HHMI Interview Exercises

The overall goal is to access the following publicly available datasets and metadata. These are all three-dimensional electron microscopy acquisitions of different specimen:

1.https://idr.openmicroscopy.org/webclient/img_detail/9846137/?dataset=10740
2. https://www.ebi.ac.uk/empiar/EMPIAR-11759/
3. https://www.epfl.ch/labs/cvlab/data/data-em/
4. https://openorganelle.janelia.org/datasets/jrc_mus-nacc-2
5. https://tinyurl.com/hemibrain-ng (download only a random 1000x1000x1000 pixel crop region)

Tasks:

1. Develop code for downloading these image datasets that are stored in different ways (ideally parallel, multi-threaded) and put the code on GitHub (ideally in a way that we can reproduce it).

   A: The code was uploaded to https://github.com/yjinbhhs.

   For **dataset 1**, it is a multi-slice .tif image, but the website only allows to download one slice per time as a .png file. I tried to parse the website to get the real downloading urls for all the slices, but when I used response.get(url) to download them, I got a 403 error code, which means permission denied. I tried to add web browser header info in the parameter "headers" to mimic the real web browser; however, it sometimes works, and sometimes doesn't. So eventually, I just used one slice (I can manually download all the slices, but I guess that it is not the purpose of the question.).

   For **dataset 2**, it is also a multi-slice image, but as .dm3 files. I used the ftp site on the website to download all the 16 slices.

   For **dataset 3**, it is a multi-slice image in the format of .tif.

   I combined all the urls from the three datasets and developed a function to download them parallelly (ThreadPool module).

   For **dataset 4**, it is stored on AWS. I can use quilt3 to download it (shown in the code). However, it is not necessary since it is in the .zarr format. I can use the "Zarr" package to read the data in the cloud (shown in the code). I originally ran into problems to execute the commands from 'Zarr" since I used Kaggle, but its "s3fs" package was so old and not compatible. Eventually, I figured it out. But it took me one additional day.

   For **dataset 5**, it is stored in Google Cloud Storage. It is not necessary to download the data. I used the "tensorstore" package to process the data and extracted a random 1000x1000x1000 pixel crop region.

2. Identify and consolidate meta-data entries across these datasets, such as resolution or pixel type (some metadata entries will be available only for some of the datasets), list everything can you

extract, find corresponding entries between datasets. Share the consolidated entry table in the GitHub repo, together with a quick summary of how you did it.

A: All the metadata info is summarized in "HHMI_interview_exercises.xlsx".

For **dataset 1**, since it is a .png file, I used PIL to read it, but didn't find a metadata tag associated with it. The only thing I can extract is the shape of the array. But I found the metadata file from https://idr.openmicroscopy.org/webclient/?show=dataset-10740. I can download the original metadata file from the Acquisition tab on the webpage. From the file, the image size is 1121x775, however, the downloaded file shows 1425x913. Also, the max and min values are different.

For **dataset 2**, I used a package called "ncempy" to read the .dm3 data. I can extract the metadata such as pixel unit, pixel size, and coordinates. The rest of the information was obtained from the website.

For **dataset 3**, I used PIL.TiffTags to extract the metadata. The rest of it is from the website.

For **dataset 4**, I used .info_complete attribute from the "zarr" package to extract the metadata.

For **dataset 5**, the "tensorstore" package can provide the metadata automatically.

I summarized the most relevant meta info from each dataset in the excel file. But I also saved others in the output of the code (It is a python notebook, so if you can open it and you will be able to see the output.) because they are very tedious, especially for dataset 5 and not all of the datasets have them.

3. Think about how to make these image dataset and metadata available for ingestion by an AI/ML pipeline. The software should be able to provide access to each image in a block-wise manner (e.g. the ML software will request 128x128x128 pixel block at various locations within each dataset). How would you go about designing and implementing such a software? The idea here is not to implement it, but to think it through and outline how you would do it, please create a PDF or similar.

   A: First of all, in order to design the software like this, we need to consult with different lab users to understand their needs and add them to the requirements. There will be several steps that will need to take into consideration.

   1. Storage Format Choices: since the size of the electronic microscopy is usually large, the ML algorithm cannot take the entire dataset at once. The software needs to have the capability for chunked and multidimensional access: After doing this exercises, I feel that converting images to a Zarr or N5 format is a good choice, like dataset 4. Right now, Zarr seems to be a good candidate since it is open access, scalable, cloud-optimized, stores chunks as independent files or in object storage and the data can also be compressed. Also, the metadata can also be saved in the zarr file format.

2. Storage backend: it needs to have the capability to store the data locally and also in the cloud, for example, AWS.
3. Design a light-weight API to access the data, blocks, and metadata.
4. Integration with the ML pipeline: Implement a PyTorch dataset class to seamlessly fetch blocks during training or inference.
5. Security and access control: authenticate users / jobs accessing the dataset or provide access tokens or API keys.