COVID-CT Starlight Saviors Team 3: Dataset Expansion

Matthew Merritt, Michael Merritt, Alexandra DeLeaver,
DaShawn Simon, Isaac Uy, and Manasa Lingireddy

July 26, 2020

Contents

Abstract

For this project, the Starlights at MISI were tasked with improving a machine learning algorithm that would be able to use CT scans of lungs to determine if a patient had COVID or not. This paper will focus specifically on the steps and work of Team 3. Team 3 was given three major tasks for this project: to increase the size of the dataset by adding other COVID CT datasets, to make a script for partitioning the train/test/val datasets, and to make an API for data loading. The work done by Team 3 can be divided into four distinct sections: the initial setup of the Linux workspaces, GitHub branch on the forked repository, and original dataset, the implementation of the larger dataset as well as the partitioning of it, the GPU setup and collaboration with Team 2 for GitHub support and Team 4 for the Python training script, and the finalization of the scripts used as the API for implementing other datasets. By the end of the Starlights' internship at MISI, the expanded dataset was implemented, partitioned, and tested, giving results comparable to the original project's dataset. The partitioning script and file sorting script were also included to serve as the API for implementing other datasets. All of this can be found in the Starlights' GitHub repository for the project at https://github.com/walkerjbuckle/COVID-CT-Starlight-Saviors.

1 Introduction

As of July 23, 2020, there are more than 4 million cases of COVID-19 in just the US alone. To help provide alternative methods of testing, a group of scientists from UCSD made a machine learning model that could use CT scans of lungs to test for COVID-19 (Zhao, Zhang, He, & Xie, 2020). As Starlights in MISI's Summer Virtual Internship, we split up into six teams to build upon the original repository and model. The main goal of Team 3's work was to increase the accuracy of the original machine learning algorithm by expanding the dataset, therefore giving the algorithm more examples of COVID CT scans and Non-COVID CT scans to train from. To accomplish this, Team 3 focused on completing three tasks: increasing the size of the dataset, dividing the dataset into three subsets (one for training the model, one for evaluating the model trained, and one for testing the model), and making an API to allow for the implementation of other datasets as well. Although we did not have much experience with using tools such as GitHub and Linux systems, we were able to collaborate with representatives from other teams and their mentor Steven Griffin to overcome any obstacles that we ran into during the project.

2 Initial Setup

As we were unfamiliar with many of the tools we would end up using throughout this project, there was a lot of setup for us to do with the assistance of our project leader and mentor Steven Griffin. The first step was to set up a fork of the scientists' original GitHub repository and to then make a branch for our code to not conflict with other groups and their results. We also had to set up virtual machines using Amazon WorkSpaces and then configure these workspaces to have all of the necessary additions, such as Conda for Python package management and Jupyter to run the models for testing. After we set all of that up, we decided to run a preliminary test using the original dataset and model, and received the following results:

```
TP= 83 TN= 90 FN= 22 FP= 8
TP+FP 91
precision 0.9120879120879121
recall 0.7904761904761904
F1 0.8469387755102041
acc 0.8522167487684729
AUC 0.9402332361516035
```

According to these results, the original model was achieving around 0.85 accuracy. As such, with our expanded dataset, we would try to produce results close to that.

3 Dataset Expansion

The next step in improving the model was to find a larger dataset to use when training the model. After some research and consulting Steven, we decided that the best dataset for our project was the SARS-COV-2 Ct-Scan Dataset (https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset), as it was considerably larger than the dataset included in the original repository (Soares, Angelov, Biaso, Higa Froes, & Kanda Abe, 2020). Using this dataset, we added the images to the Images-processed folder for use in the training, evaluation, and testing of the model. We then wrote a Python script to partition the images into three categories: train, val, and test. We used the script to maintain the original ratio of images in each category while adding all of the new images for use as well. This Python script was not included our GitHub repository, as it was only programmed to work with this dataset on a specific local environment, and was eventually going to be replaced with a more versatile and flexible script instead (this more flexible version is partitioning.py, which is included in the repository). After splitting the images into the three categories, we re-ran the training script to see if there was a significant difference in the results. These are the results we obtained from the initial test with the expanded dataset:

```
TP= 105 TN= 434 FN= 334 FP=15
TP+FP 120
precision 0.875
recall 0.23917995444191345
F1 0.3756708407871198
acc 0.60250569476082
```

AUC 0.6592846653971285

Clearly, these results were less than ideal, as the accuracy was significantly lower than the initial test and were far below acceptable results. There was also an error saying that CUDA was not enabled, so we assumed that the low accuracy and poor results were due to the lack of graphical processing capabilities.

4 Troubleshooting Accuracy and Storage Issues

As mentioned previously, we believed that the sudden drop in accuracy was due to the lack of graphical processing capacity on the virtual machine we tested it on. To test this hypothesis and fix the issue, we used SSH to connect to a computer with a more powerful GPU. Although setting up SSH was relatively easy, we ran into more issues when trying to upload the larger dataset to our GitHub branch, as the dataset was now about 160 MB of COVID images and 160 MB of Non-COVID images. This exceeded the maximum file size that GitHub allows to be pushed, so we had to split the image zip files in half and push four zip files instead of just the original two. As the process of organizing the four zip files into two folders was slightly less intuitive, we included a Python script, filesort.py, that would unzip the four zip files and sort them into the proper folders for the model to read. This allowed us to push our branch to the GPU instance, but we quickly ran into another issue. We could not manage to get Jupyter Notebook working on the GPU instance, as we only had access to the terminal. To circumvent this issue, we were able to use a Python copy of the Jupyter Notebook file to run on the GPU instance from Team 4. With the new script from Team 4, we merged out branch to master in the GitHub repository so other groups could use the dataset in testing.

5 Improving Script for Dataset Integration

Before performing the final tests on the expanded dataset, we decided to make a more formal and versatile version of partitioning.py, the script used for adding datasets to the model for training. The new script now looks for folders named CT_COVID and CT_NonCOVID in the Images-processed directory (if you ran filesort.py, these should be the folders generated by that script). However, the new partitioning.py also allows for other datasets to be added if the filepaths for the folders of COVID and non-COVID images are put in as command line arguments. For example, when we added a dataset from Downloads with the folders "COVID" and "non-COVID", we ran "python partitioning.py --c '../../../../Downloads/COVID' --nc '../../../../Downloads/non-COVID'" from the terminal, since partitioning.py was located in Documents/demo/COVID-CT-Starlight-Saviors/Data-split. The new script also maintains the original ratios of images in the train, val, and test categories no matter the size of the dataset added, whereas the previous script only worked for the one dataset we decided to implement.

6 Results

Using the Python script provided by Team 4 on the GPU instance with the expanded dataset, we ran three tests and received the following results:

```
Test 1:
TP= 427 TN= 382 FN= 12 FP= 57
TP+FP 484
precision 0.8822314049586777
recall 0.9726651480637813
F1 0.9252437703141928
acc 0.9214123006833713
AUC 0.9799658573793204
```

```
Test 2:
TP= 390 TN= 415 FN= 49 FP= 24
TP+FP 414
precision 0.9420289855072463
recall 0.8883826879271071
F1 0.9144196951934349
acc 0.9168564920273349
AUC 0.9785856237773777
```

```
Test 3:
TP= 395 TN= 410 FN= 44 FP= 29
TP+FP 424
precision 0.9316037735849056
recall 0.8997722095671982
F1 0.9154113557358055
acc 0.9168564920273349
AUC 0.9794365948703048
```

This gave us an average accuracy of about 0.92, an F1 of about 0.92, and an area under curve of 0.98. Compared to the original scientists' model, our model trained on the enlarged dataset shows an increase in accuracy of 0.03, an increase in F1 of 0.02, and the same area under curve (AUC) (Zhao et al., 2020). Compared to the model we trained using the original scientists' program, our model trained on the enlarged dataset shows an increase in accuracy of 0.07, an increase in F1 of 0.07, and an increase in AUC of 0.04.

7 Conclusion

From our three tests mentioned above, we can see that enlarging the dataset used to train the model proved to be beneficial toward the model's accuracy. As COVID-19 is very expensive to treat properly, any increase in the accuracy of testing is beneficial, as the increased time spent training the model on more images is outweighed by the potential time saved on future testing of potential false positives from the model. While the improvement of the model from the expanded dataset was minimal, it was also not very surprising, as the original scientists' model was already accurate enough to see clinical use, and as such there was not much room from improvement.

References

Soares, E., Angelov, P., Biaso, S., Higa Froes, M., & Kanda Abe, D. (2020). Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification. *medRxiv*. Retrieved from `https://www.medrxiv.org/content/early/2020/05/14/2020.04.24.20078584` doi: 10.1101/2020.04.24.20078584

Zhao, J., Zhang, Y., He, X., & Xie, P. (2020). Covid-ct-dataset: a ct scan dataset about covid-19. *arXiv preprint arXiv:2003.13865*.