or whichever storage your institution uses.* If you fail to do so the script will fail.

Good Practices with Code/Restarting

The automater is a working prototype, and as such there may be some problems you encounter when running the code. The code runs all files from start to finish, and if the computer is restarted or the code is stopped for some reason, there is potential for the container to be significantly slowed due to an uncleared cache. To keep this from happening it is recommended that the following be observed:

1. Try to let the code run from start to finish as interruptions can be troublesome. If you do need to stop the code midway through a run, you should take note of the number of the last processed scan, before exiting the terminal.
2. After this it is recommended that you restart the container to make sure that the old session does not cause any issues.
    a. To restart simply type `docker restart Slicer`
    b. Then type the normal set of commands to run docker.
    c. If the restart command doesn’t run, restart your entire computer and try again. This should fix the problem.

Storage

CT scans can be quite cumbersome in terms of storage, especially on smaller machines. Loading scans in batches is an easy solution to this problem. Once they have been processed, a .nrrd file (with the same name as the inputted folder) is produced, allowing for the initial scans to be deleted.

Security

Once Docker is downloaded, the container is pulled, and autogen.py is downloaded, this script requires no additional contact with any external resources to run. All data is stored locally in a series of folders. Once the initial scan is processed into an .nrrd format it no longer contains any patient identifiers unless there was an identifier in the file name which was originally created to store the Dicom data. For complete security, disconnecting the computer being used from all external sources will ensure nearly complete security.

Last update: 1/30/2020

Additional Notes

This script will attempt to segment the lung into right and left parts. This process has an 8% failure rate, so not all data will have these two measurements available. In order to run the analysis, the system will also require at least 1G per scan. If this is not possible, run the script in batches and save the results, deleting old files when necessary.

A more comprehensive manual can be found here:
https://hub.docker.com/r/acilbwh/chestimagingplatform

For help running the script, or additional questions contact Nathan Wies at nwies@u.rochester.edu.