Object Detection using YOLO

Machine Learning Internship Assignment

Author: Induwara Abhisheka
Date: February 14, 2026

1 Introduction

This project was conducted as part of the Machine Learning Internship program. The primary objective was to develop a robust object detection system using the YOLO (You Only Look Once) framework. The task involved labeling a real-world dataset, preprocessing the data, and training a deep learning model capable of detecting multiple gemstone types.

The project emphasizes practical implementation of deep learning techniques for real-world applications, including dataset preparation, model optimization, and performance evaluation. This work demonstrates practical understanding of computer vision pipelines, including data preparation, model training, and evaluation using state-of-the-art object detection frameworks.

2 Dataset Description

The dataset was obtained from Google Drive and processed using Kaggle for training and evaluation.

Dataset Link:
https://drive.google.com/drive/folders/1So1Ie3cpdlbhhINC8g1OueETPDoStIfc?usp=share_link

2.1 Classes

The dataset contains 29 gemstone classes.According to the requirements of the assignment 8 classes was used for the project:

- Tigers_Eye
- Obsidian
- Lapis_Lazuli
- Rose_Quartz
- Red_Jasper
- Clear_Quartz
- Amethyst
- Aventurine

2.2 Dataset Statistics

Table 1: Overall Dataset Statistics

Metric	Value
Total Images	469
Total Objects	1074
Empty Labels	0

2.3 Data Splitting Strategy

A deterministic and class-balanced splitting strategy was used instead of random splitting. This approach ensures better distribution of classes across training and validation sets, reducing bias and improving generalization.

2.4 Train Split

Table 2: Training Split Summary

Metric	Value
Images	371
Total Objects	815

Table 3: Training Class Distribution

Class	Objects
0	172
1	88
2	102
3	125
4	110
5	63
6	69
7	86

Classes 5 and 6 have relatively fewer samples. Since all available images were already used, further balancing was limited by dataset size.

2.5 Validation Split

Table 4: Validation Split Summary

Metric	Value
Images	98
Total Objects	259

Table 5: Validation Class Distribution

Class	Objects
0	52
1	35
2	44
3	37
4	34
5	17
6	15
7	25

To mitigate imbalance, duplication of underrepresented classes was selectively applied.

3 Data Labeling

Label Studio was used to annotate the dataset with bounding boxes.

3.1 Labeling Process

- Annotated gemstone instances using bounding boxes
- Exported annotations in YOLO with Images format
- Performed manual verification to check the quality of labeling
- Ensured consistency and correctness across all labels

4 Model Training

Multiple YOLO variants were evaluated, including YOLOv8 and YOLOv8s. The best performance was achieved using YOLOv8m.

4.1 Training Configuration

Table 6: Training Configuration

Parameter	Value
Model	YOLOv8m
Epochs	150
Image Size	640
Optimizer	AdamW

4.2 Data Augmentation

- Mosaic augmentation
- MixUp
- Random scaling and translation
- Horizontal flipping

- HSV color augmentation

4.3 Optimization Strategies

- Deterministic class-balanced splitting
- Targeted duplication of minority classes
- Strong augmentation to reduce overfitting

5 Evaluation and Results

5.1 Overall Performance

Table 7: Overall Model Performance

Metric	**Value**
Precision	0.737
Recall	0.775
mAP@50	0.763
mAP@50–95	0.604

5.2 Per-Class Performance

Table 8: Per-Class Detection Performance

Class	**mAP@50**
Tigers_Eye	0.917
Obsidian	0.894
Lapis_Lazuli	0.873
Rose_Quartz	0.729
Red_Jasper	0.756
Clear_Quartz	0.611
Amethyst	0.553
Aventurine	0.769

5.3 Discussion

- High accuracy achieved for well-represented classes
- Moderate performance for balanced classes
- Lower performance in Clear_Quartz and Amethyst due to limited data

6 Conclusion

This project successfully developed a YOLO-based object detection model with strong performance.Can improve the accuracy of the model by using a larger reliable dataset.

6.1 Key Findings

- Class-balanced splitting improves fairness
- Augmentation enhances generalization
- Dataset size impacts minority class performance

6.2 Future Work

- Increase dataset size
- Use larger models (YOLOv8l was not used in this project due to limitations of the dataset)
- Perform hyperparameter tuning

Declaration

I hereby declare that this assignment is my original work and has been completed solely for the purpose of the Machine Learning Internship program.