Hi Dr. Martin,

Thank you for the update, the incremental change makes sense, and just out of curiosity, are there any trade-offs with this approach? Like limitations in per-slice analysis I just want to make sure I understand what we gain vs what we might give up with this change. Also, regarding our meeting on Thursday just confirming are we still on for 10:30 am?

Best regards,

████

████

I've had to make a small but important change to your prediction script to fix the memory issue you encountered when processing more slices.

The original code stored every pixel's true and predicted label across all slices, which quickly became too large for memory. When I tried it on the large set it ran to slice 400 then returned:

--- Overall Classification Metrics for Predicted Slices (Slices 10-40) --- Traceback (most recent call last): File "./Prediction.py", line 235, in <module> actual_report_labels = np.sort(np.unique(all_true_labels)) ^^^^^^^^^^^^^^^^^^^^^^^^^ File "./Lib/site-packages/numpy/lib/_arraysetops_impl.py", line 284, in unique ar = np.asanyarray(ar) ^^^^^^^^^^^^^^^^ numpy._core._exceptions._ArrayMemoryError: Unable to allocate 6.54 GiB for an array with shape (877500000,) and data type int64

So I inserted an incremental confusion matrix that updates with each slice. This approach gives you the correct overall confusion matrix and performance metrics (accuracy, precision, recall, F1) without storing unnecessary data. Changes in the code are noted and the original work remains, but is commented out.

The rest of your code remains the same. I've pushed the updated code for large sets to the repository (Prediction_V2) for your review. Runtimes are about 10-15 minutes.

Results are copied below

Best regards,

Dr. Martin

Eric J Martin, *Ph.D., M.Sc.(Eng.), B.Sc.(Eng.),B.A., P.Eng*.

Lecturer of Civil Environmental Engineering

Programme Leader: Water, Waste and Environmental Engineering

Faculty of Engineering & Science

School of Engineering

University of Greenwich