High Impact Skills Development Program in Artificial Data Science and Intelligence

Project Title: **Object Detection GB Roads Turns**

Submitted By: Faiza Ali – Section 01

Roll No: GIL23047

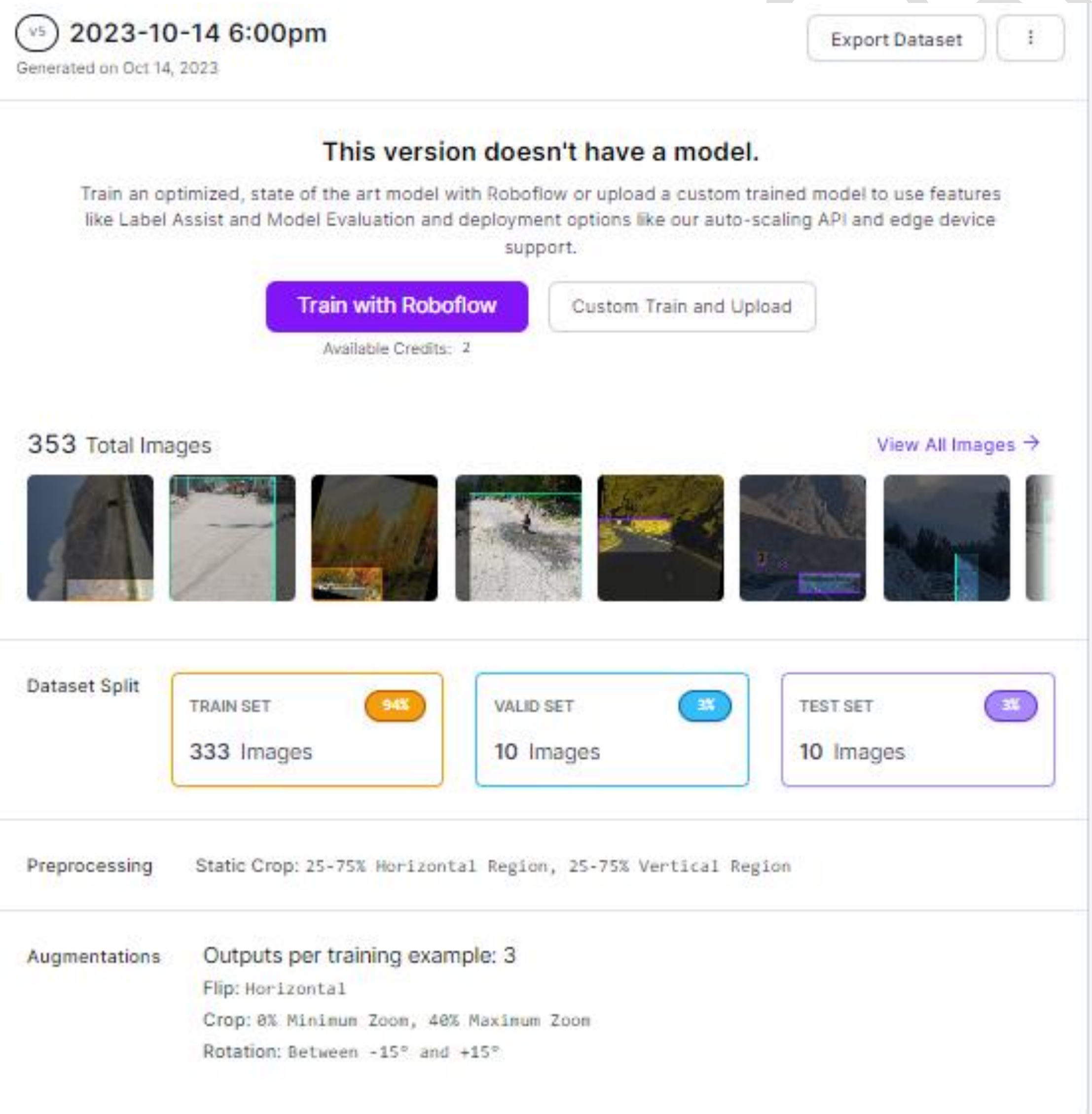
GitHub Link: https://github.com/FaizaAli-DS/Object_Detection_GB_Roads

Introduction:

I'm excited to present an overview of my project, which aims to develop an object detection model for identifying various road turns in images captured on the roads in Gilgit. The project is essential for ensuring road safety and optimizing traffic flow. In this report, I will walk you through the key steps of this project, highlighting the technical aspects involved.

Annotation using Roboflow

Annotating images is a fundamental step in building an object detection model. It involves labeling the images with information about the objects or features of interest, in our case, road conditions and turns. For this purpose, I leveraged Roboflow, a widely used platform for annotating and managing labeled image datasets. Roboflow offers an efficient and user-friendly interface for annotating images, and it supports various annotation formats.



v5 2023-10-14 6:00pm

Generated on Oct 14, 2023

This version doesn't have a model.

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

Available Credits: 2

353 Total Images

Dataset Split

TRAIN SET 94%	VALID SET 3%	TEST SET 3%
333 Images	10 Images	10 Images

Preprocessing: Static Crop: 25-75% Horizontal Region, 25-75% Vertical Region

Augmentations: Outputs per training example: 3

Flip: Horizontal

Crop: 0% Minimum Zoom, 40% Maximum Zoom

Rotation: Between -15° and +15°

Roboflow simplifies the annotation process by allowing me to label images with information about road conditions, such as right turns, left turns, and straight roads. This annotated dataset serves as the foundation for training our object detection model, enabling it to recognize and classify these road conditions accurately.

Model use - YOLOv8

1. Description of YOLOv8

In this project, I've chosen to implement the YOLOv8 (You Only Look Once) object detection model as the core technology for precisely localizing and classifying road turns in the images. YOLO is renowned for its speed and accuracy in object detection. The "v8" indicates the version of YOLO I used, which is the most recent iteration as of my knowledge cutoff date in September 2021.

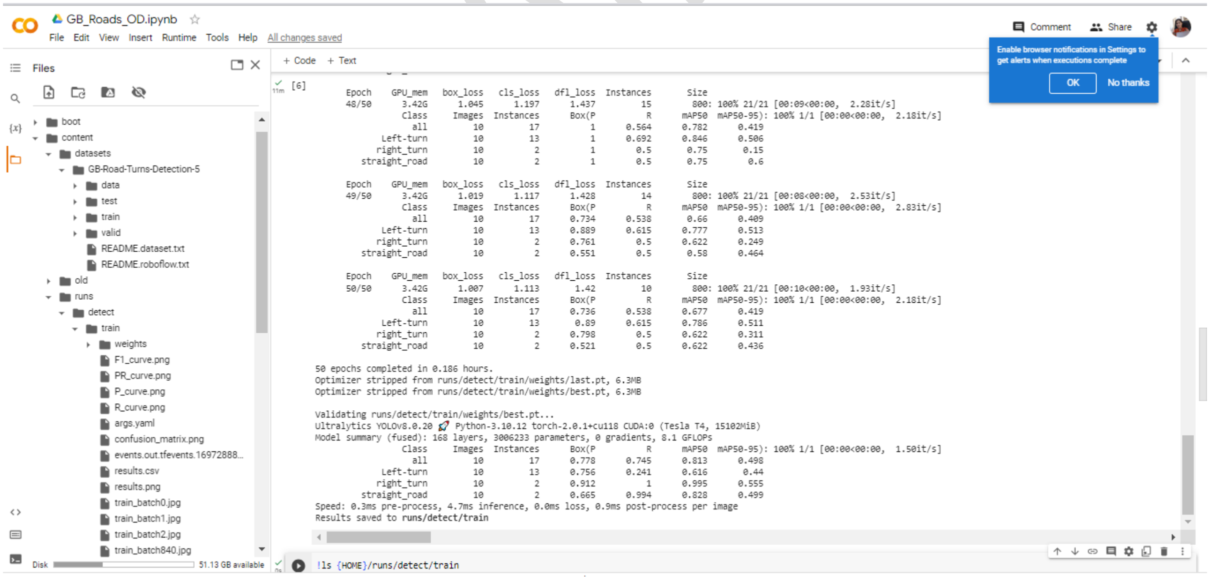
YOLOv8 offers several advantages for our project:

- Real-time Detection: YOLOv8 can perform real-time object detection, making it suitable for applications where quick decision-making is crucial, such as autonomous driving.
- High Accuracy: YOLOv8 leverages advanced deep learning techniques, resulting in high accuracy in object detection, even in complex scenes and varying lighting conditions.
- Ease of Integration: YOLOv8 is widely adopted, with a strong community, ensuring that I have access to extensive resources and support.

Model Training and Results

The heart of this project lies in training the YOLOv8 model using the annotated dataset of road condition images in Roboflow. The training process involves fine-tuning the model's parameters to accurately classify and locate road turns (right turns, left turns, and straight roads) in new images.

Training Results

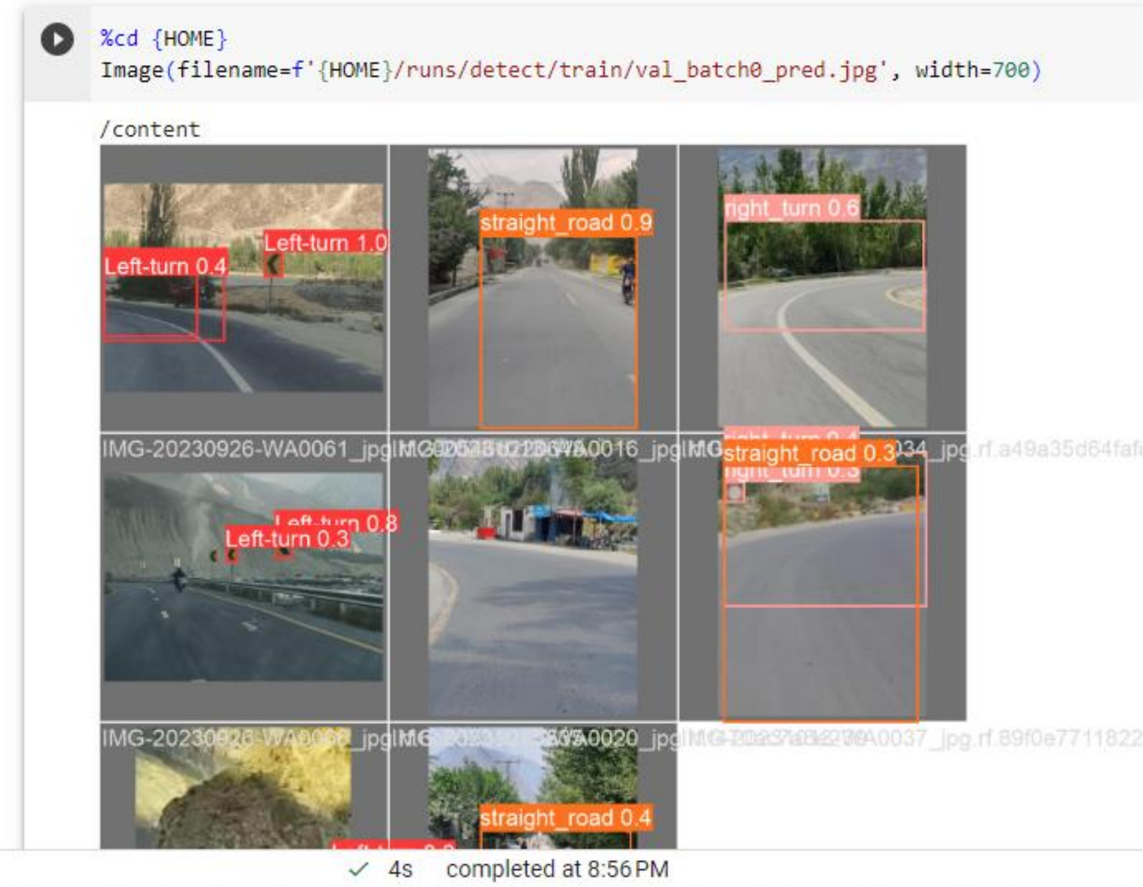


Enable browser notifications in Settings to get alerts when executions complete

[6]

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      48/50      3.42G      1.045      1.197      1.437         15        800: 100% 21/21 [00:09<00:00,  2.28it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00,  2.18it/s]
                   all         10         17          1      0.564      0.782      0.419
             Left-turn         10         13          1      0.692      0.846      0.506
            right_turn         10          2          1        0.5       0.75       0.15
         straight_road         10          2          1        0.5       0.75        0.6

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      49/50      3.42G      1.019      1.117      1.428         14        800: 100% 21/21 [00:08<00:00,  2.53it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00,  2.83it/s]
                   all         10         17      0.734      0.538       0.66      0.409
             Left-turn         10         13      0.889      0.615      0.777      0.513
            right_turn         10          2      0.761        0.5      0.622      0.249
         straight_road         10          2      0.551        0.5       0.58      0.464

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      50/50      3.42G      1.007      1.113       1.42         10        800: 100% 21/21 [00:10<00:00,  1.93it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00,  2.18it/s]
                   all         10         17      0.736      0.538      0.677      0.419
             Left-turn         10         13       0.89      0.615      0.786      0.511
            right_turn         10          2      0.798        0.5      0.622      0.311
         straight_road         10          2      0.521        0.5      0.622      0.436

50 epochs completed in 0.186 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.3MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.3MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3006233 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00,  1.50it/s]
                   all         10         17      0.778      0.745      0.813      0.498
             Left-turn         10         13      0.756      0.241      0.616       0.44
            right_turn         10          2      0.912          1      0.995      0.555
         straight_road         10          2      0.665      0.994      0.828      0.499
Speed: 0.3ms pre-process, 4.7ms inference, 0.0ms loss, 0.9ms post-process per image
Results saved to runs/detect/train
```

```
!ls {HOME}/runs/detect/train
```



```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/val_batch0_pred.jpg', width=700)
```

```
/content
```

The success of this project heavily depends on the training phase, which fine-tunes the model's ability to recognize road conditions and turns in Gilgit accurately. The training results provide insights into the model's performance and help gauge its readiness for real-world deployment.

Conclusion

I've written code to fetch a labeled dataset from Roboflow using the roboflow package. This step is crucial for obtaining the annotated dataset necessary for training the YOLOv8 model. The output in the notebook indicates that the requirements for this task are satisfied and provides information on the installed versions of the required packages.

In conclusion, I've provided an overview of the project, including its objectives and the chosen methodology. I've highlighted the importance of GPU access, the installation of YOLOv8, and the fetching of labeled datasets from Roboflow. While this report provides insights into the technical aspects of the project, please note that there may be additional content beyond what has been discussed here in the notebook, and a complete understanding of the project would require a more comprehensive review of the notebook's content.