

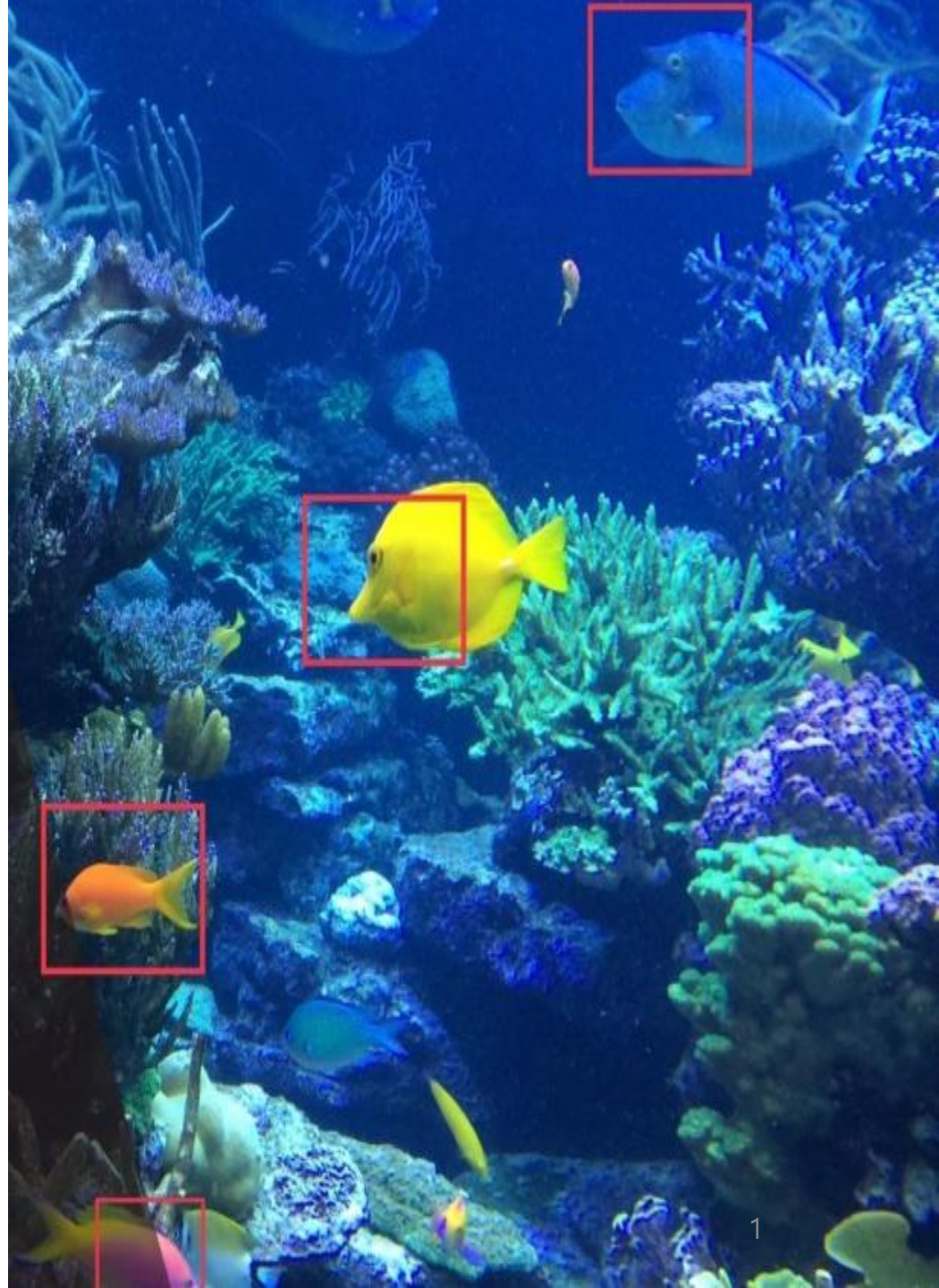


## [2023 제 3회 K-water AI 경진대회]

어종(魚種) 식별 및 분류 알고리즘 개발  
2023.10.30 ~ 2023.11.13

팀 : 조안나 맛있다

백근주, 김동영



# 목차

1. 대회 개요
2. 대회 문제 상황 (EDA)
3. 실험 세팅
4. 전처리 및 증강 기법
5. 모델
6. 수도라벨링
7. 실험 결과
8. 마무리

# 대회 개요

- 낙동강 하굿둑 물고기 영상에서 어종을 식별하고 분류하는 AI 모델 개발
- 평가 지표
  - Macro F1 Score

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

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

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

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

# 대회 문제 상황 (EDA)

- 데이터 분석결과

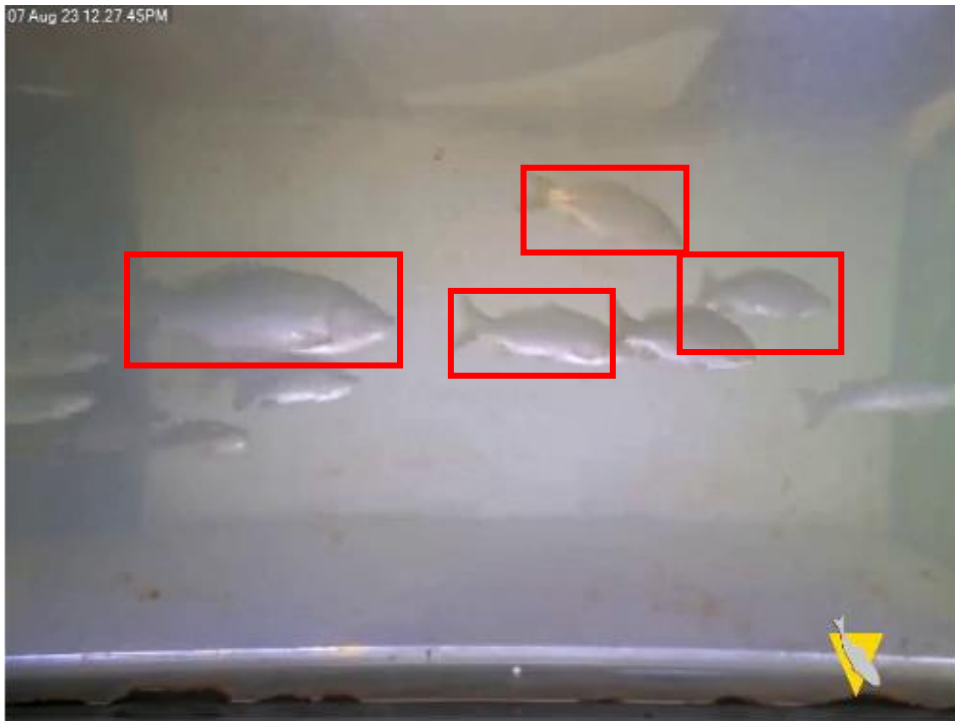
1. Background : Foreground = 95000 : 5000 으로 극단적임
  - '배경을 학습해야 하는가?' 에 대한 의문점 발생
  - 가설 검증을 위해 다양한 데이터셋 크기로 실험 진행  
(이후 슬라이드에서 설명)

# 대회 문제 상황 (EDA)

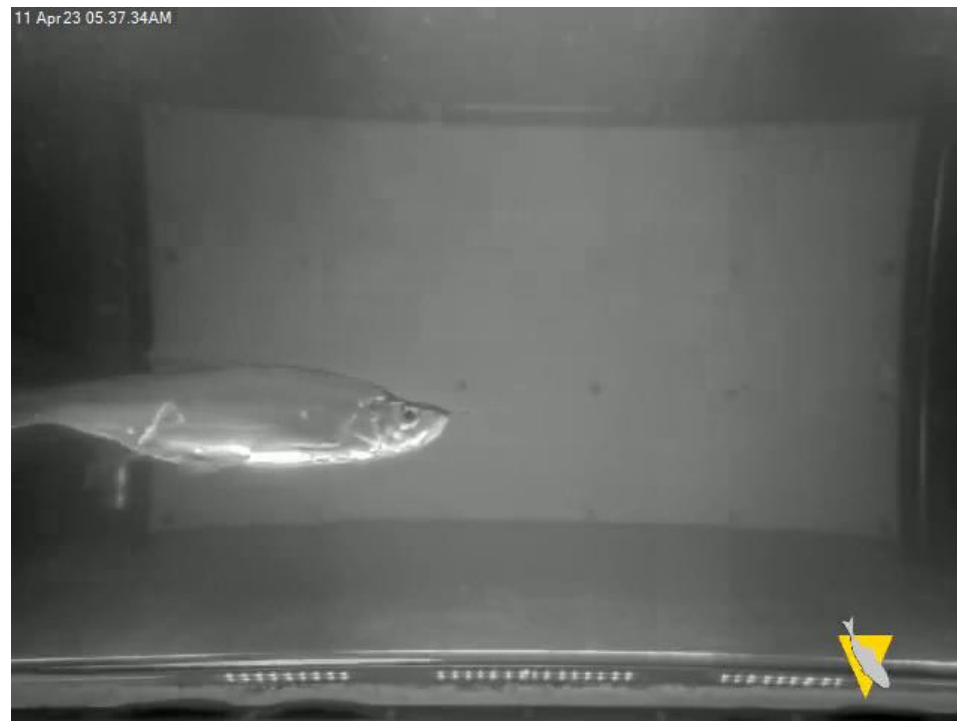
- 데이터 분석결과

2. 다양한 Noise가 존재함

→ 꼬리까지 완전히 보일 경우만 탐지 / 특정 종만 탐지 / 흐린 부분



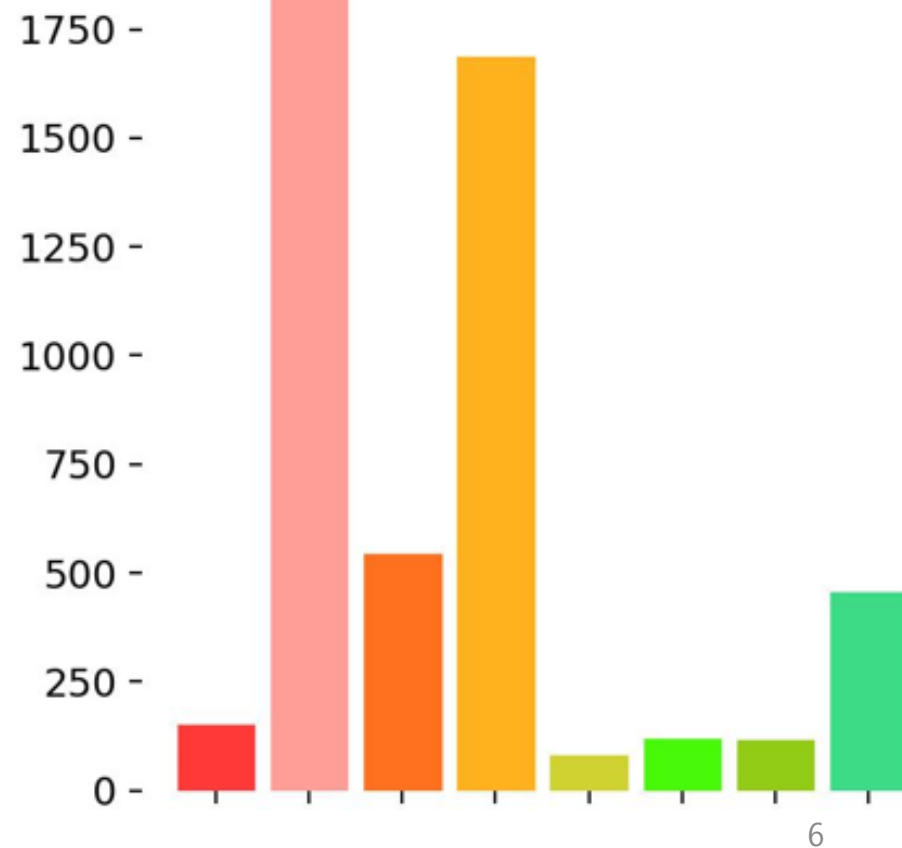
일부만 검출해야 하는 경우



검출하지 말아야 하는 경우

# 대회 문제 상황 (EDA)

- 데이터 분석결과
  3. Foreground class 간 Imbalance가 존재  
(극단적으로 20:1의 비율)



# 실험 세팅

- GPU
  - RTX 3090 24GB x 1EA
  - RTX A6000 48GB x 1EA
- 데이터를 점차 늘려가며 실험 진행
  1. Foreground만 사용
  2. Foreground : Background = 1:1이 되도록 Background 추가
  3. 2번 상황에서 참고 데이터셋으로 받은 동영상 pseudo labeling
  4. Foreground + Background 전체 사용
  5. 4번 상황에서 참고 데이터셋으로 받은 동영상 pseudo labeling→ 1에서 5로 갈수록 성능이 점진적으로 향상되는 것 확인

# 전처리 및 증강 기법

- Flip, Mosaic, Custom cutout, mixup, clahe를 이용함
- Flip : 수중 카메라 환경에서 촬영된 영상이므로 좌우 반전 시 다양성 늘릴 수 있음
- Mosaic : 객체의 수가 상당히 적기 때문에 mosaic으로 객체 수를 늘리는 방법 적용
- Mixup : 이미지 간 분류 능력 향상을 위해 적용함
- Clahe : 이미지가 흐린 경우가 존재하여 histogram equalization 기법 중 하나인 clahe를 적용하여 선명하게 해줌

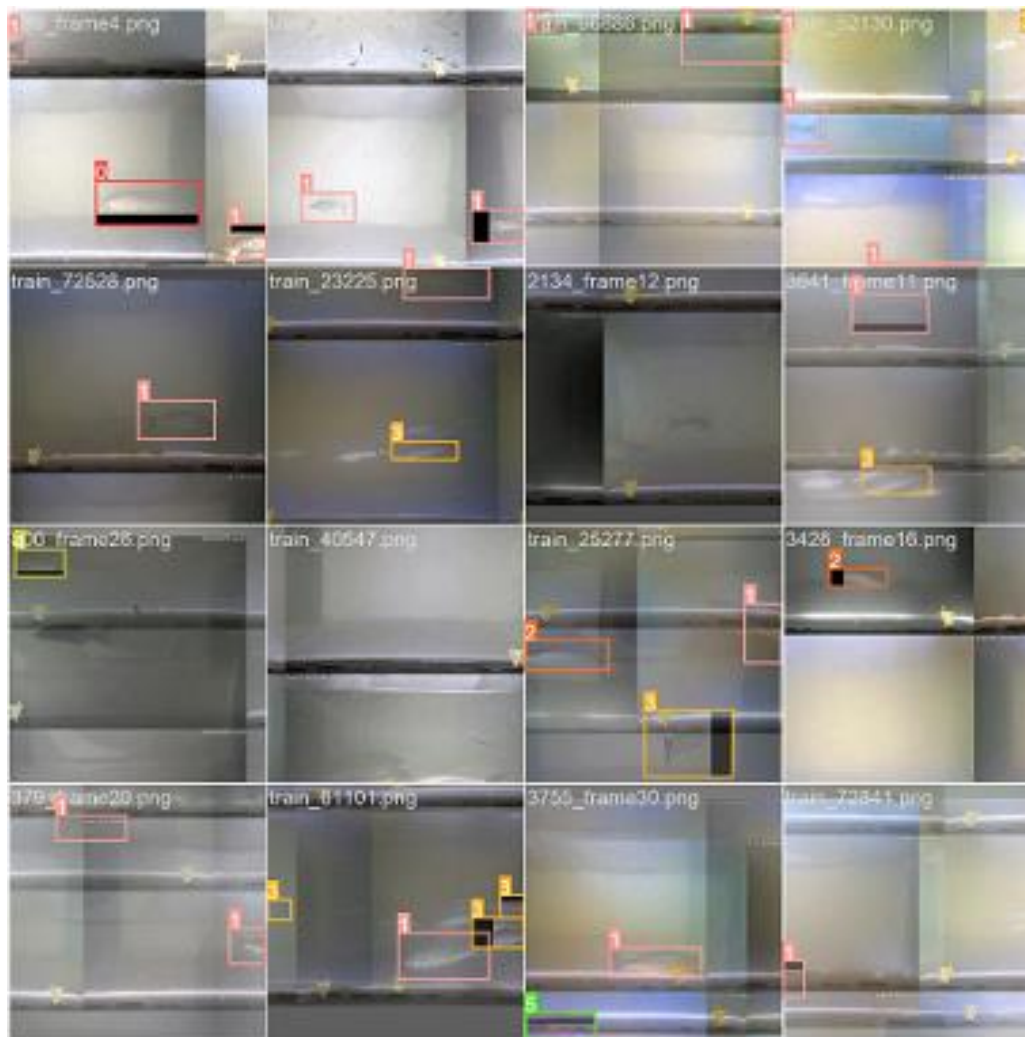


# 전처리 및 증강 기법

- Custom Cutout의 사용 배경
  - Albumentations의 Cutout을 이용시 객체 / 배경 간 구분을 힘들어하는 것을 확인함
    - bbox에 대응되는 영역에서만 cutout을 적용해주면 어떨까?
  - 따라서, 해당 기능이 있는 Custom Cutout을 적용
    - 객체 / 배경 간 구분 힘들어하는 것 개선
  - 특히, Mixup과 같이 썼을 경우 성능 향상이 두드러짐을 확인함

# 전처리 및 증강 기법

- 전처리 및 증강기법 적용된 Batch 이미지 시각화



# 모델 선정

- 모델로는 Cascade R-CNN / Co-DETR / YOLOv8 중 YOLOv8을 선택
  - Cascade R-CNN은 타 대회에서 나온 1등 솔루션으로, IoU threshold를 점점 높여가며 더 좋은 detection 결과가 나오도록 refinement를 진행
  - Co-DETR의 경우 현재 Coco dataset에서 SOTA 모델
  - YOLOv8의 경우, YOLOv5를 개선하여 추론속도 및 검출 정확성 향상
  - 최종적으로 YOLOv8이 노이즈 있는 상황에서도 객체 분류를 가장 잘하여 선정

# 모델 선정

- YOLOv8 중 m, l, x 모델 선정
  - 실험 결과 m, l, x 모두 비슷한 결과를 보여줌
    - m을 기준으로 학습 진행함
- Open Images V7으로 Pre-train 된 YOLOv8 m model 채택
  - Open Images 에는 Fish class가 있기 때문에, 처음부터 직접 학습시키는 것보다 더 좋은 효과를 보여줌
    - 실험결과 사전학습 모델 사용 유무가 성능에 3정도 영향을 미침
  - 처음 3개 layer에 대해 Freeze 적용
    - 모델의 학습 파라미터는 줄이면서 성능은 비슷함

# 모델 검증

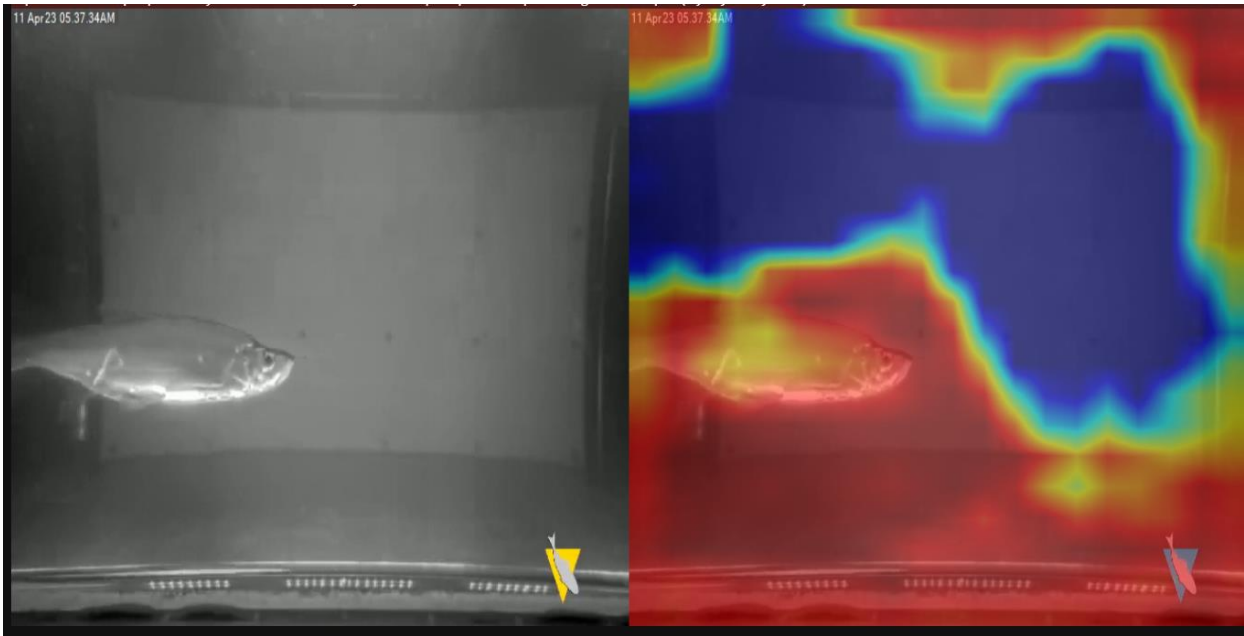
- 모델검증은 Train을 Train / Val 로 분할하여 Val 기준으로 성능을 평가함
  - 이 때, Val로 평가할 뿐만 아니라 Eigen CAM을 이용하여 모델이 객체를 탐지할 때, 어디 부분에 집중하는지 확인하는 절차 진행
    - 모델이 잘못 검출했을 경우, 해당 부분을 시각화로 확인하여 모델이 해당 객체로 검출한 이유 분석 및 개선과정 거침

# 모델 검증

- 모델검증 간 문제 상황 발생 및 해결
  - Validation F1-score 측정 성능과 Test F1-score 측정 성능 결과에 차이가 존재했음
    - 또한, Validation F1-score가 약간 낮아지는 상황에서 Test F1-score가 오르는 상황 존재
      - Validation 성능의 신뢰도가 떨어지는 문제가 발생함
      - 이에 따라, training model이 적은 데이터로 인해 과적합했다고 결론
  - Mixup + Custom Cuotut 적용 시 해당 문제를 해결할 수 있었음
    - Validation F1-score와 Test F1-score가 같이 오르는 것을 확인함

# 모델 검증

- CAM 결과 시각화



배경 학습 전



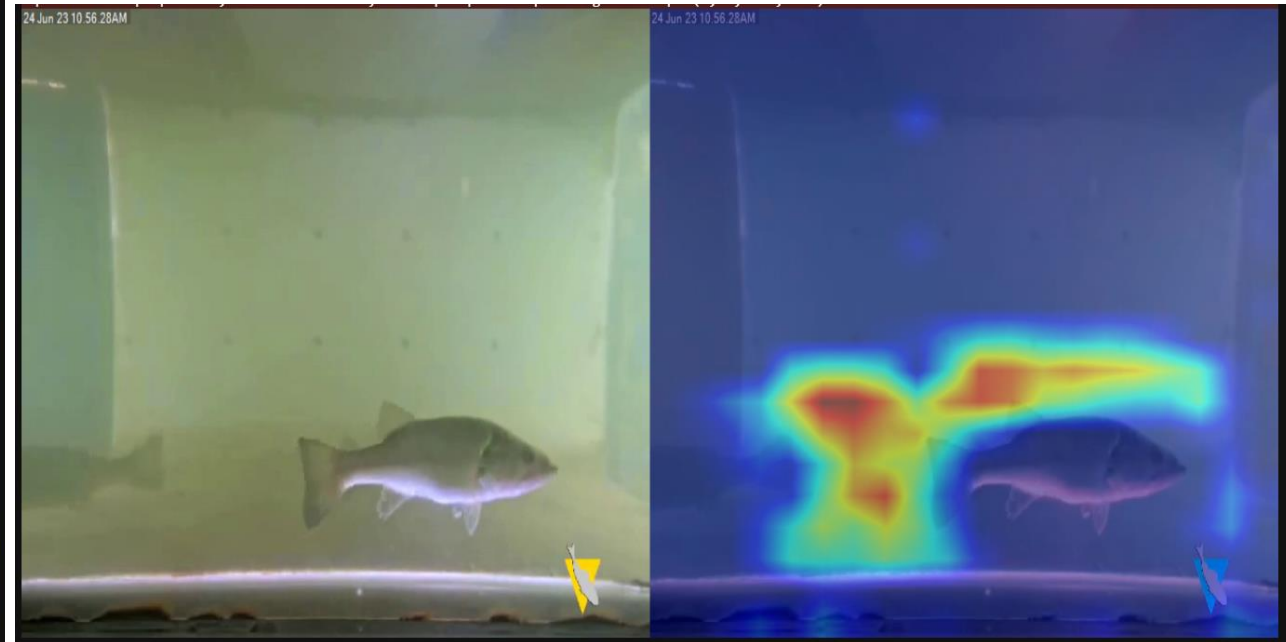
배경 학습 후

# 모델 검증

- CAM 결과 시각화



Custom Cutout 적용 전

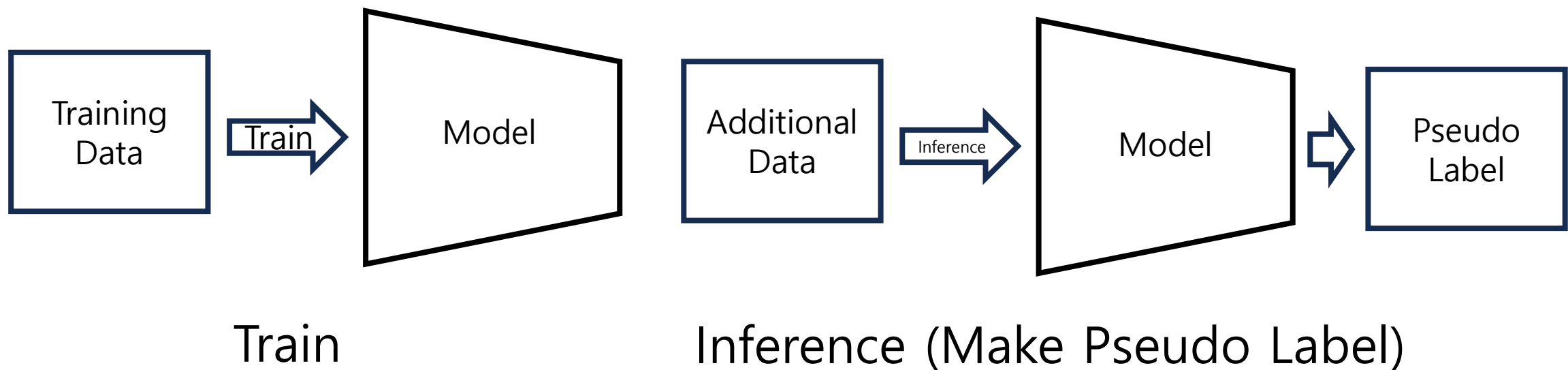


Custom Cutout 적용 후



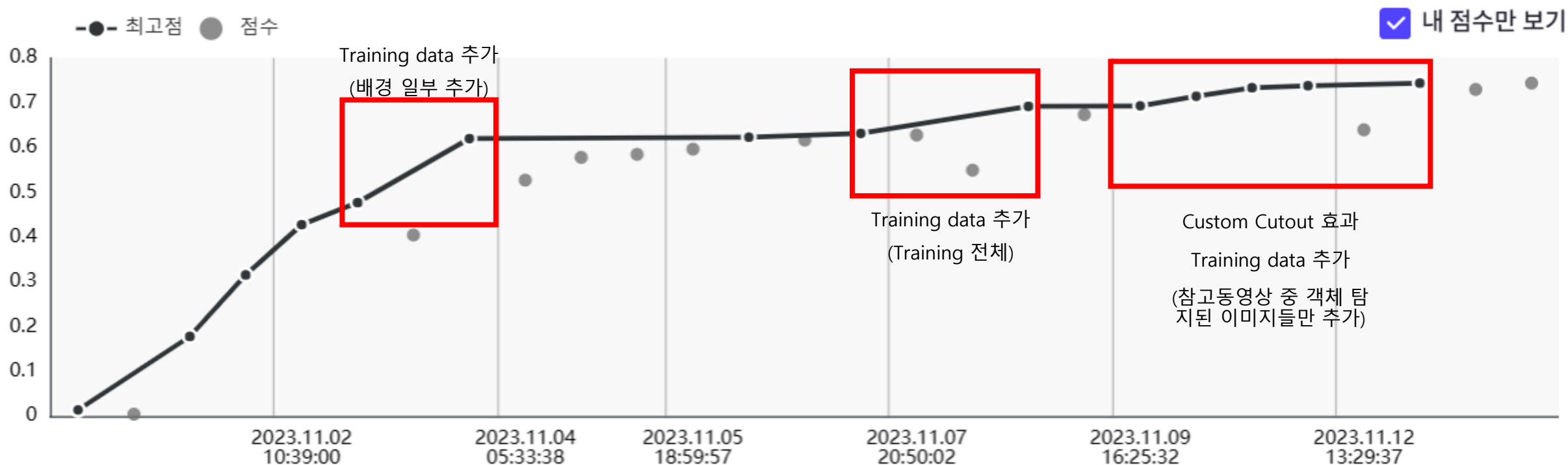
# 수도 라벨링 (Pseudo Labeling)

- 참고 데이터로 받은 동영상을 Frame 별로 이미지화 (Additional Data)
- 기존 Train 데이터로 학습한 모델에 Additional Data로 inference  
→수도 라벨링을 얻게 됨



# 실험 결과

## • Graph



# 마무리

- YOLOv8이라는 실시간으로 활용이 가능한 모델을 통해 총 8종의 물고기를 정확하게 검출하는 모델을 생성할 수 있었음
- 특히, Eigen CAM, Custom Cutout, Pseudo labeling을 통해 모델의 성능을 비약적으로 향상시킬 수 있었음
- 단일모델로도, Public & Private Score가 높은 강건하며 정확한 모델링이 가능함.
- 더 정확한 annotation 정보 이용 혹은  
WBF (Weighted Boxes Fusion)의 앙상블 기법 도입 제안