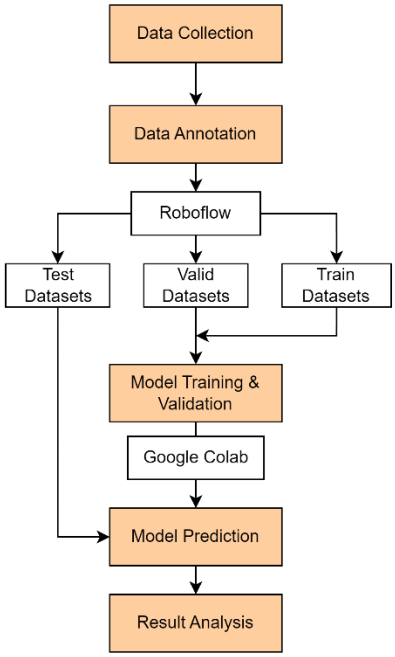
WEEK 6 - WEEK 9

Objectives:

1. Create prediction code in Google Colab to test the prediction of the model
2. Train and test using augmented and non-augmented datasets
3. Create code in Google Colab to train the YOLOv8-seg model
4. Proceed with data annotation



While proceeding with the data annotation, a training and prediction code was created through Google Colab. The reason to use Google Colab is because of its cloud GPU which can speed up the training and prediction process of the model.

Student's Name/Signature: MUHAMMAD SHAHIRUL AFIQ BIN SUKAIRI **Date: 20/6/2024**	**Supervisor's Name/Signature:** **Date:**

WEEK 6 - WEEK 9

PredictionCode.ipynb

File Edit View Insert Runtime Tools Help Last edited on June 11

Comment Share M

+ Code + Text Connect T4 Gemini

```
# Mount Google Drive to access the dataset and weights
from google.colab import drive
drive.mount('/content/gdrive')
```

Show hidden output

```
# Install necessary dependencies and import libraries
!pip install ultralytics

from ultralytics import YOLO
from pathlib import Path
import numpy as np
from sklearn.metrics import f1_score, accuracy_score      # To access performance metrics
from tqdm import tqdm                                     # To have progress bar

import cv2
import glob    # Use to get all .jpg files
import os      # Create new directory
```

Show hidden output

Prediction code

- use glob to get all .jpg files

```
# Define paths
custommodel_path = "/content/gdrive/My Drive/FYP/DATASETS_V2/version1/100epochs/runs/segment/train/weights/best.pt"
test_dataset_path = Path("/content/gdrive/My Drive/FYP/DATASETS_V2/version1/test/images")

testimage_path = glob.glob(str(test_dataset_path / "*.jpg"))

# Saving results in Gdrive
results_save_path = Path("/content/gdrive/My Drive/FYP/DATASETS_V2/version1/100epochs/predictions")

# Create the results directory if it does not exist
os.makedirs(results_save_path, exist_ok=True)

# glob to get all .jpg files in the directory
image_paths = glob.glob(str(test_dataset_path / "*.jpg"))
```

```
model = YOLO(custommodel_path)
```

+ Code + Text

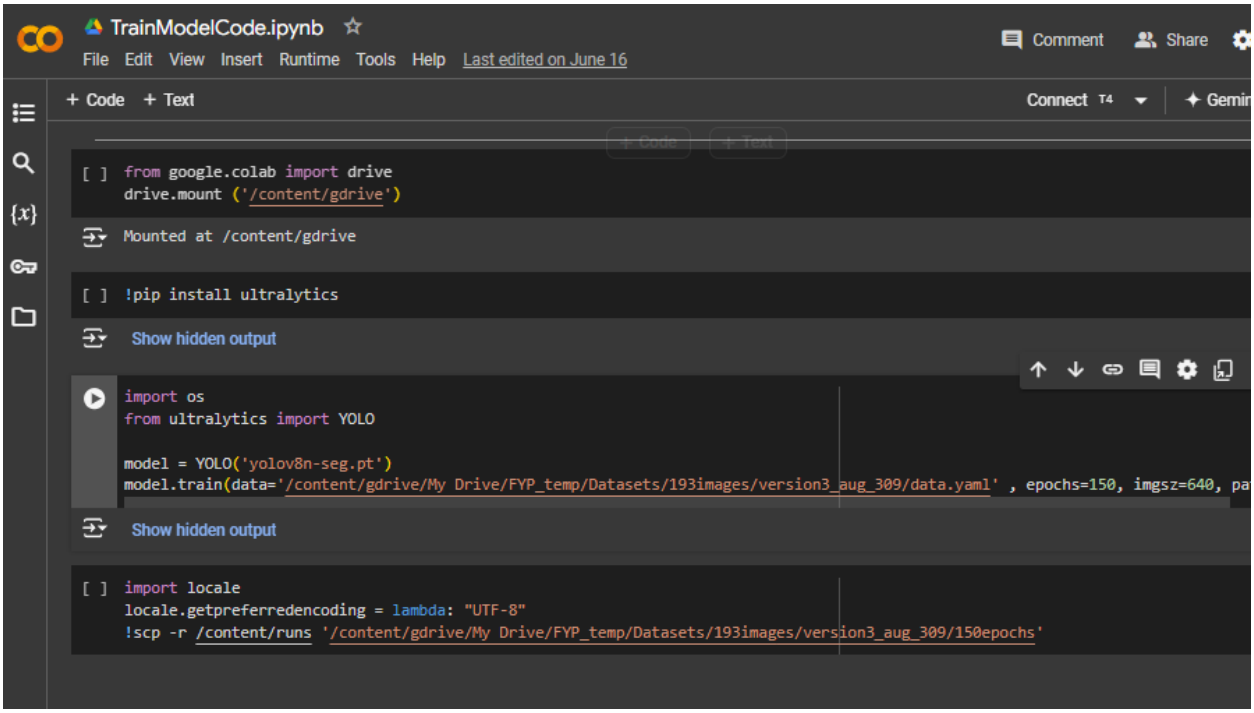
```
# Iterate images and process the images
for image_path in image_paths:
    # Load the image
    image = cv2.imread(image_path)

    # Ensure the image is loaded successfully
    if image is None:
        print(f"Failed to load image: {image_path}")
        continue

    # Perform segmentation using YOLOv8
    results = model.predict(image)

    # Draw the predictions on the image
    for result in results:
        if hasattr(result, 'masks') and result.masks is not None:
            for mask in result.masks.data:

                mask = mask.cpu().numpy()

                # Convert the mask to the same size as the image
                mask_resized = cv2.resize(mask, (image.shape[1], image.shape[0]))


                image[mask_resized > 0] = [0, 255, 0]  # Overlay green mask colour

    # Extract the filename from the image path
    filename = os.path.basename(image_path)
    # Set the save path for the result
    save_path = results_save_path / filename

    # Save the result to the specified path
    cv2.imwrite(str(save_path), image)
    print(f"Saved result to: {save_path}")
```

Show hidden output

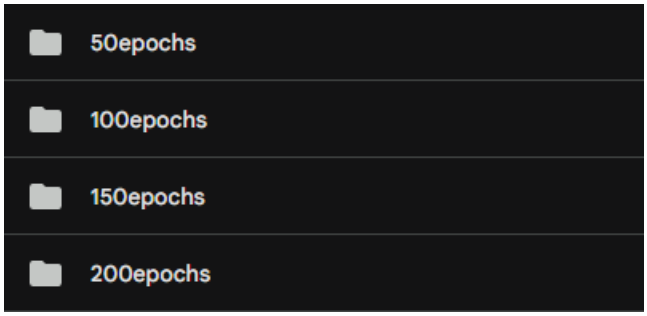
The code for prediction of the model

Student's Name/Signature: MUHAMMAD SHAHIRUL AFIQ BIN SUKAIRI **Date: 20/6/2024**	**Supervisor's Name/Signature:** **Date:**

WEEK 6 - WEEK 9



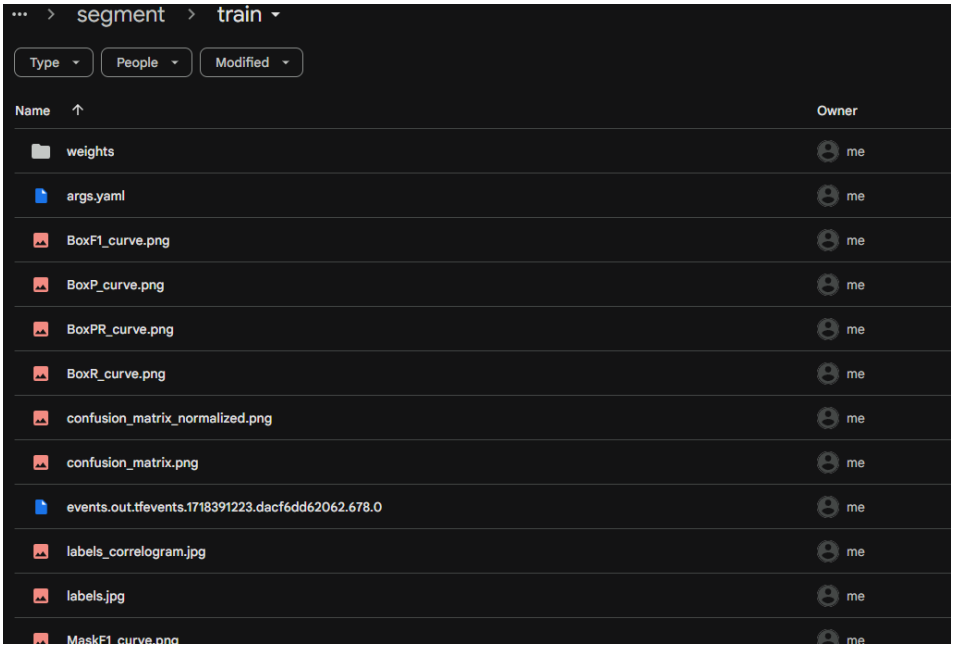
Training code



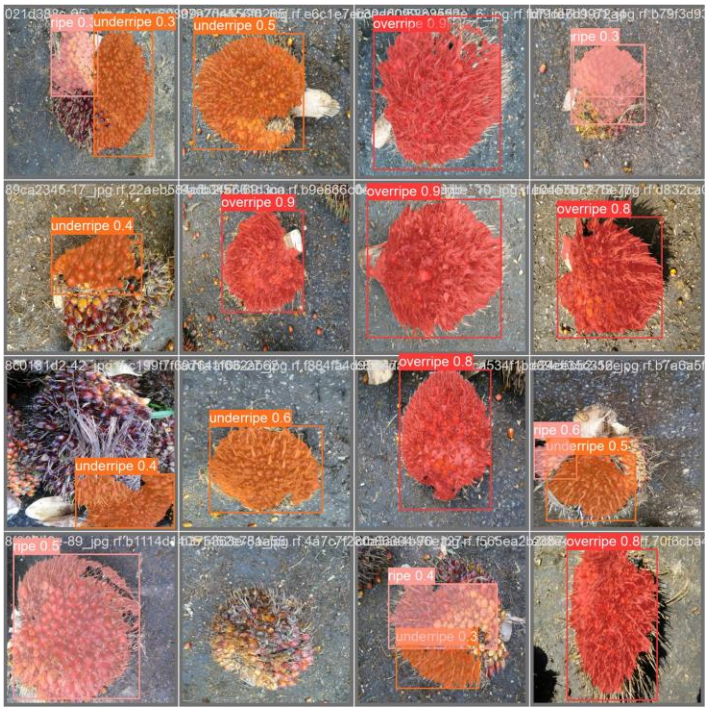
50, 100, 150, 200 epochs are trained

Student's Name/Signature: MUHAMMAD SHAHIRUL AFIQ BIN SUKAIRI **Date: 20/6/2024**	**Supervisor's Name/Signature:** **Date:**

WEEK 6 - WEEK 9



Results after training



Segmentation results after training

Student's Name/Signature: MUHAMMAD SHAHIRUL AFIQ BIN SUKAIRI **Date: 20/6/2024**	**Supervisor's Name/Signature:** **Date:**

WEEK 6 - WEEK 9

Student's Name/Signature: MUHAMMAD SHAHIRUL AFIQ BIN SUKAIRI  **Date: 20/6/2024**	**Supervisor's Name/Signature:** **Date:**