Cross-Platform Application to Diagnose COVID-19 From Chest CT Scans

Ballhaus, Brooke
brookeballhaus@gmail.com

El Kochta, Ryan
relkochta@protonmail.com

Kathuria, Daman
damankathuria7@gmail.com

22 July 2020

Abstract

In the final months of 2019, an outbreak of a novel coronavirus strain known as SARS-CoV-2, causing a disease known as COVID-19, began in Wuhan, Hubei, China, and soon spread to surrounding regions. Within months, it had reached Europe, the Americas, Oceania, and Africa, and the rest of Asia. Governments around the globe have tried to control the pandemic, and part of that involves ramping up testing capacity. However, traditional nose-swab tests have proven difficult to mass-produce, distribute, and properly administer. As a possible alternative, many scientists have trained experimental neural networks to predict COVID-19 based on a CT scan of a patient's lungs. Some have had promising results — however, they are not particularly useful without an intuitive way for a doctor, who may not know how to load and evaluate a neural network manually, to diagnose patients.

MISI's 2020 summer interns were divided into six teams. As part of the first team, our task was to develop a set of tools to allow a doctor to easily evaluate a CT scan against a pre-trained model, helping them be informed

in diagnosing a patient. Our tools consist of two portable Python scripts, with binaries provided for Windows, macOS, and Linux. The first script is a command line tool that takes an image (a patient's CT scan) as input and outputs the model's prediction as for whether the patient is COVID-19 positive or negative. The second adds a PyQt5-based GUI, allowing evaluation to be point-and-click.

We started by cloning the dataset and models compiled by He et al. at UCSD [He et al., 2020]. Next, we prepared our development environment. Any Python 3 environment should work for development of these tools, however we used a combination of Fedora Linux 32 and Amazon Linux 2 with no issues. You will need to use a Python package manager to install the following dependencies: PyQt5 (for GUI application), PyTorch, Torchvision, and Pillow, in addition to the standard Python 3 library. We had success with both Pip and Anaconda. The scripts were developed in Vim, however we recommend a proper Python IDE such as PyCharm for new users.

The command-line script starts by initializing the transformer constants provided in He et al.'s code. It next loads the image provided as the first argument and the model into memory. Next, it evaluates the image using the model on either the CPU or the GPU if CUDA is available. Finally, it outputs a prediction.

The functionality described above is implemented in its near entirety in the GUI application's *eval()* function. A *Predict(QWidget)* class is loaded and the application is shown. A screen appears prompting the doctor to load a CT scan image from a file chooser. The *eval()* function is then called on the selected image, and an output is printed to the window. We recognize that there is the possibility for some threading here, however there is little reason to interact with the app while the model is evaluating in the first place.

For end users, binaries generated using PyInstaller for the three major desktop operating systems are provided. There is one outstanding issue — the binary packages are quite large, as they include the entire Python stack. There is doubtlessly some size optimization that can be done here, however we were unable to trim them down any further.

We believe that the tools we have developed are very useful in fighting the COVID-19 pandemic. Once production-ready models are available, a doctor can simply take a CT scan of a patient, load it into the application, and make a diagnosis, all in a fraction of the time it is currently taking for nose-swab test results. We hope that research on the models themselves continues or accelerates, as this is a very time-sensitive issue.

References

[He et al., 2020] He, X., Yang, X., Zhang, S., Zhao, J., Zhang, Y., Xing, E., and Xie, P. (2020). Sample-efficient deep learning for COVID-19 diagnosis based on CT scans. *medrxiv*.