# Detailed Instructions for Running Code

## Instructions

1. **Set up your environment:**

   - This coursework is entirely written by PyTorch and environment "*comp*0197_*cw*1_*pt*" is used in this coursework
   - packages used in Colab environment:

     ```
     torch: 2.0.0
     torchvision: 0.15.1
     nunpy: 1.22.4
     pandas: 1.4.4
     tqdm: 4.65.0
     PIL: 8.4.0
     matplotlib: 3.7.1
     ```

   - We used three packages that's not in the environment, and it is listed below:

     - segmentation_models_pytorch:

       ```
       !pip install git+https://github.com/qubvel/segmentation_models.
           pytorch
       ```

     - torchinfo (1.7.2):

       ```
       !pip install torchinfo
       ```

     - transformer (4.27.4):

       ```
       !pip install transformers
       ```

2. **Upload the dataset:**

   - Locate the dataset files (img.npy, label.npy, size.npy) on your computer.
   - Create a folder named "0197Group" in your Google Drive.
   - Inside "0197Group", create another folder named "Dataset".
   - Upload the dataset files to the "Dataset" folder in Google Drive.

3. **Mount Google Drive in Colab:**

   - Open Google Colab and create a new Jupyter Notebook.
   - Copy and paste the provided code into the notebook.
   - Run the code cell that mounts your Google Drive to make the dataset accessible in Colab:

     ```
     from google.colab import drive
     drive.mount('/content/drive')
     ```

4. **Unzip the dataset files:**

   - Run the code cell that unzips the dataset files, which will extract the data and make it available for processing:

```
!unzip '/content/drive/MyDrive/0197Group/Dataset/img.npy.zip'
!unzip '/content/drive/MyDrive/0197Group/Dataset/label.npy.zip'
!unzip '/content/drive/MyDrive/0197Group/Dataset/size.npy.zip'
```

5. **Load the pre-trained model:**
   - Upload the pre-trained model file (yingshuai_pretrain_AB.pth) to the "0197Group" folder in Google Drive.
   - The provided code will load the pre-trained model when it executes.
6. **Run the code:**
   - Execute the entire code in the Jupyter Notebook. The code will perform data loading, preprocessing, model training, and evaluation.
   - Make sure to execute the cells in the correct order, following the flow of the provided code.
7. **Monitor the training progress:**
   - Observe the printed training loss and validation loss values during the training process. This will give you an idea of how well the model is learning from the data.
   - Watch for signs of overfitting, such as when the training loss decreases, but the validation loss increases.
8. **Results:** We split the data into 60% unlabelled data and 40% labelled data. Within the labelled data, we further divided it into 80% for training, 5% for validation, and 15% for testing.
   - baseline: training with 80% labelled training data.
   - pretrain: pretrained with all 60% unlabelled data and 80% labelled training data.
   - baseline_OEQ: code for open-ended question: is there margin effect for increasing labelled data? Inside, based on original baseline, we tried to train with 10%, 30%, 50%, 70%, and 90% of training data.
   - upperbound: load the pretrain model, then fine-tune with all unlabelled data and labelled training data.
   - semi_supervised: load the pretrain model, then fine-tune with labelled training data.
   - Results Analysis: load all the models above and provide masks of sample images, as well as a table displaying the mean IoU, precision, recall, and F1-score of each model.