Homework 4: Object Detection & Segmentation

The use of existing models is an important thing to learn. Not every model you need to make has to be from scratch, take advantage of past work and efforts of others. Utilization of model architectures, pretrained models, or public datasets help aide in advancements.

Part 1 – Semantic Segmentation:

Goal: Train a popular semantic segmentation architecture, UNET, for generating masks of bubbles in boiling images. (Which can be used to predict vapor fraction. We won't go over this but if you take interest there are several publications out there that report it.)

Data: This data is generated from in lab pool boiling experiments using a high-speed camera and labeled manually labeled using labelme.

Model Inputs: Gray scale images during boiling

Model Outputs: A matrix of the same size where each value is between 0 and 1. These are the pixel classifications. A pixel is either classified as bubble or background.

Assignment-pt1:

1. Train the UNET semantic segmentation model architecture (found in the github) on the boiling images and corresponding masks
2. Display at least two input images, predicted masks, and true masks
3. With the testing data, compute the pixel accuracy

Part 2 – Object Detection:

Goal: Use transfer learning with a pretrained model (yolov8) and public dataset for object detection.

Data: You will choose a public dataset from roboflow for object detection: https://public.roboflow.com/object-detection/. Roboflow is cool because it offers several datasets that people have made and gives you different ways to download the data. There are quite a few object detection models out there that have their own format for inputs. We will be using the Yolov8 object detection model so that is the type you should download.

Model Inputs: Images with some specific objects in them

Model Outputs: Bounding box coordinates that enclose object, class prediction of object, and confidence values.

Assignment-pt2:

1. Choose a public dataset and download in the proper format. (YOLOv8)
2. Use transfer learning with a pretrained Yolov8 model from ultralytics.
3. Display at least two testing input images with predicted masks.
4. Performance Metrics:
   a. mAP50 and mAP50-90