

K-Water 어종 탐지 프로젝트

- 김 현 승 -

● Contents

데이터 설명

데이터 전처리

모델링

결론




Part 1

데이터 설명

● 데이터 설명

진행중



[2023 제3회 K-water AI 경진대회] 어종(魚種) 식별 및
분류 알고리즘 개발

📅 2023-10-30 - 2023-11-30

👤 420

🏆 1,000만원

♡

🔗


참여중 ⓘ


제출하기


진행내용	일정
참가접수 및 팀빌딩	10월 18일(수) ~ 11월 8일(수) 17:00
대회 기간	10월 30일(월) ~ 11월 13일(월) 17:00
검증 기간(입상후보자 대상)	11월 14일(화) ~ 11월 20일(월)
결과 발표	11월 20일(월) 15:00
시상식(오프라인)	11월 하순경 @대전

● 데이터 설명

dataset.zip

|—  **train** : 학습 데이터 이미지 ➡ 약 100,000개

|—  **test** : 평가 데이터 이미지 ➡ 약 45,000개

|—  **labels** : coco format json

|—  **train.json** : 학습 데이터에 대한 coco format label

|—  **answer_sample.json** : 제출용 coco format label

● 데이터 설명

dataset.zip

└─ **train** : 학습 데이터 이미지 ➡ 약 100,000개

└─ **test** : 평가 데이터 이미지 ➡ 약 45,000개

└─ **labels** : coco format json

└─ **train.json** : 학습 데이터에 대한 coco format label ➡ 약 6,000개 annotations

└─ **answer_sample.json** : 제출용 coco format label

```
"images": [  
  {  
    "id": 1,  
    "width": 640,  
    "height": 480,  
    "file_name": "train_0.png",  
    "license": 0,  
    "flickr_url": "",  
    "coco_url": "",  
    "date_captured": 0  
  },  
  ...  
]
```

```
"annotations": [  
  {  
    "id": 1,  
    "image_id": 1,  
    "category_id": 3,  
    "segmentation": [],  
    "area": 40939.4791,  
    "bbox": [  
      37.98,  
      209.94,  
      369.19,  
      110.89  
    ],  
    "iscrowd": 0,  
    "attributes": {  
      "occluded": false,  
      "rotation": 0.0  
    }  
  },  
  ...  
]
```

```
"categories": [  
  {  
    "id": 0,  
    "name": "농어",  
    "supercategory": ""  
  },  
  {  
    "id": 1,  
    "name": "베스",  
    "supercategory": ""  
  },  
  {  
    "id": 2,  
    "name": "송어",  
    "supercategory": ""  
  },  
  ...  
]
```

농어-0

베스-1

송어-2

강준치-3

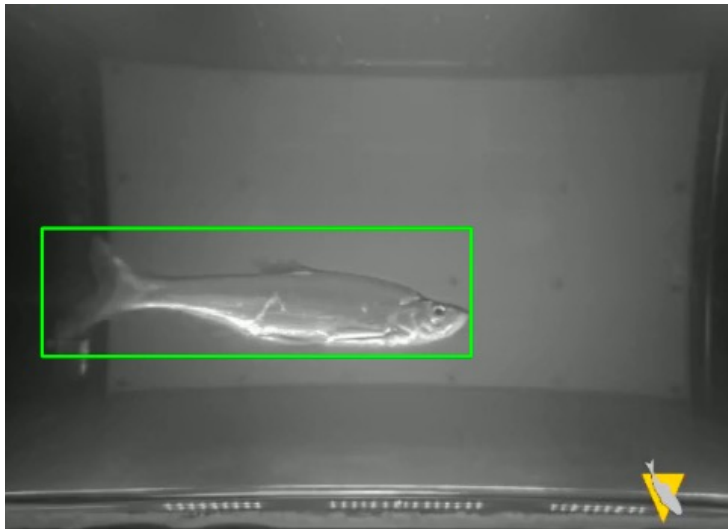
블루길-4

잉어-5

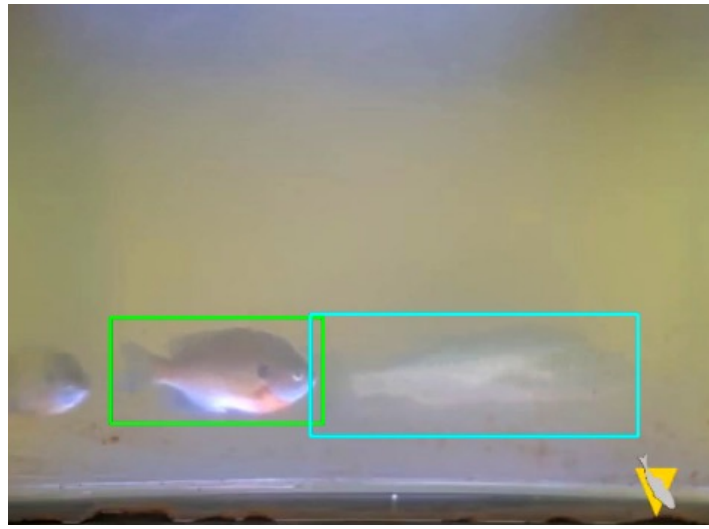
붕어-6

누치-7

● 데이터 설명



객체 1개



객체 여러개



객체 존재 X



Part 2

데이터 전처리

● 데이터 전처리

Train 100,000개 ➡ 객체 6,000개 ➡ Train ?
Val ?

Dataset	Train(객체/배경)	Val(객체/배경)	Class Distribution	Augment
Sample_v1	9,500개(4,500/5,000)	1,000개(500/500)	베스(37.1%), 강준치(34.1%), 송어(10.3%), 누치(9.3%), 농어(2.8%), 잉어(2.4%), 붕어(2.3%), 블루길(1.6%)	False
Sample_v2	18,000개(13,000/5,000)	2,000개(1,500/500)	강준치(35.1%), 베스(25.5%), 송어(17.7%), 누치(9.6%), 농어(4.8%), 붕어(3.2%), 블루길(2.2%), 잉어(1.6%)	True
Sample_v3	22,000개(17,000/5,000)	2,500개(2,000/500)	송어(28%), 강준치(27.7%), 베스(20.2%), 농어(7.6%), 누치(7.5%), 붕어(5.7%), 블루길(1.7%), 잉어(1.3%)	True
Sample_eq	26,000개(21,000/5,000)	2,800개(2,300/500)	강준치(23%), 송어(21.3%), 누치(19.3%), 베스(16.8%), 농어(6%), 잉어(5.2%), 붕어(4.9%), 블루길(3.3%)	True

● 데이터 증강



Origin



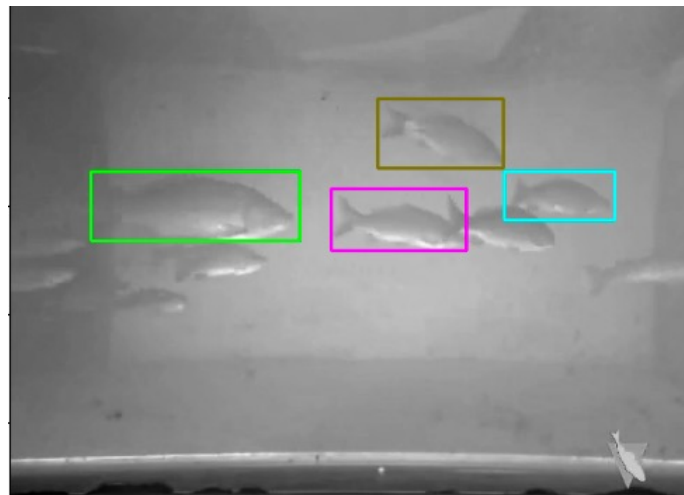
Blur



Brightness



Gaussian noise



Gray

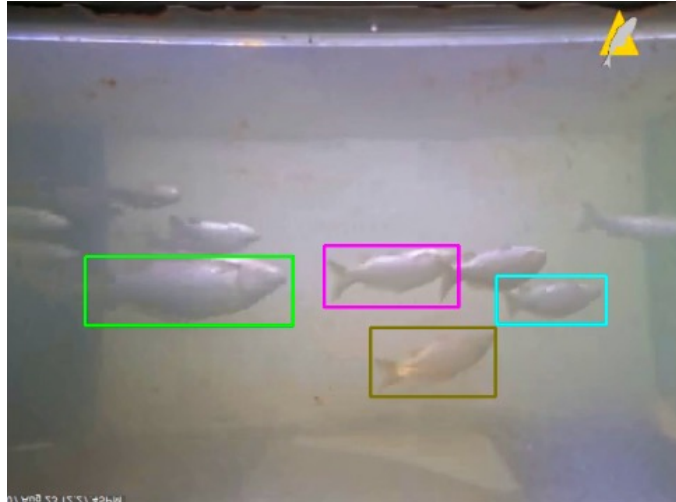


Clahe

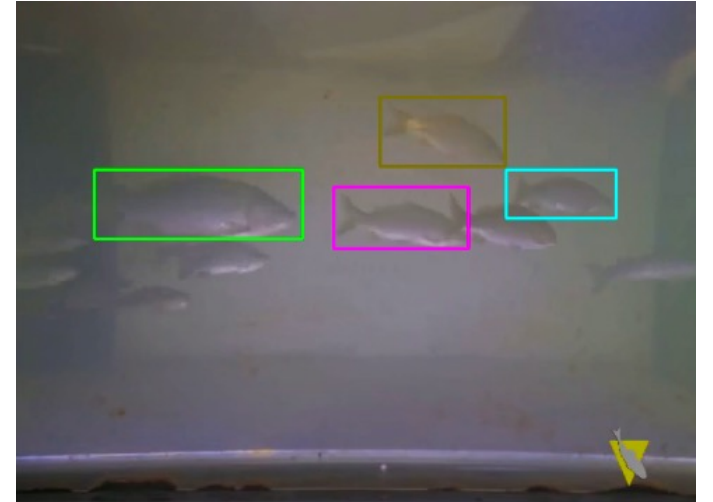
● 데이터 전처리



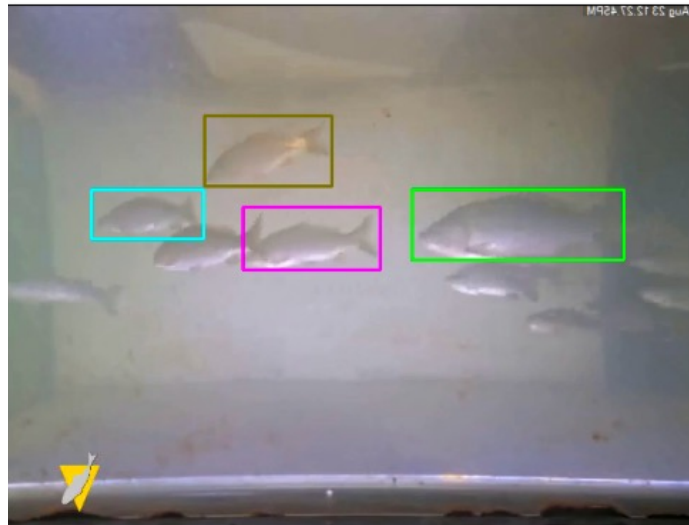
Origin



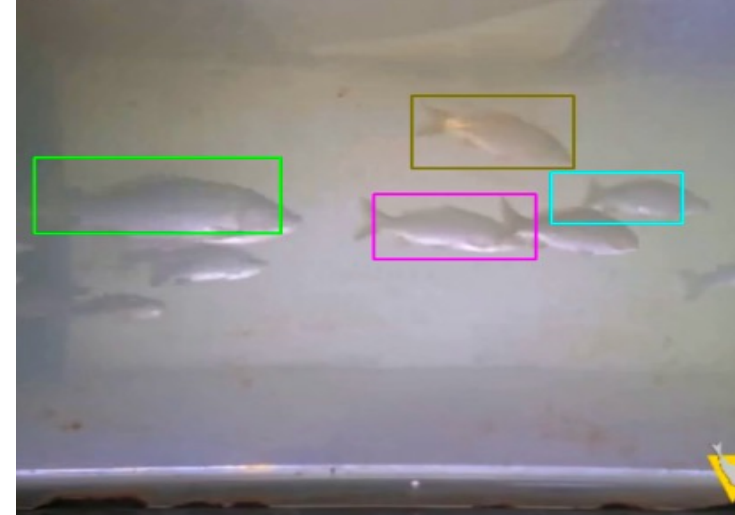
Vertical flip



HSV



Horizontal flip

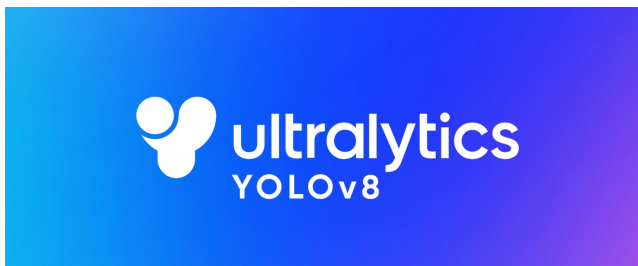


Resize and Crop

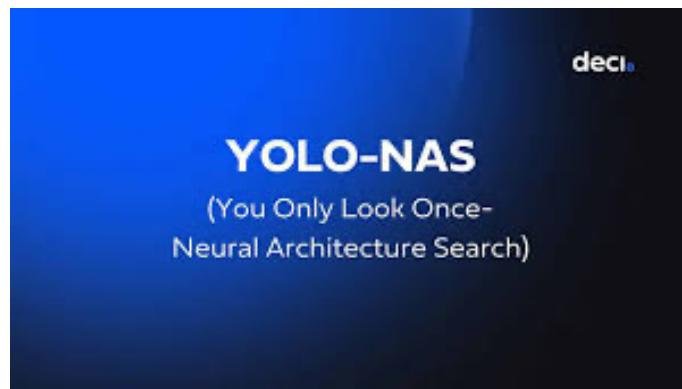


Part 3

모델링



Ultralytics latest model
트랜스포머 기반
학습이 편리하고 준수한 성능
n, s, m, l



Latest YOLO model
Quantization 적용
YOLOv8보다 준수한 성능
s, m, l



2 stage detector
Anchor Box 사용
R-CNN 준수한 성능
ResNetX-101-FPN

YOLOv8

$$\text{Precision}_{IoU} = \frac{\Sigma IoU_{TP}}{TP+FP}$$

$$\text{Recall}_{IoU} = \frac{\Sigma IoU_{TP}}{TP+FN}$$

$$\text{F1-Score}_{IoU} = \frac{2 \times \text{Precision}_{IoU} \times \text{Recall}_{IoU}}{\text{Precision}_{IoU} + \text{Recall}_{IoU}}$$

$$\text{MacroF1}_{IoU} = \frac{1}{N} \sum_{i=0}^N \text{F1-Score}_{IoU}$$

각각의 클래스에 대해 F1-score를 계산 후 평균

Model	Dataset	Hyperparameters	Val mAP	Test F1
YOLOv8n	Sample_v1	epochs=20, optimizer = 'auto'	0.79	0.529
YOLOv8n	Sample_v2	epochs=60, patience=30, lr0 = 0.0005, momentum = 0.85, weight_decay = 0.05, batch=64, warmup_epochs = 0, close_mosaic = 60, optimizer = 'AdamW'	0.88	0.635
YOLOv8n	Sample_v3	epochs=100, patience=20, batch=64, mosaic = 0, optimizer = 'AdamW'	0.87	0.616
YOLOv8n	Sample_eq	epochs=120, patience=20, batch=64, optimizer = 'SGD', lr0 = 0.001, mosaic = 0, exist_ok = True, save_period = 10	0.87	0.622

● YOLO-NAS & Faster R-CNN

Model	Dataset	Hyperparameters	Val mAP@50	Test F1
YOLO-NAS-S	Sample_v2	"warmup_mode": "linear_epoch_step", "warmup_initial_lr": 1e-6, "lr_warmup_epochs": 3, "initial_lr": 5e-4, "lr_mode": "cosine", "cosine_final_lr_ratio": 0.1, "optimizer": "AdamW", "optimizer_params": {"weight_decay": 0.001}, MAX_EPOCHS = 40	0.87	0.567
YOLO-NAS-M	Sample_v2	"warmup_mode": "linear_epoch_step", "warmup_initial_lr": 1e-6, "lr_warmup_epochs": 3, "initial_lr": 5e-4, "lr_mode": "cosine", "cosine_final_lr_ratio": 0.1, "optimizer": "Adam", "optimizer_params": {"weight_decay": 0.0001}, MAX_EPOCHS = 20	0.83	0.42
YOLO-NAS-L	Sample_v2	"warmup_mode": "linear_epoch_step", "warmup_initial_lr": 1e-6, "lr_warmup_epochs": 3, "initial_lr": 5e-4, "lr_mode": "cosine", "cosine_final_lr_ratio": 0.1, "optimizer": "AdamW", "optimizer_params": {"weight_decay": 0.001}, MAX_EPOCHS = 35	0.78	0.529
Faster R-CNN	Sample_eq	cfg.optimizer.type = 'SGD' cfg.optimizer.lr = 0.001 cfg.optimizer.momentum = 0.93 cfg.optimizer.weight_decay = 0.0005 cfg.runner.max_epochs = 25	0.74	0.507

● Weighted Boxes Fusion

YOLOv8n_e60_best YOLOv8n_e40_best YOLOv8n_e100_best YOLOv8n_e120_sgd_best

Sample_v2

Sample_v2

Sample_v3

Sample_eq

Test 0.635

Test 0.625

Test 0.616

Test 0.622



WBF weights = [2, 2, 1, 1], iou_thr = 0.6, skip_box_thr = 0.81

Final Test F1-Score 0.637

Part 4

결론



결론

- 1) YOLO 외에 더 다양한 모델들을 돌려보지 못했고 데이터 증강과 YOLOv8-nano 모델의 하이퍼파라미터 튜닝에 의존
 - ▶ 다른 아키텍처와 알고리즘을 가진 모델들을 다양하게 돌려보는 것
- 2) 데이터가 충분하지 않아서 데이터 증강을 적용했지만 여전히 부족, 상대적으로 가벼운 모델의 성능이 더 좋았음
 - ▶ 학습 할 데이터가 충분하지 않을때 어떻게 성능을 높여야하는지에 대한 경험과 지식이 필요
- 3) 모델 구축을 완료했는데도 GPU 메모리의 한계로 여러 모델들을 학습시키지 못함
 - ▶ 현재 환경에서 가용한 모델과 데이터셋의 크기가 어느 정도인지를 빠르게 파악하여 시간 낭비를 최소화

● 개발환경



colab

kaggle



PYTORCH

MMDetection



Thank You!