Presentation 2

- Links to presentation(s) and code(s) on GitHub
    - Presentation
    - Code
- What did you do?
    - I fixed the train-test split mechanism so that it no longer mixes data from the same case across splits. Since the labels are assigned at the case level, this change prevents data leakage that could affect model performance. I also redesigned the data loader to follow the hierarchical structure required for our new pipeline (from patch to slice to stain to case). In addition, I improved the code formatting and overall efficiency.
- How does it help the project?
    - This establishes a foundation for the new pipeline, which will allow us to define and train the actual model later with properly structured and reliable data.
- Issues faced (if any)
    - Google Cloud was very inefficient in handling file operations. Reading the directories repeatedly often failed, and generating the data loader took an extremely long time because of the repeated reading process.
- Attempts to resolve issues (if any)
    - I implemented directory caching so that the code no longer needs to reread directories every time. However, reading the image files themselves still takes a long time, although this may improve once we migrate to Quest.
- Issues resolved (if any)
    - The data splitting issue has been fully resolved, and caching now works properly. I also built multiple tests to confirm that there is no overlap between the train and test sets.
- Next steps
    - The next step is to migrate all code and data to Quest, train the model there, and evaluate its performance.
- References (Mention if you built up on someone else's work)
    - ChatGPT (coding)