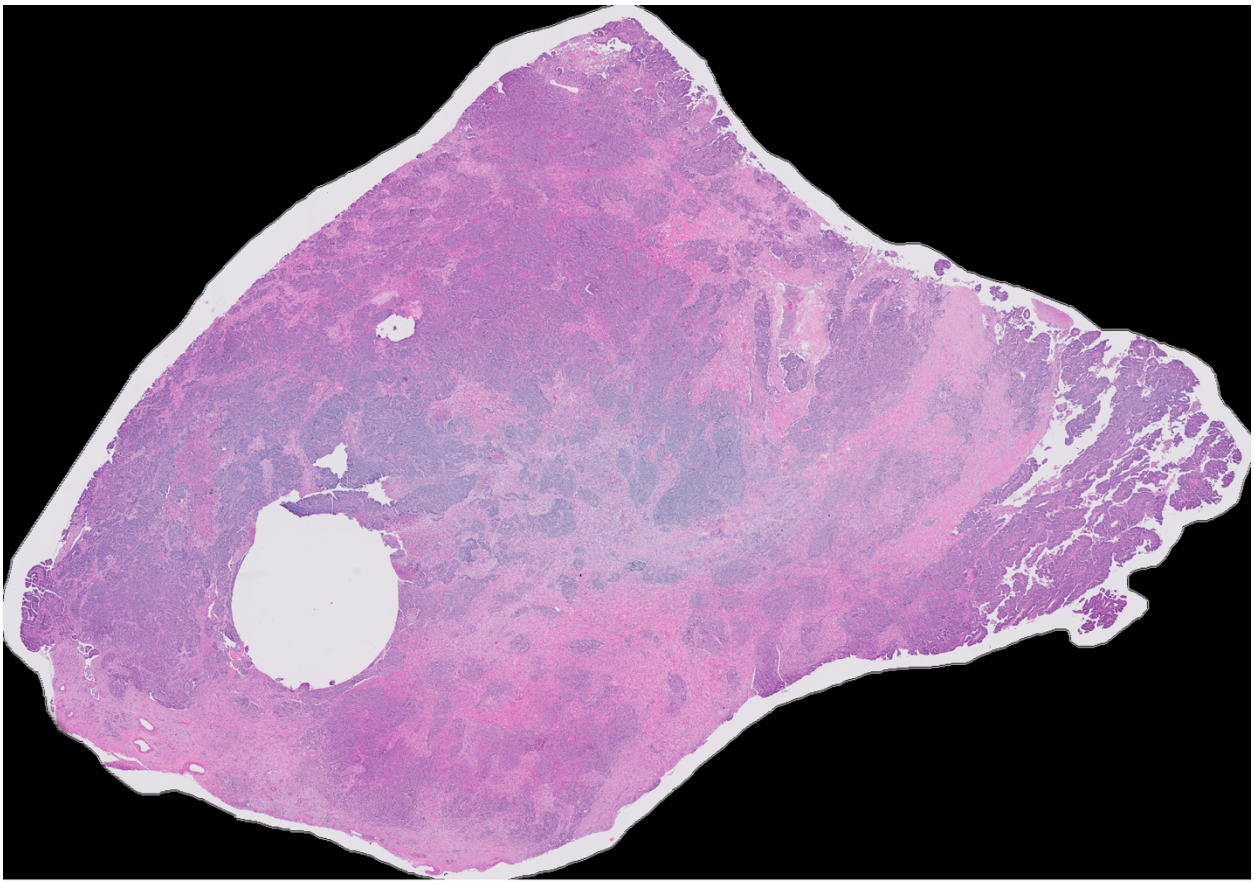
1. Members
    a. Mikey Joyce; Master's student
2. Title
    a. UBC Ovarian Cancer Subtype Classification and Outlier Detection (UBC-OCEAN)
    b. https://www.kaggle.com/competitions/UBC-OCEAN/overview
3. Scope
    a. This dataset is from a Kaggle competition. The data within are images of microscopy scans of biopsy samples. Each image is said to have a different type of ovarian cancer: high-grade serous carcinoma (HGSC), clear-cell ovarian carcinoma (CC), endometrioid (EC), low-grade serous (LGSC), mucinous carcinoma (MC), or a rare subtype (Other).
    b. One of the main issues with this dataset are the images. Since the data is collected from various hospitals a lot of the images are of inconsistent size. Not only that a lot of these images are absolutely massive, running at 1-2 GB per image (the entire dataset is almost 800 GB). That is not ideal as it is essentially the same as looking at a satellite image of NYC and trying to identify a person on the ground (check the image at the bottom). There are lots of moving parts to a Kaggle competition and the following parts are what I think would be required to complete the competition:
        i. Preprocessing
        ii. Identify the cancerous regions
        iii. Train a CNN to classify the images
        iv. Use the trained model to perform classification on the test set
    c. I aim only to complete the first two parts I have outlined for this project, and later after this class has ended, I will implement the classification.
    d. So, the scope of this project will be to import the given images, apply necessary preprocessing, find a subset of each image that has cancerous cells, and then save this new image. All of the resulting subset images should have the same dimensions and ideally have a good number of cancerous cells within to be determined as satisfactory. To check whether or not I have accomplished what needed to be accomplished I will have to check each resulting image manually to verify if said image has cancerous cells within and I will then enter into a CSV whether the result was satisfactory or not. I can then use this CSV to generate metrics on how my implementation performed.
    e. I am not sure what exact method will accomplish what I need to accomplish, but a possible idea is to have a sliding window Laplacian of

Gaussian blob detector, pick the window/region that maximizes the number of blobs.



Sample image from the UBC-OCEAN dataset. This image is 484,109,318 pixels large and there are many more that are much larger (this is only a screenshot of the image so resolution may not be the same). The ideal result image that would be produced would ideally be much smaller.