Matthew Backlund

CS 5600 Project 1

I started this project by getting an environment set up for the project to run in. I set up a python venv called *beeenv* and imported the necessary packages. However, I quickly ran into issues with the dependencies being incompatible with each other. After about 2 hours looking at forum posts and asking ChatGPT (which was absolutely no help) I realized that the version of CUDA I had installed on my computer was incompatible with pytorch. After this realization I uninstalled CUDA and removed all packages from *beeenv*, then redownloaded CUDA 12.4 and installed versions of pytorch, torchvision, and torchaudio that were compatible with this CUDA version.

After getting CUDA set up correctly I tried to install the remaining packages my project needed, specifically onnx, but couldn’t download anything. Somehow my venv had become corrupted and had lost the ability to download pip packages. After retracing my steps I realized that I had changed the name of one of the parent folders which caused *beeenv* to try to download to a path that didn’t exist anymore. After deleting and recreating *beeenv* I was finally able to get all the python packages I needed installed.

My next challenge arose when the YOLO would still crash almost immediately when it started running with the following error:

RuntimeError:

An attempt has been made to start a new process before the current process has finished its bootstrapping phase.

After a bit of research I realized that the starter code we’d been given needed to be inside of a function and that the program should have an `if \_\_name\_\_ == “\_\_main\_\_”’ block. My understanding of this is that multiprocessing is implemented differently on windows and uses ‘spawn’ instead of ‘fork’ (The Linux Foundation, n.d.). Because of this our code needs to be wrapped in an if-clause to protect it from executing multiple times.

Once the code was wrapped in an if-clause I was finally able to get the code to run, only to be sorely disappointed. The model was able identify Beehive frames with about 95% accuracy, but that was it. All the cells were labeled as background, and all the background was labeled as cells. I bumped the number of epochs up to 300 and got nearly identical results, with only the average certainty for beehive frames increasing slightly.

After running a few more test runs my model wasn’t getting any better. I was still using the default YOLO model (yolov8n.pt), so I decided to use the model I did 300 epochs on as the new baseline model. I trained about 300 more epochs on this model. This gave me immediate positive results. Now the model was able to correctly identify about 6% of pollen cells and about 2% of capped worker brood cells! This does not sound like much, but it was a huge improvement from where I had been previously. I moved this new model to be the new starting model and did the same thing again. However, this time the model stopped training after 101 runs with the following output:

EarlyStopping: Training stopped early as no improvement observed in last 100 epochs. Best results observed at epoch 1, best model saved as best.pt.

The model had stopped training because it was not getting any better results after 100 epochs. My model had gotten as good as it could get with only increasing the number of epochs it was running.

I did some research on the earlier method I had been using, where I used a pre-trained DNN as a starting point for another. From my reading, the strategy is called transfer learning. It is often used to take knowledge a model has learned from one dataset and leverage that knowledge on another dataset. Since we are using the same dataset each time we train, there is not a huge advantage to using this strategy unless I begin doing hyperparameter tuning to change the way the model trains each run.

At this point my only thought was to start hyperparameter tuning. I did some research on hyperparameters for YOLO and settled on epochs, image size, learning rate, momentum, and weight decay as the starting hyperparameters to tune.

# References

The Linux Foundation. (n.d.). *PyTorch*. Retrieved from https://pytorch.org/docs/stable/notes/windows.html#multiprocessing-error-without-if-clause-protection