README for using how to use CAD software with MC-GPU mammogram data.

I used Docker two different docker containers to run MC-GPU and the CAD software since they had different dependencies with Python (the model for the CAD software uses Python 2).

Relevant GitHub/Docker Resources: (will add explanations)

* <https://github.com/riblidezso/frcnn_cad>
* <https://github.com/rbgirshick/py-faster-rcnn/blob/master/lib/fast_rcnn/test.py>
* <https://gist.github.com/cewee/356b941a4006a502a67f68213f1a76b5>

Glossary (will add explanations):

* Docker image
* Docker container

Source: <https://github.com/rbgirshick/py-faster-rcnn/issues/509?_pjax=%23js-repo-pjax-container>

1. Install Docker on your computer.
2. Make sure your computer has the right dependices for running NVIDIA GPU with a driver
3. Run nvidia-smi to test whether the GPU works (will expand on more)
4. Create a file named ‘Dockerfile’ with no file extension with the information in <https://gist.github.com/cewee/356b941a4006a502a67f68213f1a76b5> inside. The file should be created in its own folder.
   1. The Dockerfile creates the Docker image that we will use to run our CAD software.
   2. Edit the first line of the Dockerfile so that it works with the systems of your GPU and NVIDIA driver. (I’m not sure if changing these from the original Dockerfile will mess up its dependencies with the actual CAD software).
5. Create the Docker image by running the command below. I named the Docker image *py-faster-rcnn.*
   1. **docker build - -tag <NAME OF DOCKER IMAGE> .**
6. Run the Docker image you just created as a Docker container.
   1. **docker run -it --gpus all -v /home:/home -p 8008:8008 --name <INSERT NAME OF CONTAINER> <NAME OF DOCKER IMAGE>:latest /bin/bash**
   2. - - gpus is where we can specify the number of GPUS, and allow us to use GPUs inside the Docker container.
   3. -v (will expand on more) allow us to access the data/images we want to pass through the CAD software. Now, you will be able to access the /home folder and all of the files inside while you are inside the Docker container.
   4. The port 8008 is what we will use to communicate the docker container and our host computer.
      1. From <https://docs.docker.com/get-started/part2/>: --publish (or - -p) asks Docker to forward traffic incoming on the host’s port 8000 to the container’s port 8080. Containers have their own private set of ports, so if you want to reach one from the network, you have to forward traffic to it in this way. Otherwise, firewall rules will prevent all network traffic from reaching your container, as a default security posture.
7. Once the Docker container is created, enter the container by running the command below.
   1. **docker exec -it <CONTAINER ID> /bin/bash**
   2. The CONTAINER ID can be found by running **docker ps** and finding the CONTAINER ID from the image with the Docker image name you choose.
8. Once inside the container, you should be the root user and inside the folder /opt/py-faster-rcnn. You need to download the models by downloading the Faster-RCNN models.
   1. **cd data**
   2. **wget** [**https://dl.dropboxusercontent.com/s/o6ii098bu51d139/faster\_rcnn\_models.tgz**](https://dl.dropboxusercontent.com/s/o6ii098bu51d139/faster_rcnn_models.tgz)
   3. **tar zxvf faster\_rcnn\_models.tgz**
9. Navigate to the /opt folder and download the GitHub repo with the demo Jupyter notebook.
   1. **git clone** [**https://github.com/riblidezso/frcnn\_cad.git**](https://github.com/riblidezso/frcnn_cad.git)
10. Download the weights for the model into the /opt/frcnn\_cad folder.
    1. Weights can be downloaded through this link: <http://dkrib.web.elte.hu/cad_faster_rcnn/vgg16_frcnn_cad.caffemodel>
11. Install Jupyter notebook.
    1. **pip install notebook**
12. Run Jupyter notebook and access the demo notebook in your localhost computer’s browser by running the command below.
    1. **jupyter notebook --ip 0.0.0.0 --port 8008 --no-browser --allow-root (will expand on more)**
    2. For the host, make sure to use the port that was indicated when creating the container.
13. The output after running the command should have links that allow you to access the Jupyter notebook file system on your computer’s browser.
    1. Another way: On your local host computer, go the url localhost:8008 (or whichever port used when creating the container). You should be prompted to enter a password or token ID, and that can be found in the output in the terminal after running the command in #12.
14. Now you can access the demo notebook which contains the CAD software. Run the cells in order, and the last cell ‘Load and analyze image’ allows you to insert your own image to detect malignant masses or microcalcification clusters by changing the file path to the image in the method **cv2.imread(<INSERT PATH OF IMAGE>).** Usually, the file path to the image is through the shared folder indicated when the Docker container was first created.