

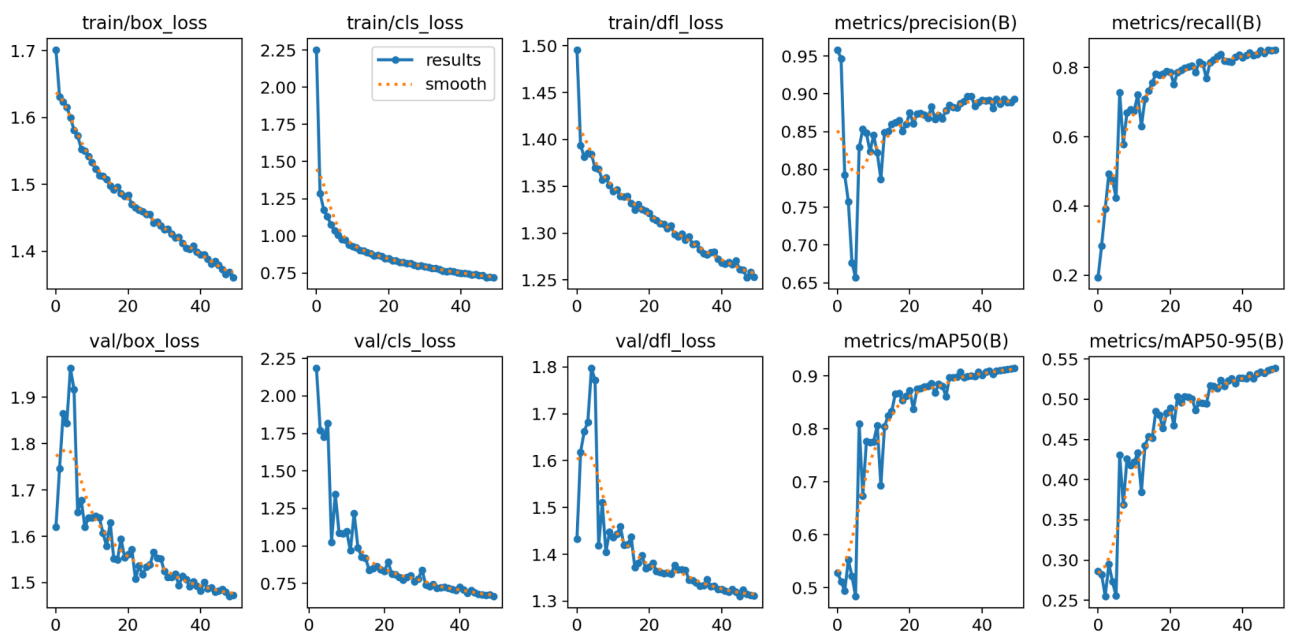
Summary (Team-Detectron)

- ❖ An **85:15 ratio** used to divide the supplied dataset into **training and validation sets**.
- ❖ The data were evaluated using a number of YOLOv8 model variants, including YOLOv8s, YOLOv8m, YOLOv8n.

Model	mAP50	Inference Time*	Parameters
YOLOv8n	0.915	34.17 sec	3.2 M
YOLOv8s	0.922	42.94 sec	11.2 M
YOLOv8m	0.922	46.78 sec	25.9 M

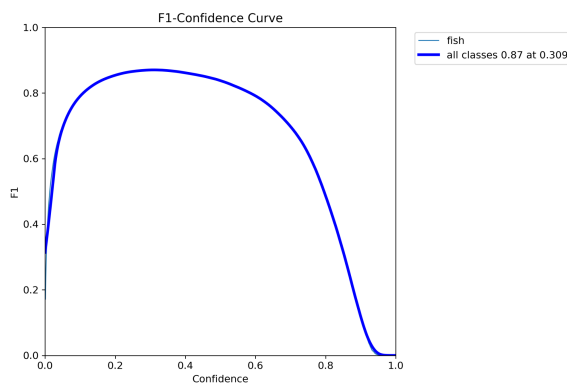
*Total Inference Time calculated on given 1100 Test Images (Tesla T4 GPU) with TensorRT fmt. weights

- ❖ **YOLOv8n** triumphs because of the extremely short inference time and nearly identical MAP.
- ❖ **Batch size of 128** and over **50 epochs** was used to train the model, which produced the best results with our data. Below are the metrics of our trained YOLOv8n

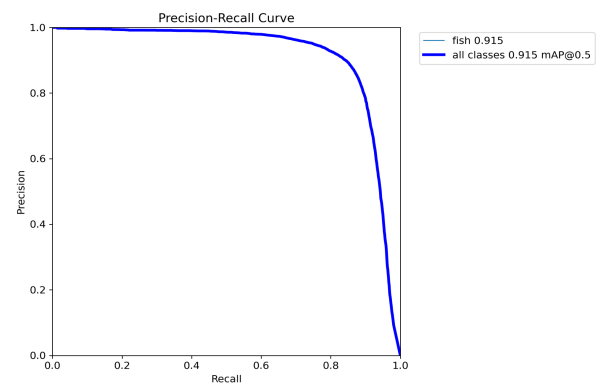


- ❖ Here are some graphs of our model.

F1-Confidence Curve



Precision-Recall Curve



- We downloaded the model weights in a variety of file formats, including.pt,.onnx, and.engine. In particular, the **TensorRT(engine) format** weights showed exceptionally fast inference times, demonstrating their effectiveness.

Model weights	Inference_Time
Yolov8n.pt (Colab Tesla T4)	39.42 sec
Yolov8n.onnx (Colab Tesla T4)	39.79 sec
Yolov8n.engine (Colab Tesla T4)	34.17 sec

- ❖ **yolov8n.engine** is the winner due to its **faster inference time** than others.
- ❖ We developed a **video** of the test data and tested it using our model.
- ❖ We offer both batch images with the original labels and batch images with model predictions.



Batch with **original** label



Batch with **predicted** label

References:

- <https://docs.ultralytics.com/>
- <https://github.com/ultralytics/ultralytics>