Traffic and road sign detection using YOLOv8 and RT-DETR

By Shefali Shrivastava

Background and problem statement

Object detection represents a crucial domain within computer vision and machine learning, dedicated to discerning instances of objects within static images, videos, and camera feeds. Over the past decade, this field has witnessed substantial transformations, largely attributed to the integration of advanced deep learning models.

Within this field, the identification and interpretation of traffic and road signs constitute a critical application. Accurate and real-time detection of these signs is pivotal for enhancing road safety and facilitating intelligent transportation systems. This report delves into the implementation of two cutting-edge models, YOLOv8 and RT-DETR, to tackle the challenges associated with traffic and road sign detection. YOLOv8, renowned for its speed and efficiency, has set a benchmark in object detection tasks, while RT-DETR introduces a novel approach that claims superior performance. The report aims to give an initial comparison of the model models on a real-life dataset with low-resolution images – commonly used in training for self-autonomous driving vehicles.

This report touches upon multiple topics learned in COMS 4995, including but not limited to Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and the practical application of machine learning in production.

Dataset

The dataset used for this report is 'Traffic and Road Signs', an openly available dataset on Roboflow. The dataset has 10,000 images and annotations across 29 classes, with an average image ratio of 416x416. The train, validation, and test sets have 7092, 1884 and 1024 images respectively. No additional pre-processing was performed on the dataset. The dataset had slight imbalance, with 9 classes being underrepresented and 1 classes being overrepresented (check Appendix for visualization).

Figure 1: A few random samples from the 'Traffic and Road Sign Detection' dataset on Roboflow

Overview of the models

YOLOv8

YOLOv8 ('You Only Live Once', version 8), represents the latest iteration in the series of convolutional neural network (CNN)-based real-time object detection algorithms (Ultralytics YOLOv8, 2023). Originally proposed by Joseph et al. in 2015, this model has gained prominence for its exceptional speed, boasting an impressive frames per second (FPS) rate of 155 at a mean average precision (mAP) of 52.7% in its non-enhanced version. Fundamentally, YOLO model treats object detection as a single regression problem, employing layers of deep convolutional neural networks to predict bounding boxes and probabilities for each region simultaneously. An unofficial visual representation of YOLOv8's architecture is presented in the Appendix (GitHub, 2023).

RT-DETR

Real-Time Detection Transformer (RT-DETR) was recently proposed by Lv et al. (2023) as a transformer-based model to detect objects in real time. Trained on Microsoft's COCO 2017 dataset, the model boasts of surpassing YOLOv8 model in terms of speed and accuracy (as well as other famous detection models). The authors presented a detailed analysis of Non-Maximum Suppression (NMS) and concluded that it significantly contributes to delay in inference by real-time detectors. They introduced a hybrid encoder, improving the efficiency in processing of multi-scale features by segregating interactions within each scale from the fusion of information across different scales. Additionally, the model incorporates 'IoU-aware' object query selection, which prioritizes queries with higher relevance to the ground truth (constrains the model to produce high classification scores for features with high IoU scores and vice versa).

Model performance

The models were trained on the entire train set (7092 images) and tested on the test set (1024 images). The model training, prediction and inference was executed on Ultralytics YOLOv8 (version 8.0.225) environment, featuring Python 3.10.12, Torch 2.1.0 with CUDA support (CUDA:0 on Tesla T4 with 15102MiB), encompassing 2 CPUs, 12.7 GB RAM, and 27.1/166.8 GB disk space. The training process has the following limitations:

- Hyperparameter tuning and cross-validation were constrained by limited space and resources, influencing the extent of model optimization. Consequently, an alternative approach involved experimenting with 'best' hyperparameters gleaned from analogous open-source datasets, models, and relevant literature.
- RT-DETR model is unavailable for training on custom datasets; instead, a pre-trained model (on COCO17 dataset) was used. Due to differences in classes of COCO17 and current dataset, inference power of the model is limited.

'Best' hyperparameters	YOLOv8	RT-DETR
Epochs	50	20
Batch size	16	16
Image size	416	416
Optimizer	Auto	Auto
Learning rate	Auto	Auto
Dropout regularization	0.15	0.15

The results from training process are presented below:

Figure 2: Comparison of training metrics for YOLOv8 and RT-DETR

As expected, RT-DETR model did not train well. A pre-trained model was used, which probably contributed to wild fluctuations in mean average precision at each epoch. Based on the loss plot for RT-DETR, it seems that the model needs to be tuned at higher epochs (the original authors of the paper trained the model for 71 epochs for COCO17 dataset).

Evaluation and conclusion

The YOLOv8 model's mAP50 was 0.288 (all classes) and mAP50-95 was 0.239. The model completed 50 epochs in 1.432 hours. On the other hand, RT-DETR model had a mean average precision (aAP50) is 0.291 for all classes, and mAP50-95 of 0.245. The model completed 20 epochs in 2.52 hours. Class-wise predictions on test dataset showed that both models performed poorly on 2 out of 3 classes in the test set (check Appendix for details). Due to constraints in the training process, definitive claims regarding model's prediction accuracy on the test set are not possible at this time; however, it does seem that RT-DETR takes longer to train as compared to YOLOv8. As for next steps, the focus would be on refining the training process and considering dataset augmentation, especially given the insufficient classes in the test set for accurate model assessment.

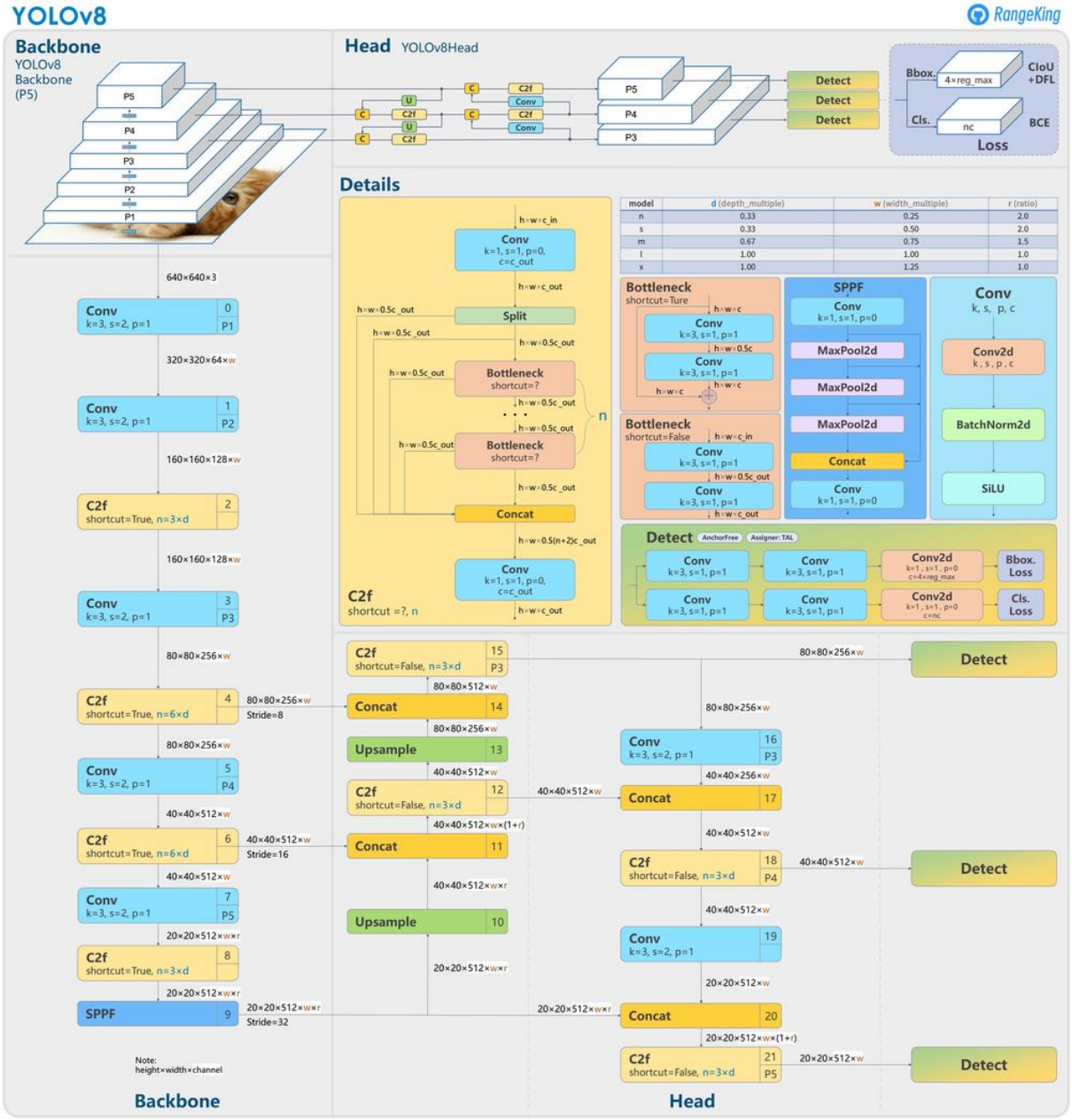
Sources

Lv, W., Xu, S., Zhao, Y., Wang, G., Wei, J., Cui, C., ... & Liu, Y. (2023). Detrs beat yolos on real-time object detection. *arXiv preprint arXiv:2304.08069*.

Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 (Version 8.0.0) [Software]. https://github.com/ultralytics/ultralytics

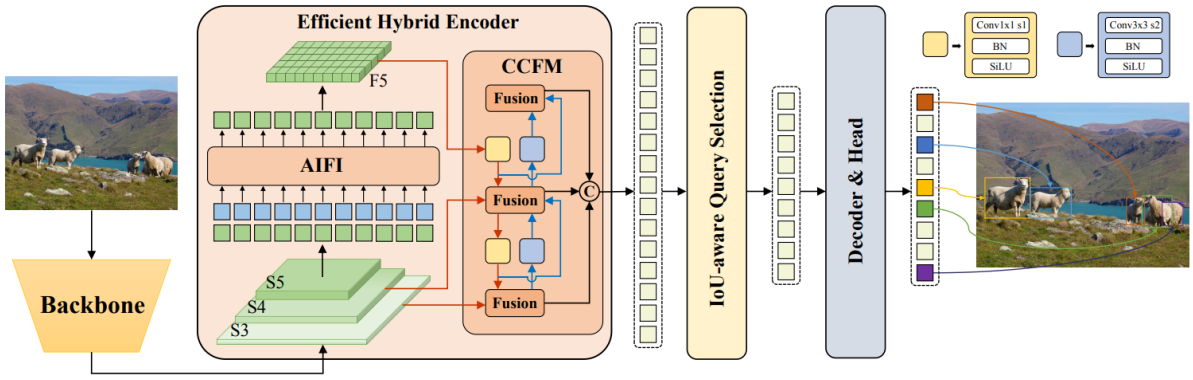
Appendix

1. Architecture of YOLOv8



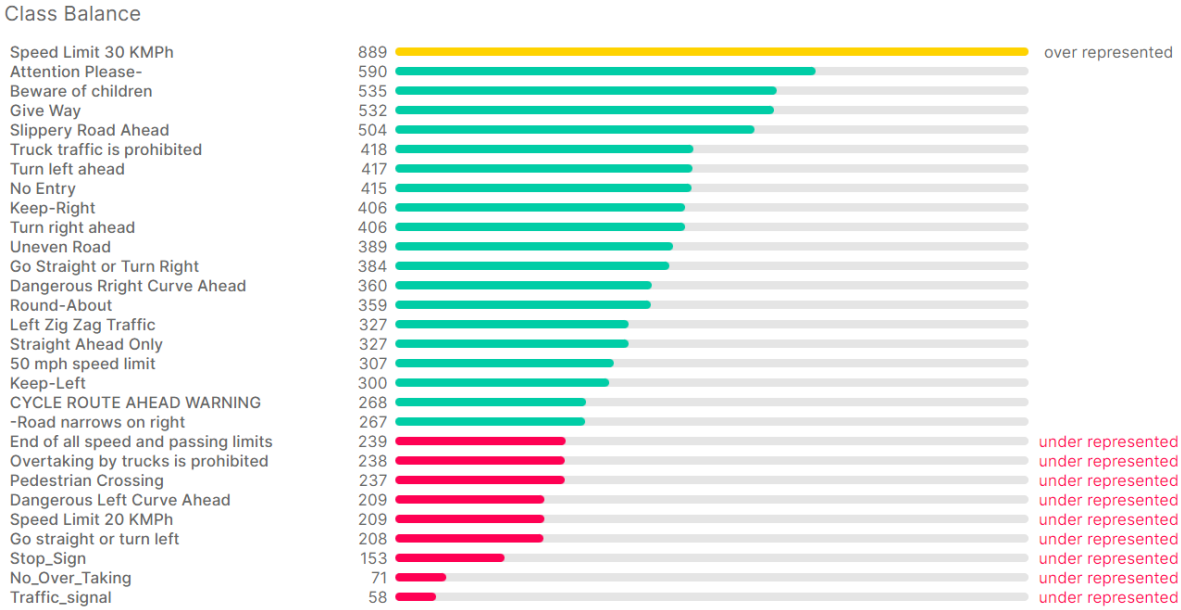
Source: https://github.com/ultralytics/ultralytics/issues/189

2. Architecture of RT-DETR



Source: Lv, W., Xu, S., Zhao, Y., Wang, G., Wei, J., Cui, C., ... & Liu, Y. (2023). Detrs beat yolos on real-time object detection. *arXiv preprint arXiv:2304.08069*.

3. Class balance in the 'Traffic and road sign detection' dataset on Roboflow

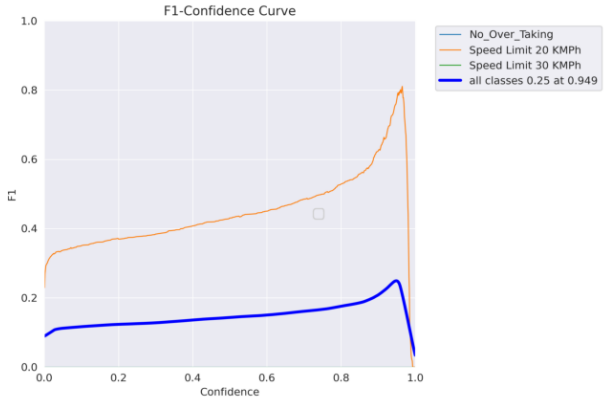


Source: Roboflow. (2022, November). Traffic and Road Signs Dataset [Open Source Dataset]. Retrieved from https://universe.roboflow.com/usmanchaudhry622-gmail-com/traffic-and-road-signs

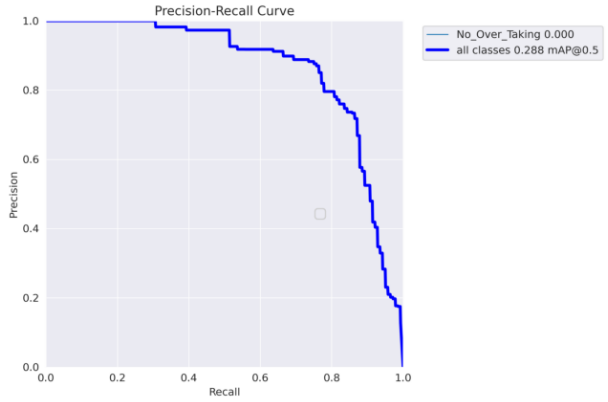
4. Evaluation metrics for YOLOv8 and RT-DETR – the models performed poorly on test classes due to incomplete training process (constrained by space and resource constraint).

Comparison of evaluation metrics (F1 and PR Curves) for YOLOv8 and RT-DETR

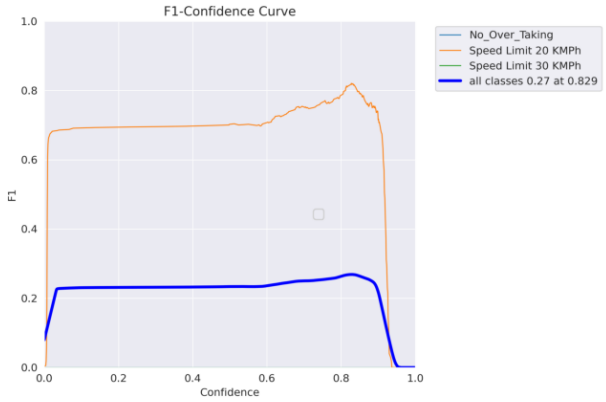
F1 Curve - YOLOv8



PR Curve - YOLOv8



F1 Curve - RT-DETR



PR Curve - RT-DETR