High Impact Skills Development Program
AI & Data Science

Object Detection for Road Turn Detection in Gilgit using YOLOv8

Submitted By: Saleem Ullah Baig

Reg # GIL-DSAI-184

GitHub Profile: https://github.com/saleem98/DSAI.git

Team Members are:

Paras Makhootee

Sania Imam

Learning Objectives:

I. Understanding Object Detection with YOLOv8
II. Data Preparation and Annotation for Object Detection
III. Model Training and Hyperparameters Tuning
IV. Evaluating Object Detection Models
V. Implementing Object Detection in Real-World Scenarios
VI. Code Documentation and Report Writing
VII. Exploring Advanced Techniques for Model Improvement

Project Overview

This project aims to detect various road conditions (e.g., right turn, left turn, straight, unexpected) in road images using an object detection model trained with YOLOv8. The model uses images labeled with bounding boxes, converted to YOLO format, to improve safety and aid in navigation. The goal is to develop a model that accurately identifies road features in challenging environments like Gilgit.

Data Preparation and Preprocessing

Dataset and Classes

The dataset consists of three splits: train, valid, and test. Each image is labeled with one of four classes:

a) Right-Turn
b) Left-Turn
c) Straight
d) Unexpected

Images are labeled with bounding boxes in CSV format, which were converted to YOLO format. The annotations were transformed to YOLO's bounding box format with normalized center coordinates (x, y) and dimensions (width, height).

Data Conversion Code Highlights

The conversion script reads bounding boxes from CSV files and converts them into .txt files per YOLO format:

```
# Convert CSV annotations to YOLO format
for _, row in annotations.iterrows():
    class_id = class_names.index(row['class'])
    center_x = (row['xmin'] + row['xmax']) / 4 / img_width
    center_y = (row['ymin'] + row['ymax']) / 4 / img_height
    width = (row['xmax'] - row['xmin']) / img_width
    height = (row['ymax'] - row['ymin']) / img_height
```

Model Training Configuration

YOLOv8 was trained with the following configurations:

Model: yolov8m.pt (medium model) initially, later experimented with yolov8s.pt (small model).

Epochs: First run with 20 epochs, followed by additional 10 epochs with adjusted parameters.

Batch Size: 16 initially, then adjusted to 5.

Image Size: 640x640

Learning Rate: Adjusted in later training to 0.001.

Training and Validation Results

Initial Training Results (20 Epochs, Batch Size 16)

The initial training run resulted in the following metrics:

Precision: 74.1%

Recall: 15%

mAP50: 0.269

mAP50-95: 0.155

Interpretation: The model demonstrated reasonable precision but had low recall, indicating that while predictions were often correct, the model did not detect all possible instances in the images.

Test Evaluation Metrics

Precision: 74.2%

Recall: 15%

mAP50: 0.268

mAP50-95: 0.155

Second Training Run (10 Epochs, Batch Size 5, LR 0.001)

After modifying the batch size and learning rate, metrics dropped notably:

Precision: 10.3%

Recall: 28.9%

mAP50: 0.147

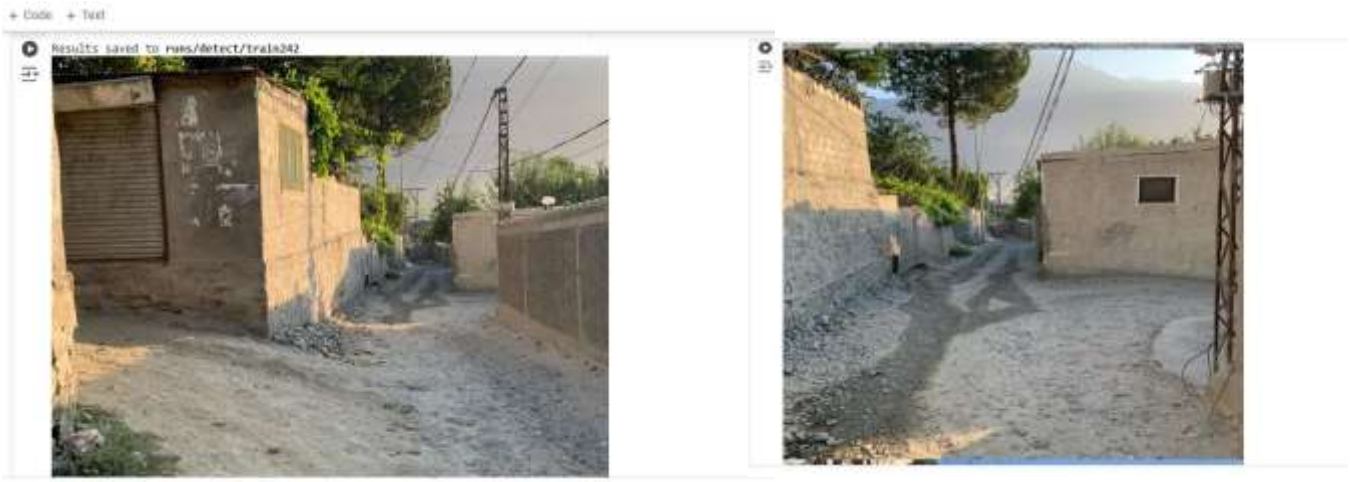
mAP50-95: 0.029

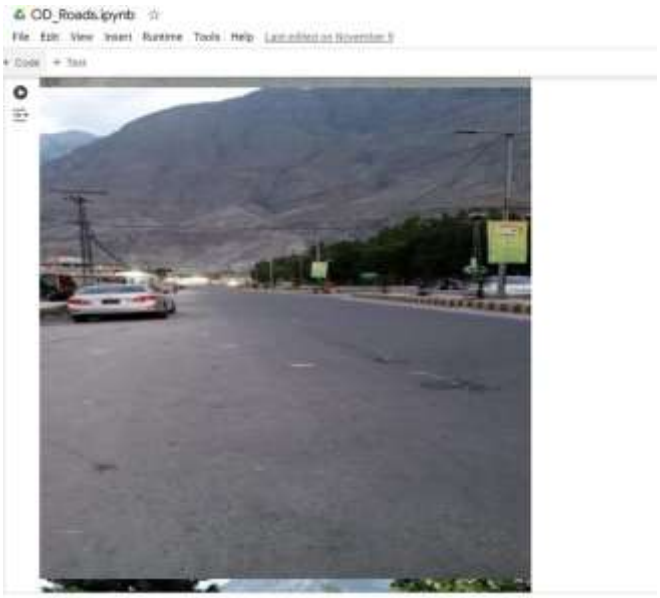
Interpretation: This configuration did not yield improved results, suggesting the model requires a larger batch size and further tuning for optimal performance.

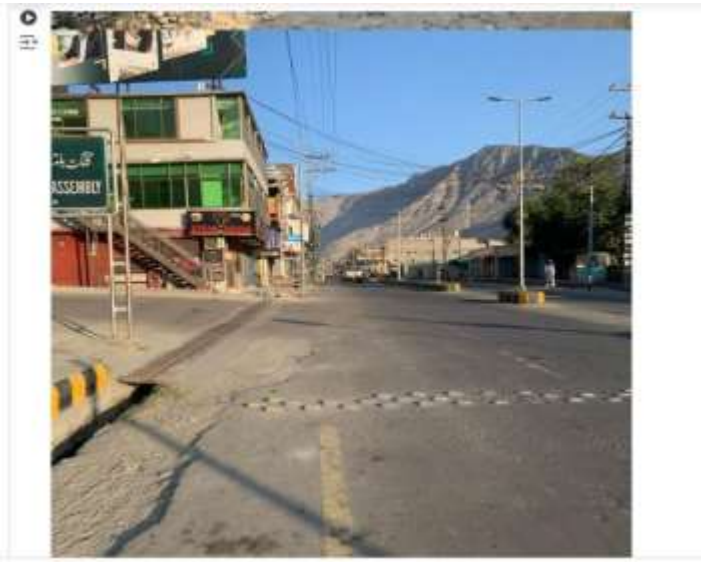
Model Evaluation and Prediction Results

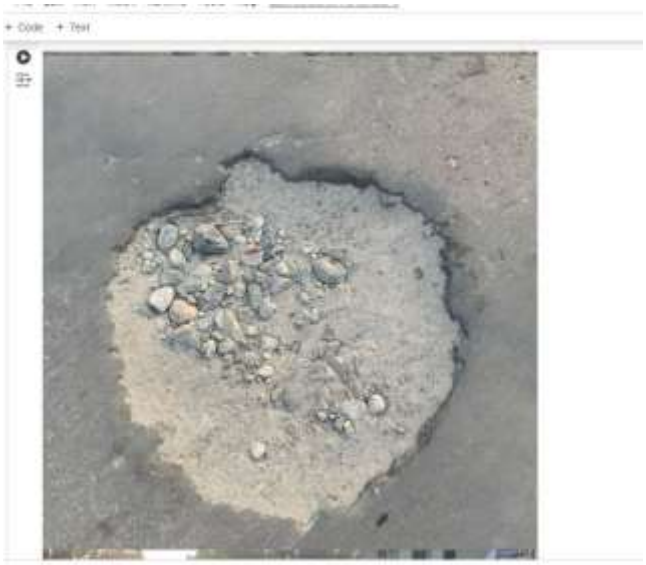
The model was evaluated on the test images, with predictions saved for visual analysis. The following code snippet was used for generating and displaying predictions:

```
# Use the model to make predictions on test data
results = model.predict("/path/to/test/data", save=True)
for result in results:
    result.show()
```









Conclusion

The YOLOv8 model showed promising initial results with high precision but struggled with recall, suggesting it identifies turns accurately but misses some instances. With additional tuning, augmentation, and possible model adjustments, it is expected that the model's detection accuracy for road turns can be significantly improved. This model has potential applications in real-time road detection systems, enhancing navigation and road safety in regions like Gilgit

Referencing:

- Git-Hub
- YouTube : https://www.youtube.com/

THE END