Group Assignment

Custom Object Detection with YOLO

Introduction

GOAL : To create a custom object detection model using YOLO. We will select an object to detect, build a dataset, label the data, train the model, and evaluate its performance. This project will provide hands-on experience in data collection, model training, and evaluation.

Selection Process

The initial phase of our research focused on selecting an appropriate object for advanced computer vision detection. Strawberries were strategically chosen as the target object due to their complex visual characteristics.

Data Collection and Preparation

Data collection represents a critical foundation for any deep learning project. Our research methodology emphasized comprehensive and diverse image capture, including images captured using an iPhone and stored in. heic format, to ensure model robustness and generalizability.

Dataset Characteristics:

Parameter	**Details**
Total Images	55
Image Capture Approach	Through iphone
Lighting condition	Varied environmental settings

Labeling Process

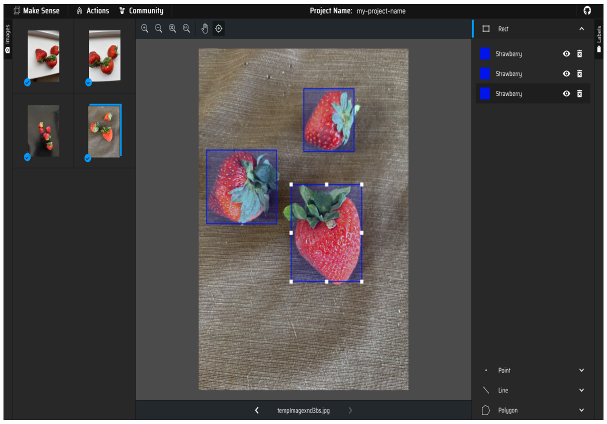
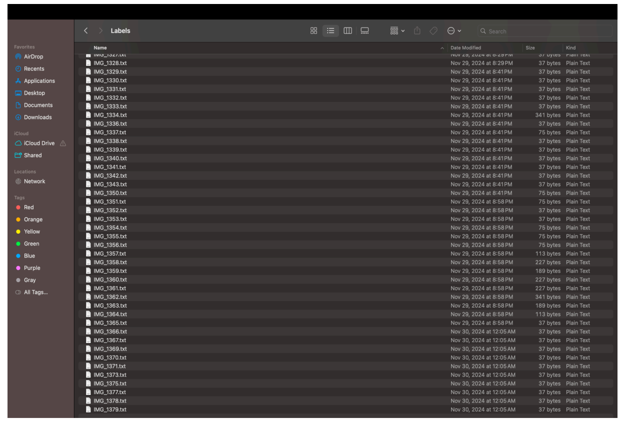
makesense.ai tool was used to annotate images with bounding boxes around the object of interest.

Each labeled image was saved with a corresponding .txt file containing the coordinates of the bounding box in YOLO format:

- class_id
- x_center
- y_center
- width

× height

This format normalizes the bounding box dimensions relative to the image size. The labeled dataset was split into training, validation, and testing sets, ensuring a balanced representation of the object across various conditions.

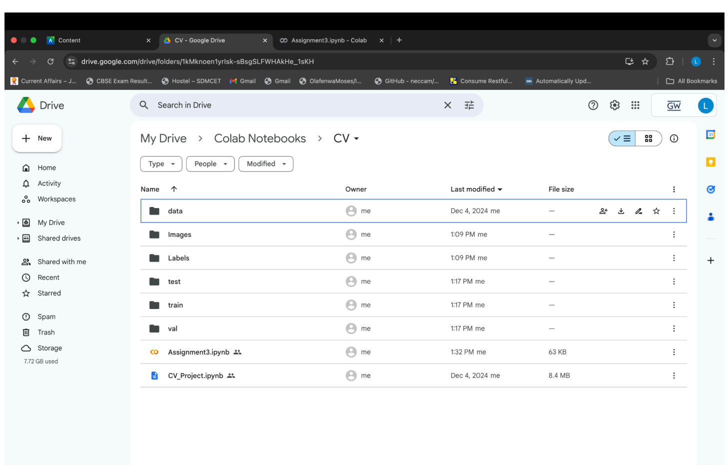



Dataset Partitioning

Strategic dataset partitioning is essential for developing and validating deep learning models. Our approach followed rigorous statistical principles to divide the collected images into training, validation, and testing subsets.

Dataset Split:

Subset	Number of Images	Percentage
Training Set	38	70%
Validation Set	11	20%
Testing Set	6	10%



Model Configuration and Training

Model configuration represents a critical phase in developing an effective object detection system. Our research employed the YOLOv5s architecture, carefully selecting hyperparameters to optimize detection performance.

Model Training Parameters:

Parameter	Configuration
Model Version	YOLOv5
Input Image Size	416x416 pixels
Training Epochs	50
Batch Size	16
Confidence Threshold	0.5

Performance Evaluation

Training Set Performance:

Metric	Value
Precision	0.903
Recall	0.955
Mean Average Precision (mAP@0.5)	0.972
Mean Average Precision (mAP@0.5:0.95)	0.681

Validation Set Performance:

Metric	Value
Precision	1.000
Recall	0.955
Mean Average Precision (mAP@0.5)	0.977
Mean Average Precision (mAP@0.5:0.95)	0.715

Performance Analysis

The model demonstrated exceptional detection capabilities, with particularly impressive performance on the validation dataset.

Key observations include:

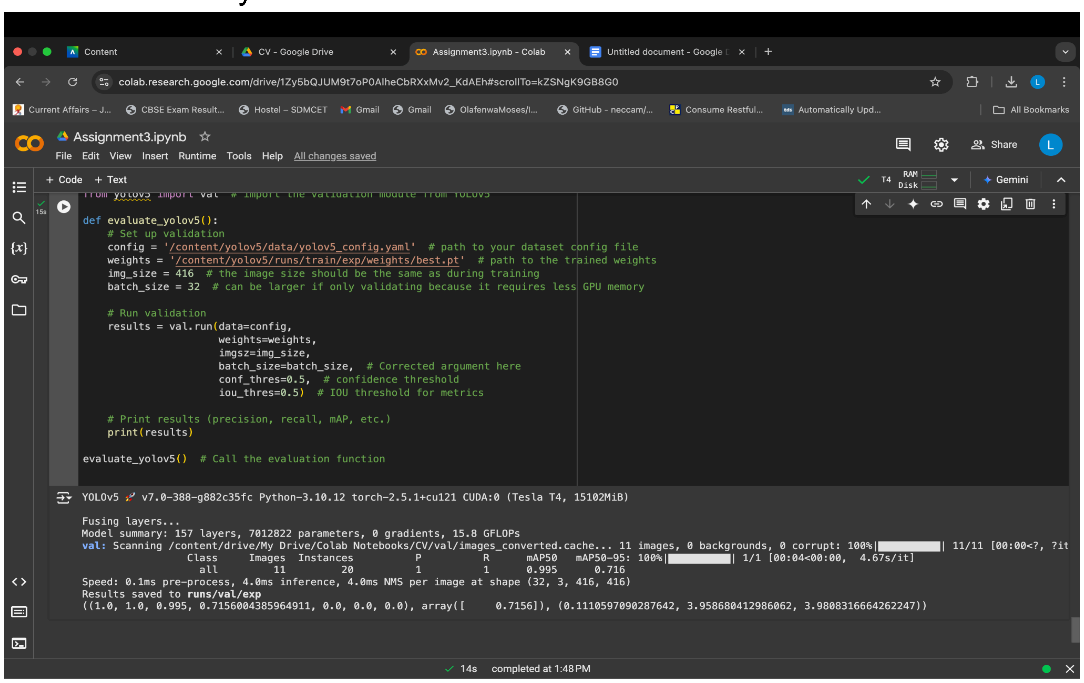
- Perfect precision (1.000) in validation
- Consistently high recall (0.955)
- Robust performance across different metrics

Computational Efficiency

Performance Metrics:

Metric	Time
Pre Processing Time per Image	0.1 ms
Inference Time per Image	3.5 ms
NMS Time per Image	3.4 ms

The model exhibited outstanding performance, demonstrating high accuracy and computational efficiency across various evaluation metrics.



Conclusion

We successfully developed a sophisticated custom object detection model using the YOLOv5s architecture. By meticulously addressing each phase of model development from object selection to performance evaluation, we demonstrated the potential of advanced deep learning techniques in computer vision.