

Solution to DEPONDFI'23 Challenge

Presentation by DMACS_SAI, SSSIHL July 21, 2023





Train Data Distribution

- Color
- Background
- Size of fish
- Type of fish



Train Data Distribution

- Color
- Background
- Size of fish
- Type of fish



Train Data Distribution

- Color
- Background
- Size of fish
- Type of fish



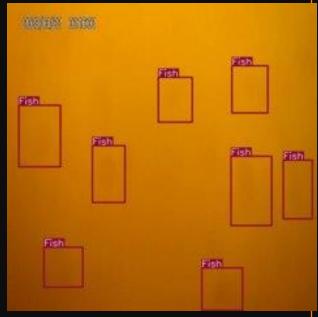


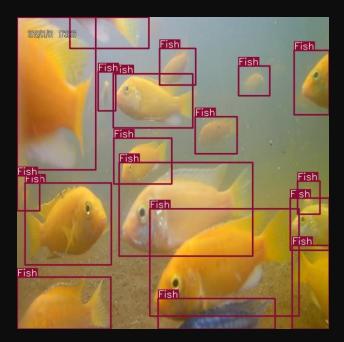
Train-test distribution shift

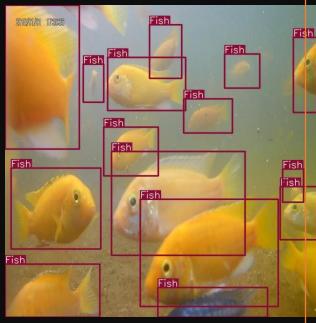
- Background
- Fish orientation
- Color
- Type of fish

Inconsistent Ground-truths









Inconsistent Ground-truths

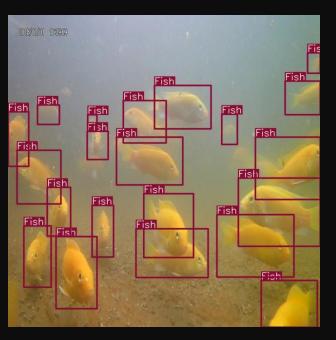




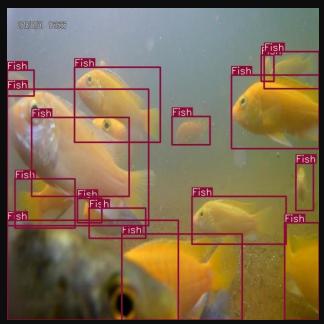


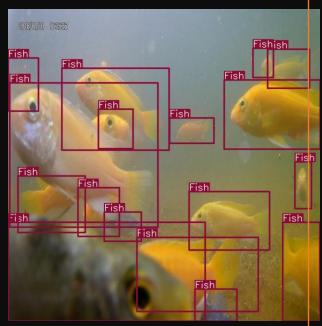


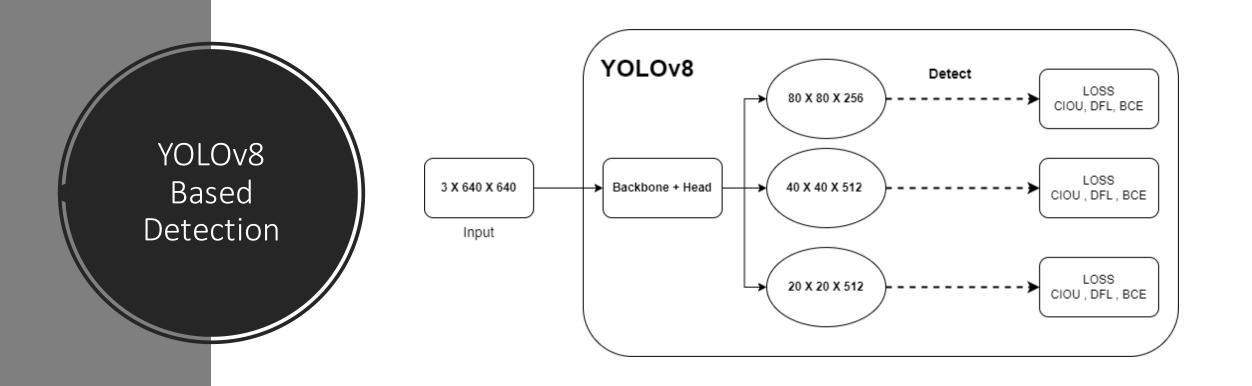
Inconsistent Ground-truths

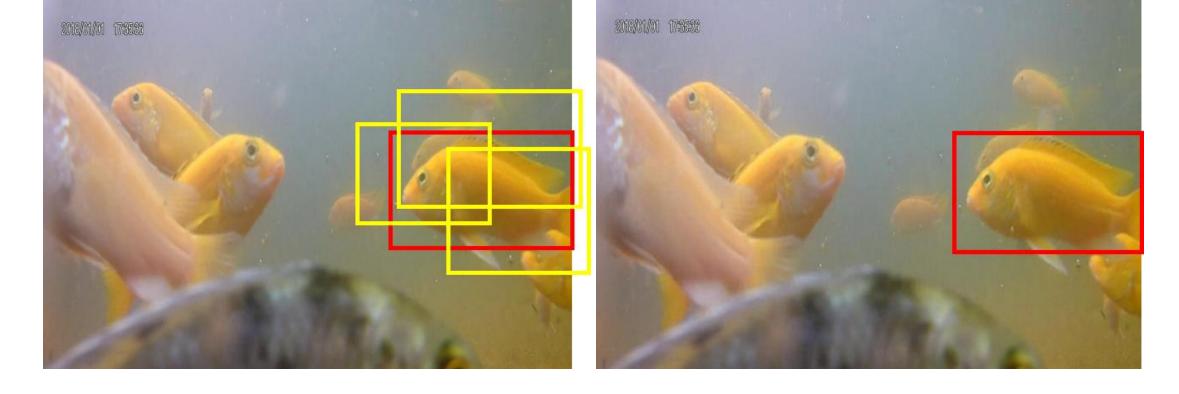












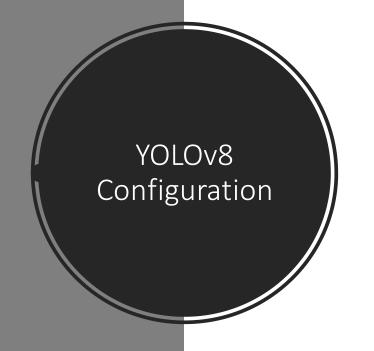
Soft Non-Max Suppression

- Significant overlap of fishes standard NMS likely to miss out some detections
- Soft-NMS modulates original confidence by weight inversely proportional to keeper and the bounding box
- Soft-NMS performed better than standard NMS in our experiments

Hyperparameter Tuning

- On average, only 16 Bboxes/image in DePondFi dataset
- A lower Bbox confidence C_t should do well in this case
- To avoid colliding
 Bboxes of different fishes a lower
 loU threshold I_t is better
- Performed grid search to determine best C_t & I_t





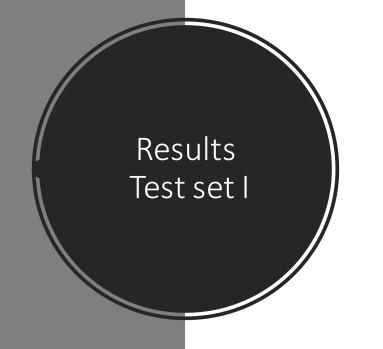
Parameter	Details
Train Count	5706
Validation Count	152
Augmentation	Random flipping, color jittering, mosaic augmentation
Optimizer	SGD
Learning Rate	0.01 initial and linearly decayed
Model	YOLOv8 short
Batch size	16
СТ	0.25
IT	0.6



Model name	mAP@0.5	mAP@0.5-0.95	Inference time (GPU/CPU in ms)
YOLOv8-nano	0.939	0.545	10.8/58.9
YOLOv8-short	0.937	0.547	11.1/163.6
YOLOv8-medium	0.932	0.559	12.9/361.9
YOLOv8-large*	0.935	0.546	26.8/659.3
YOLOv8-XLarge**	0.932	0.56	55.4/1005.4

^{*} Batch size is 24

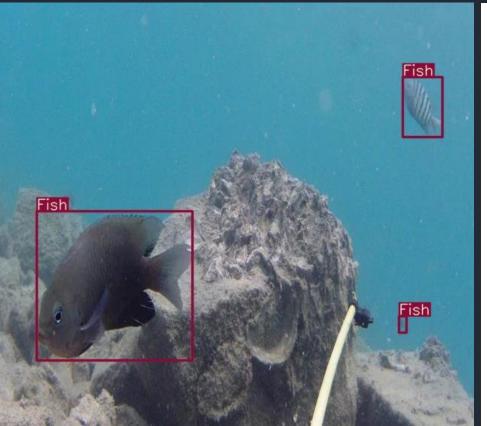
^{**} Batch size is 12



Model name	mAP@0.5-0.95
YOLOv8-nano	0.426
YOLOv8-short	0.447
YOLOv8-medium	0.408
YOLOv8-XLarge	0.413

Glimpse into other attempts
Expanding train set





- DeepFish dataset
- Around 3800 images added to DePondFi train set after 640X640 resize
- Poor performance probably because of significant shift in data distribution







Glimpse into other attempts – Underwater image enhancement

- <u>U-shape transformer for underwater</u> <u>image enhancement</u>
- Poor performance probably because of incompatibility between revealed information and ground truth
- Artifacts introduced by enhancement could also be a reason for poor performance





Thank you

