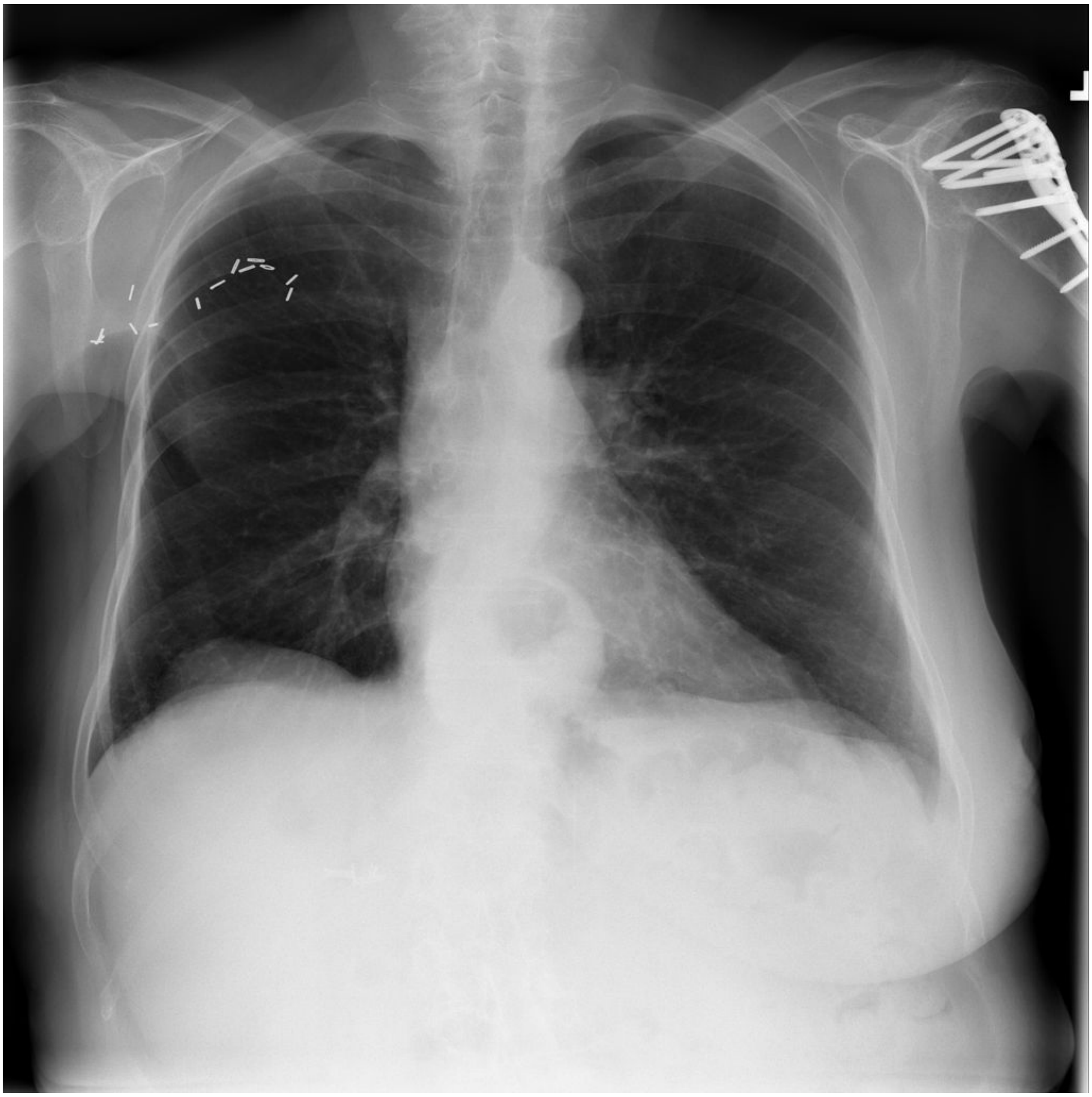
Image analysis of lung diseases



By: Sander, Stephan, Joshua & Michelle

Date: 04/10/2020

The Classifier

The model created by this project is a convolutional neural network (CNN), which is common practice when analyzing visual data. The first four scripts are part of preprocessing steps whilst the last does the training of the actual model. The first script, filter_diseases.py, filters the data by selecting the data belonging to three diseases. We chose for the diseases effusion, cardiomegaly and pneumothorax due to the fact that effusion and pneumothorax look alike on the scans, and cardiomegaly for being more divergent. The second script, resize_images, crops and resizes images with the goal of discarding irrelevant content. The third script, reconcile_labels.py, utilizes the csv file to “tag” the resized images with their corresponding diseases. The fourth script, image_to_array.py, very straightforwardly creates a numpy array from these images. The last script splits the data in testing and training sets, creates a model using the training set and tests the model on the test set. A table of these different modules and their description can be found below in **Table 1**.

Table 1: *Overview of scripts used in this project.*

Step	**Description**	**Script**
1	Split data for three diseases	filter_diseases.py
2	Resize Images	resize_images.py
3	Reconcile Labels	reconcile_labels.py
4	Convert Images to Arrays	image_to_array.py
5	CNN Model	cnn_keras.py

Cuda

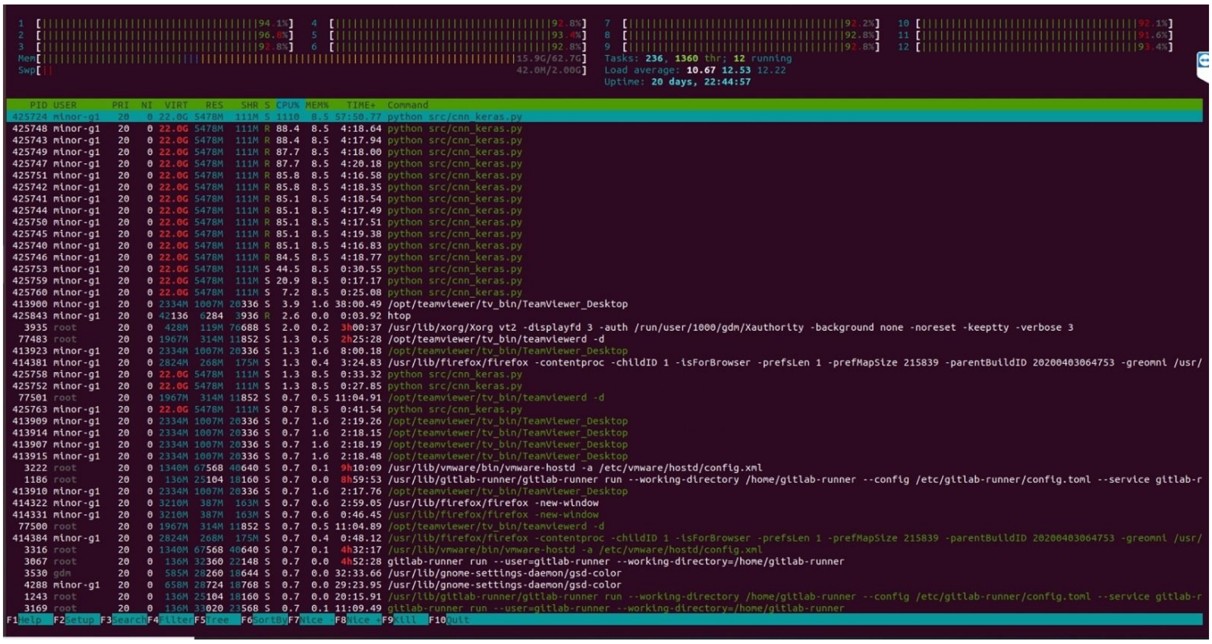
A comparison was made between the performance of the model training when using a CPU and a GPU (**Table 2**). The runtime of the CNN training model without using GPU was just over 8 hours. To find out how much faster the prediction can be by using GPU, the GPU of Minor-g1 was used. With a runtime of 32 minutes, the speed of the prediction model was improved by approximately 15 times.

Table 2: *Comparison of the runtime, accuracy, precision, recall and F1 score of the CNN model between using CPU and GPU.*

	Runtime (hh:mm:ss)	**Accuracy**	**Precision**	**Recall**	**F1 score**
CPU	08:01:49	0.4008	0.16	0.40	0.23
GPU	00:33:23	0.4000	0.16	0.41	0.23

The results from the test dataset are viewed as validation of the model, considering testing on another validation set would result in very similar results. Also this helps keep the training set of substantial size for proper training of the model.

Model Optimization

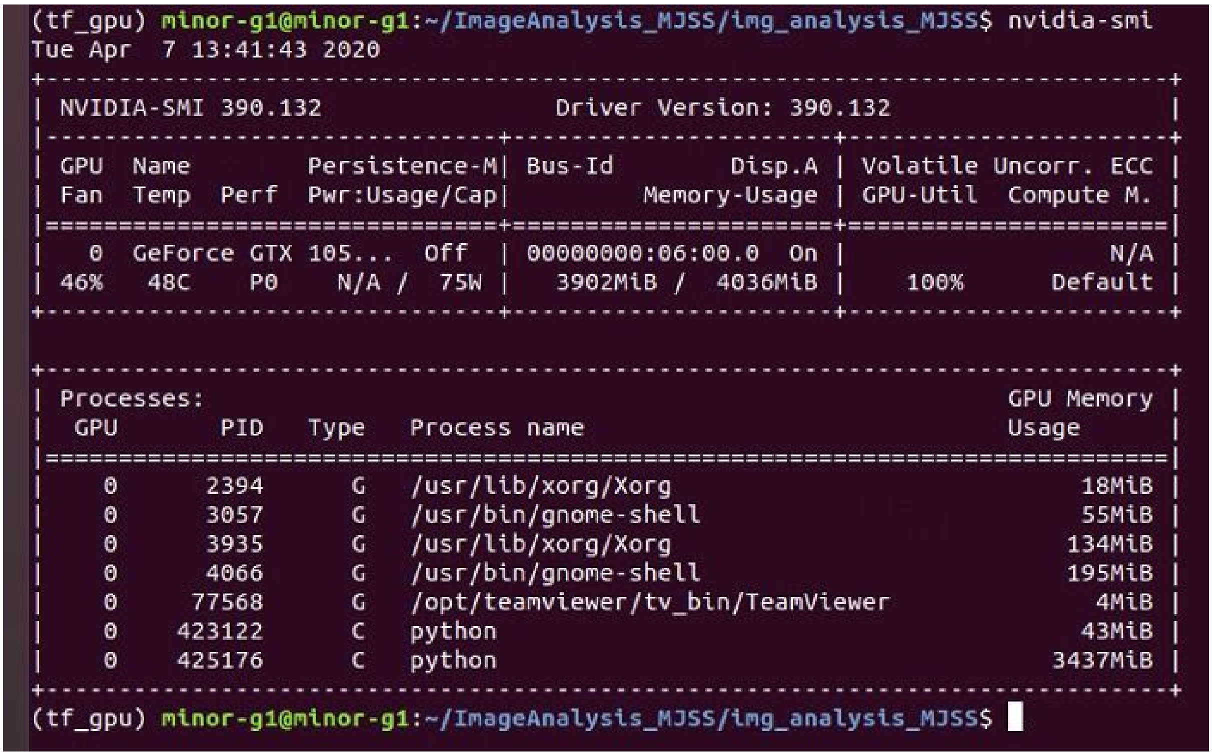


```
  1  [||||||||||||||||||||||||||||||||||||||||94.1%]   4  [||||||||||||||||||||||||||||||||||||||||92.8%]   7  [||||||||||||||||||||||||||||||||||||||||92.2%]   10 [||||||||||||||||||||||||||||||||||||||||92.1%]
  2  [||||||||||||||||||||||||||||||||||||||||96.8%]   5  [||||||||||||||||||||||||||||||||||||||||93.4%]   8  [||||||||||||||||||||||||||||||||||||||||92.8%]   11 [||||||||||||||||||||||||||||||||||||||||91.6%]
  3  [||||||||||||||||||||||||||||||||||||||||92.8%]   6  [||||||||||||||||||||||||||||||||||||||||92.8%]   9  [||||||||||||||||||||||||||||||||||||||||92.8%]   12 [||||||||||||||||||||||||||||||||||||||||93.4%]
  Mem[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||15.9G/62.7G]   Tasks: 236, 1360 thr; 12 running
  Swp[||                                                                                   42.0M/2.00G]   Load average: 10.67 12.53 12.22
                                                                                                          Uptime: 20 days, 22:44:57

    PID USER       PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 425724 minor-g1    20   0 22.0G 5478M  111M S 1110  8.5 57:50.77 python src/cnn_keras.py
 425748 minor-g1    20   0 22.0G 5478M  111M R 88.4  8.5  4:18.64 python src/cnn_keras.py
 425743 minor-g1    20   0 22.0G 5478M  111M R 88.4  8.5  4:17.94 python src/cnn_keras.py
 425749 minor-g1    20   0 22.0G 5478M  111M R 87.7  8.5  4:18.00 python src/cnn_keras.py
 425747 minor-g1    20   0 22.0G 5478M  111M R 87.7  8.5  4:20.18 python src/cnn_keras.py
 425751 minor-g1    20   0 22.0G 5478M  111M R 85.8  8.5  4:16.58 python src/cnn_keras.py
 425742 minor-g1    20   0 22.0G 5478M  111M R 85.8  8.5  4:18.35 python src/cnn_keras.py
 425741 minor-g1    20   0 22.0G 5478M  111M R 85.1  8.5  4:18.54 python src/cnn_keras.py
 425744 minor-g1    20   0 22.0G 5478M  111M R 85.1  8.5  4:17.49 python src/cnn_keras.py
 425750 minor-g1    20   0 22.0G 5478M  111M R 85.1  8.5  4:17.51 python src/cnn_keras.py
 425745 minor-g1    20   0 22.0G 5478M  111M R 85.1  8.5  4:19.38 python src/cnn_keras.py
 425740 minor-g1    20   0 22.0G 5478M  111M R 85.1  8.5  4:16.83 python src/cnn_keras.py
 425746 minor-g1    20   0 22.0G 5478M  111M R 84.5  8.5  4:18.77 python src/cnn_keras.py
 425753 minor-g1    20   0 22.0G 5478M  111M S 44.5  8.5  0:30.55 python src/cnn_keras.py
 425759 minor-g1    20   0 22.0G 5478M  111M S 20.9  8.5  0:17.17 python src/cnn_keras.py
 425760 minor-g1    20   0 22.0G 5478M  111M S  7.2  8.5  0:25.08 python src/cnn_keras.py
 413900 minor-g1    20   0 2334M 1007M 20336 S  3.9  1.6 38:00.49 /opt/teamviewer/tv_bin/TeamViewer_Desktop
 425843 minor-g1    20   0 42136  6284  3936 R  2.6  0.0  0:03.92 htop
   3935 root        20   0  428M  119M 76688 S  2.0  0.2 3h00:37 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
  77483 root        20   0 1967M  314M 11852 S  1.3  0.5 2h25:28 /opt/teamviewer/tv_bin/teamviewerd -d
 413923 minor-g1    20   0 2334M 1007M 20336 S  1.3  1.6  8:00.18 /opt/teamviewer/tv_bin/TeamViewer_Desktop
 414381 minor-g1    20   0 2824M  268M  175M S  1.3  0.4  3:24.83 /usr/lib/firefox/firefox -contentproc -childID 1 -isForBrowser -prefsLen 1 -prefMapSize 215839 -parentBuildID 20200403064753 -greomni /usr/
 425758 minor-g1    20   0 22.0G 5478M  111M S  1.3  8.5  0:33.32 python src/cnn_keras.py
 425752 minor-g1    20   0 22.0G 5478M  111M S  1.3  8.5  0:27.85 python src/cnn_keras.py
  77501 root        20   0 1967M  314M 11852 S  0.7  0.5 11:04.91 /opt/teamviewer/tv_bin/teamviewerd -d
 425763 minor-g1    20   0 22.0G 5478M  111M S  0.7  8.5  0:41.54 python src/cnn_keras.py
 413909 minor-g1    20   0 2334M 1007M 20336 S  0.7  1.6  2:19.26 /opt/teamviewer/tv_bin/TeamViewer_Desktop
 413914 minor-g1    20   0 2334M 1007M 20336 S  0.7  1.6  2:18.15 /opt/teamviewer/tv_bin/TeamViewer_Desktop
 413907 minor-g1    20   0 2334M 1007M 20336 S  0.7  1.6  2:18.19 /opt/teamviewer/tv_bin/TeamViewer_Desktop
 413915 minor-g1    20   0 2334M 1007M 20336 S  0.7  1.6  2:18.48 /opt/teamviewer/tv_bin/TeamViewer_Desktop
   3222 root        20   0 1340M 67568 40640 S  0.7  0.1 9h10:09 /usr/lib/vmware/bin/vmware-hostd -a /etc/vmware/hostd/config.xml
   1186 root        20   0  136M 25104 18160 S  0.7  0.0 8h59:53 /usr/lib/gitlab-runner/gitlab-runner run --working-directory /home/gitlab-runner --config /etc/gitlab-runner/config.toml --service gitlab-r
 413910 minor-g1    20   0 2334M 1007M 20336 S  0.7  1.6  2:17.76 /opt/teamviewer/tv_bin/TeamViewer_Desktop
 414322 minor-g1    20   0 3210M  387M  163M S  0.7  0.6  2:59.05 /usr/lib/firefox/firefox -new-window
 414331 minor-g1    20   0 3210M  387M  163M S  0.7  0.6  0:46.45 /usr/lib/firefox/firefox -new-window
  77500 root        20   0 1967M  314M 11852 S  0.7  0.5 11:04.89 /opt/teamviewer/tv_bin/teamviewerd -d
 414384 minor-g1    20   0 2824M  268M  175M S  0.7  0.4  0:48.12 /usr/lib/firefox/firefox -contentproc -childID 1 -isForBrowser -prefsLen 1 -prefMapSize 215839 -parentBuildID 20200403064753 -greomni /usr/
   3316 root        20   0 1340M 67568 40640 S  0.7  0.1 4h32:17 /usr/lib/vmware/bin/vmware-hostd -a /etc/vmware/hostd/config.xml
   3067 root        20   0  136M 32360 22148 S  0.7  0.0 4h52:28 gitlab-runner run --user=gitlab-runner --working-directory=/home/gitlab-runner
   3530 gdm         20   0  585M 28260 18644 S  0.7  0.0 32:33.66 /usr/lib/gnome-settings-daemon/gsd-color
   4288 minor-g1    20   0  658M 28724 18768 S  0.7  0.0 29:23.95 /usr/lib/gnome-settings-daemon/gsd-color
   1243 root        20   0  136M 25104 18160 S  0.7  0.0 20:15.91 /usr/lib/gitlab-runner/gitlab-runner run --working-directory /home/gitlab-runner --config /etc/gitlab-runner/config.toml --service gitlab-r
   3169 root        20   0  136M 33020 23568 S  0.7  0.1 11:09.49 gitlab-runner run --user=gitlab-runner --working-directory=/home/gitlab-runner
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Figure 1: *Overview of CPU usage with htop command.*

Figure 1 shows that the script utilizes around 1100% of the CPU when training and validating the model with tensorflow cpu. Individual cpu cores show an average usage of 91%.



```
(tf_gpu) minor-g1@minor-g1:~/ImageAnalysis_MJSS/img_analysis_MJSS$ nvidia-smi
Tue Apr  7 13:41:43 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 390.132                Driver Version: 390.132                   |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX 105...  Off  | 00000000:06:00.0  On |                  N/A |
| 46%   48C    P0    N/A /  75W |   3902MiB /  4036MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0      2394      G   /usr/lib/xorg/Xorg                            18MiB |
|    0      3057      G   /usr/bin/gnome-shell                          55MiB |
|    0      3935      G   /usr/lib/xorg/Xorg                           134MiB |
|    0      4066      G   /usr/bin/gnome-shell                         195MiB |
|    0     77568      G   /opt/teamviewer/tv_bin/TeamViewer              4MiB |
|    0    423122      C   python                                        43MiB |
|    0    425176      C   python                                      3437MiB |
+-----------------------------------------------------------------------------+
(tf_gpu) minor-g1@minor-g1:~/ImageAnalysis_MJSS/img_analysis_MJSS$ █
```

Figure 2: *Overview of GPU usage with nvidia-smi command.*

Figure 2 shows the script using 3437MiB of the available 4036MiB memory from the GPU. Which corresponds to a GPU memory utilization of 85.16%.

Discussion

We noticed that the model training using a CPU took 15 times longer than when using the GPU. This is likely due to GPU's being better at floating point and matrix calculations. While the use of one GPU already increased training performance, using multiple GPU's can certainly be more beneficial. Given the available memory of the single GPU used in this project a batch size of 32 was used. By increasing the number of GPU's, the available GPU memory can be increased significantly and consequently a greater batch size can be used.

A dataset containing the images of three diseases were used (effusion, cardiomegaly and pneumothorax) resulting in scores described in **Table 2**. The scores resulting from the CPU and GPU trained models are very similar. The GPU did perform better when it comes to the running time (8h v.s. 0.5h). However, the difference in recall score (0.40 v.s. 0.41), that is the proportion of actual positives that are identified correctly, is minimal and may be a consequence of chance. The use of a smaller dataset may contribute to overall lower scores. In addition, the fact that at least 10% of the images are incorrectly annotated further reduces these final scores. To improve on the model accuracy the incorrectly annotated images could be identified and the provided bounding boxes could be used.

Data Availability

The code used in this project can be found at https://github.com/MichelledeGroot/img_analysis. Furthermore, the chest x-ray images used to train and validate our model were made publicly available by the National Institutes of Health (NIH) on their website; https://nihcc.app.box.com/v/ChestXray-NIHCC.

Acknowledgements

Parts of Greg Chase’s code (https://github.com/gregwchase/nih-chest-xray) were used in this project.