



객체탐치기술 YOLO 커스텀데이터

최석재 lingua@naver.com

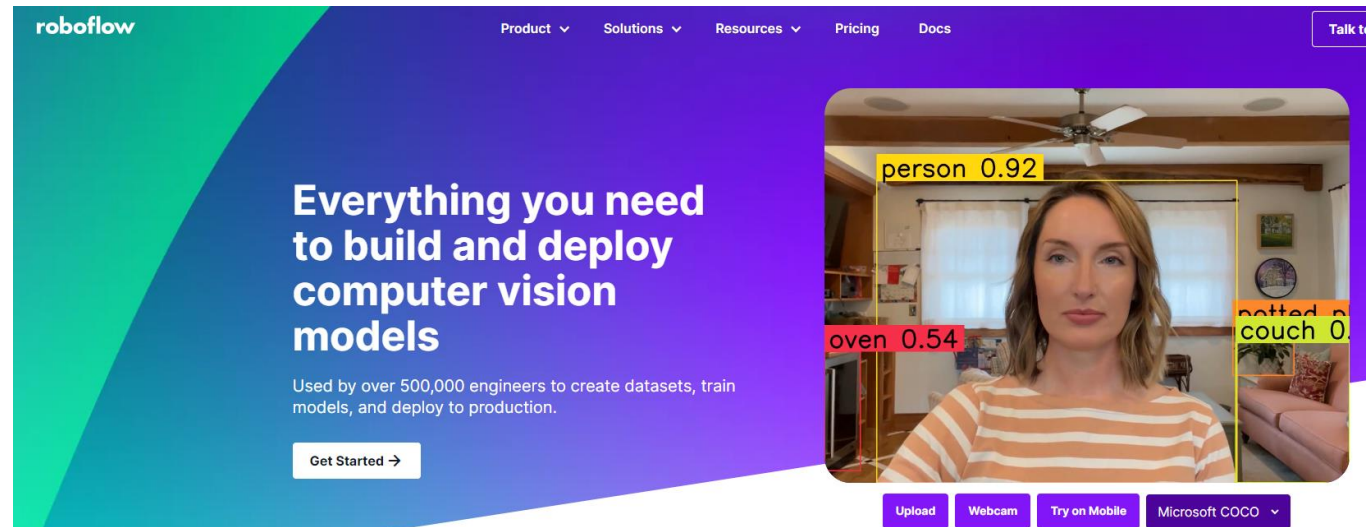
YOLO 커스텀 데이터 사용하기

- 사용자 데이터를 이용하여 YOLO 모델을 만들 수 있다
 - 그러나 ChatGPT처럼 기존의 학습 능력에 정보를 추가하는 것이 아니라,
 - 학습된 데이터의 사물만 판별할 수 있다
 - 예를 들어, YOLOv8은 버스와 사람을 판별할 수 있지만, 학습데이터에 버스와 사람이 없다면 커스텀 데이터로 학습된 모델은 이를 구분할 수 없다
-
- 즉, YOLO가 가지고 있는 학습 능력만 사용하는 것
-
- 여기서 사용할 파일은 `pytest_img > YOLO > nets` 폴더에 있다
 - 해당 폴더를 다운로드한다

roboflow를 이용한 데이터 전처리

- YOLO 학습데이터 생성을 위한 라이브러리로 labelme, labelling 등이 있다
- 최근에는 전처리 과정을 손쉽게 해주는 사이트들이 생기기 시작하였다
- 여기서는 roboflow를 이용한 데이터 전처리 작업을 알아본다
 - Roboflow는 YOLO v8 가이드에서도 추천하는 사이트
 - YOLO 모델에 따라 달라지는 전처리 작업을 자동으로 진행한다

<https://docs.ultralytics.com/modes/train/#introduction>



회원 가입

- 다음의 사이트에서 회원가입을 한다
- <https://app.roboflow.com/login>
- 구글 계정으로 가입할 경우
- 이름, 이메일 주소, 프로필 사진 정도가 공유된다

roboflow

Sign In or Sign Up



Continue with Google



Continue with Github



Continue with Email

By continuing, you are indicating that you accept our
[Terms of Service](#) and [Privacy Policy](#).

Plan 설정

- 워크스페이스와 Plan을 설정한다
- 여기서는 워크스페이스 이름으로 "YOLOv8"을,
- Plan은 Free Plan을 설정하였다
 - Free Plan도 학습용 데이터 생성은 횟수에 제한받지 않는다
- Invite teammates는 skip

Welcome! Let's get started.

Create your first workspace to house all of your projects and collaborate with teammates.

Name your workspace:

YOLOv8

Choose your plan:



Public Plan

For hobbyists, students, and personal use

Free

With public data and limited features

- Open source datasets and models on [Roboflow Universe](#)
- No Commercial Deployment License
- Model-Assisted Labeling, Image Preprocessing & Augmentations, and Dataset Health Check



Public Data



3 Training Credits



1,000 Hosted Inference API Calls



Starter Trial

For any business looking to productionize

Free 14 Day Trial

No credit card required

\$249 / mo to continue

- Private datasets and models
- Commercial Deployment License to deploy YOLOv8
- Active Learning, Automated Labeling, Outsource Labeling, Accurate Train, Model Evaluation, plus all Public Plan features.



Private Data



10 Training Credits



10,000 Hosted Inference API Calls

You can always customize and add more limits later.

Invite teammates.

Add collaborators to help with labeling, upload data, train models, and more.

Invite Teammates via Email:

joe@email.com, sara@email.com

Role: Admin

Free Plan의 경우 3회 훈련만 허가된다
유료 버전의 경우 월 \$249 약 30만원 정도가 소요되나,
상업적 이용이 가능하다

Skip

Create Workspace

Project Type 설정

- 특징이 드러나도록 Project Name과 Annotation Group을 설정한다
- 여기서는 각각 Fishing Net, net로 하였다
- Project Type은 Object Detection을 선택한다
- Create Public Project 클릭

Let's create your project.

YOLOv8 > [New Public Project](#)

Project Name

Fishing Net

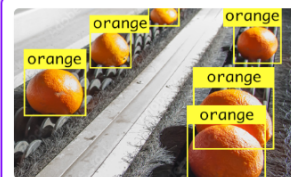
License

CC BY 4.0

Annotation Group

net

Project Type

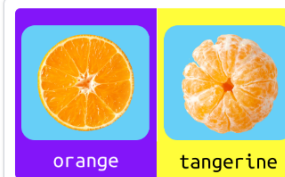


Object Detection

Identify objects and their positions with bounding boxes.

Best For

Counting 🔄 Tracking



Classification

Assign labels to the entire image.

Classification Type

☒ Multi-Label ☐ Single-Label

Best For

🔍 Filtering ✓ Content Moderation

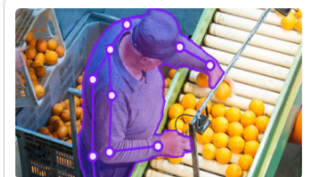


Instance Segmentation

Detect multiple objects and their actual shape.

Best For

📏 Measurements 🔄 Odd Shapes



Keypoint Detection

Identify keypoints ("skeletons") to subjects.

Best For

🏹 Pose Estimation

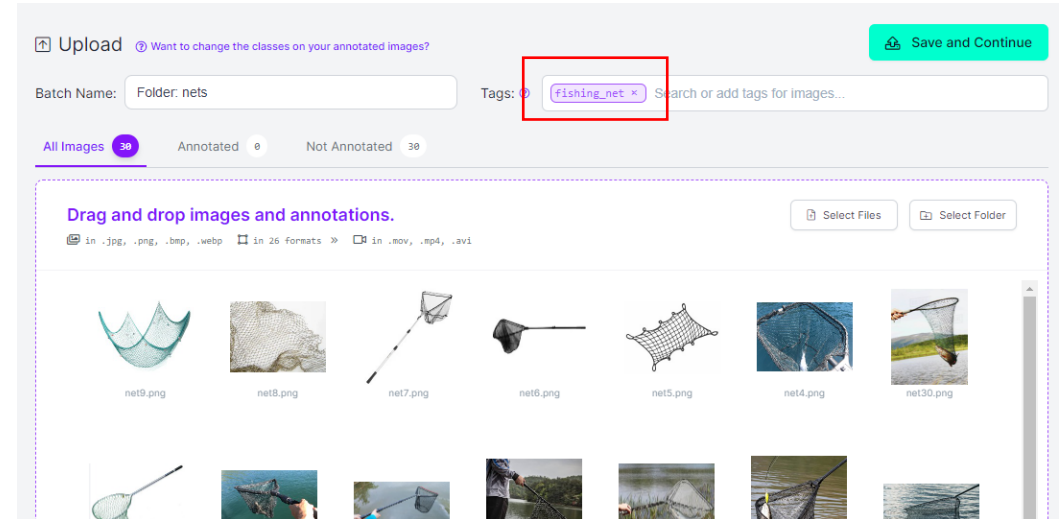
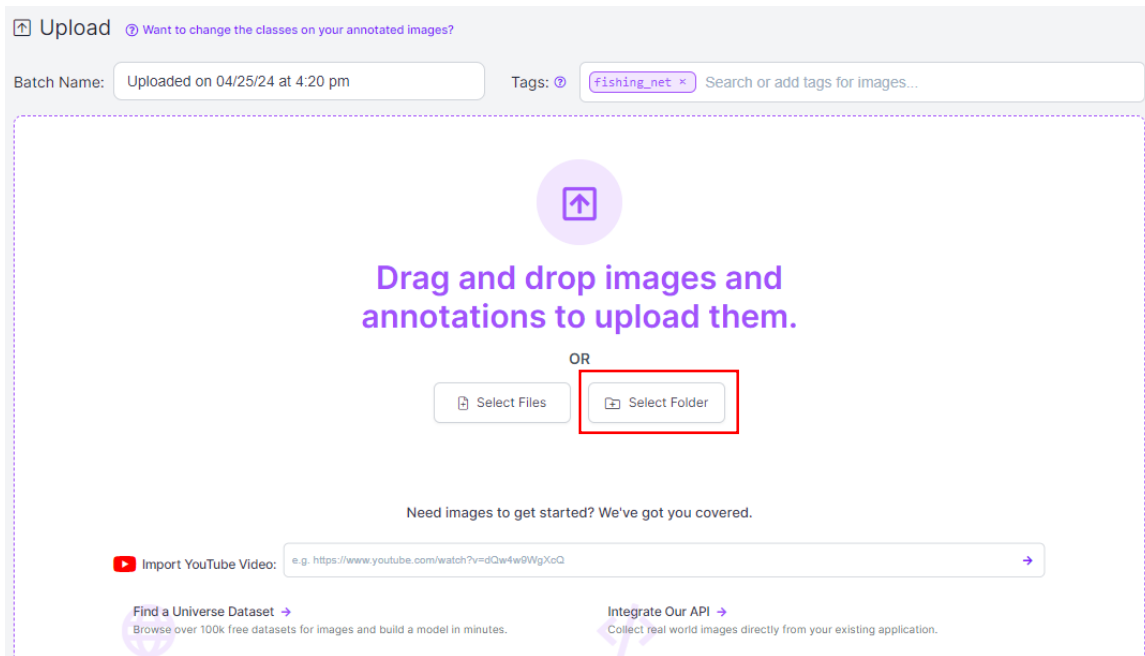
[Show More](#) ↓

Cancel

Create Public Project

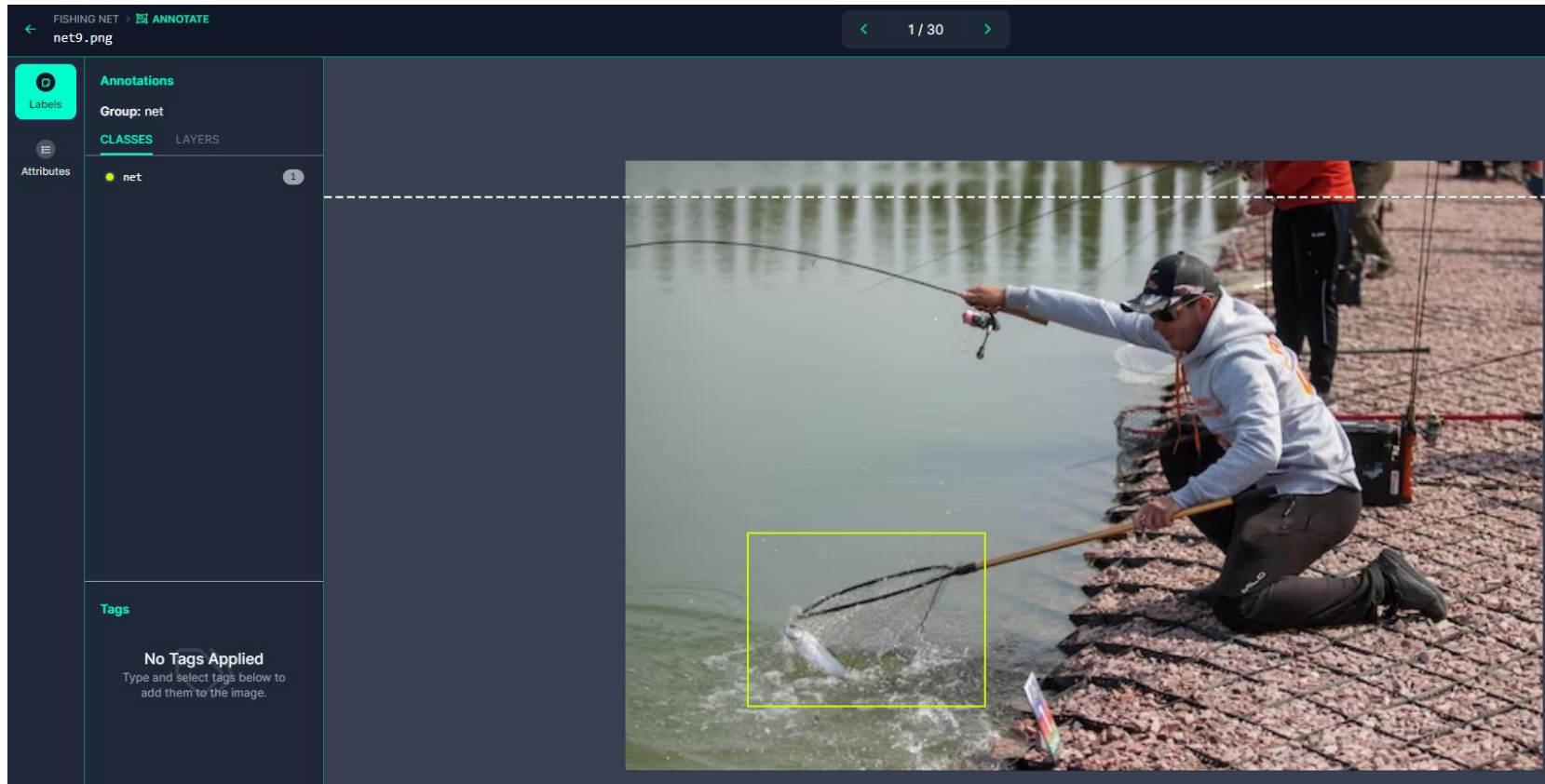
파일 업로드

- 파일 폴더의 경로를 선택하여 30개의 이미지를 업로드한다
 - 상단의 Tags에는 fishing_net라는 이름을 입력



바운딩 박스

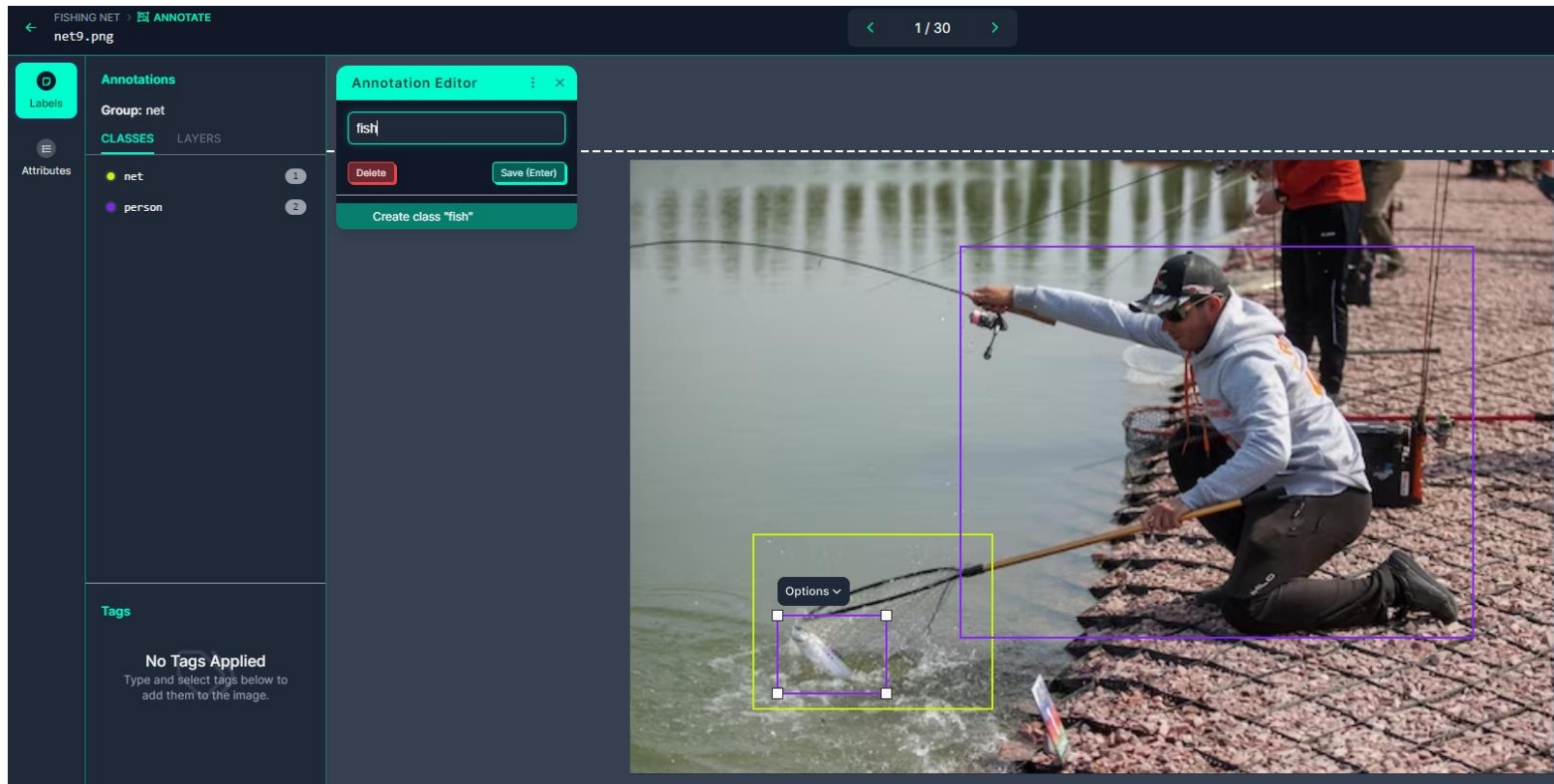
- 이미지를 하나씩 클릭하여 그물이 있는 곳을 드래그하여 바운딩 박스를 만든다
- Annotation Editor에는 'net'가 입력되어 있으니 Save 버튼 클릭



좌측 하단의 'Tags'는
작성하지 않아도 된다

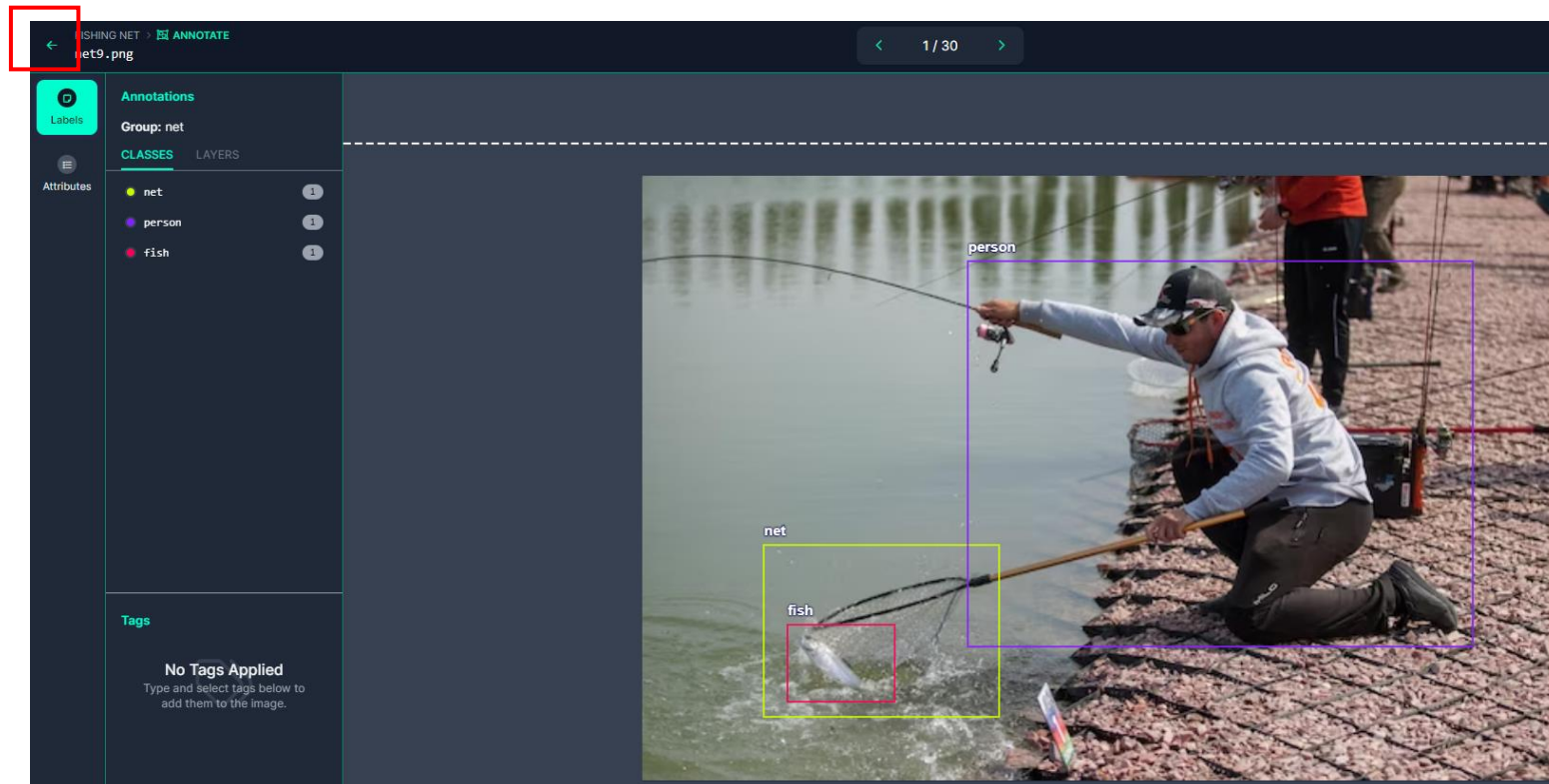
바운딩 박스

- 같은 방식으로 사람이 있는 곳과 물고기가 있는 곳에 바운딩 박스를 그리고
- Annotation Editor에는 각각 person, fish로 입력한다
- 이번 실습에서는 net, person, fish 세 가지를 사용한다



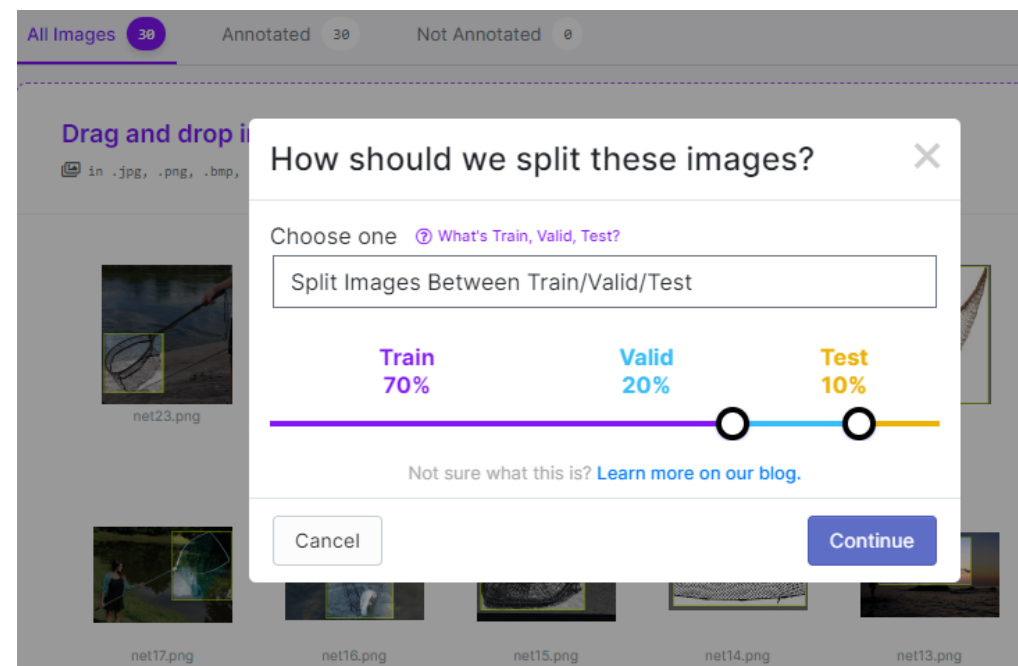
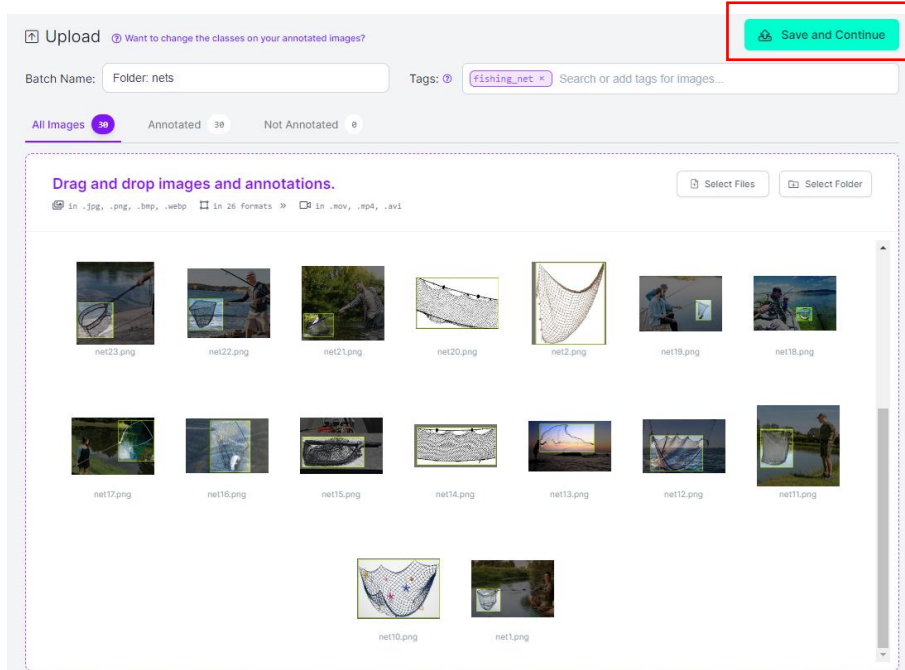
바운딩 박스

- 모두 만들어졌으면 좌측 상단의 '←' 버튼을 눌러 메인 페이지로 돌아가서 계속 작업한다
 - 뒤로 가기 버튼 또는 중앙에 있는 '<' 또는 '>'는 누르지 않는다



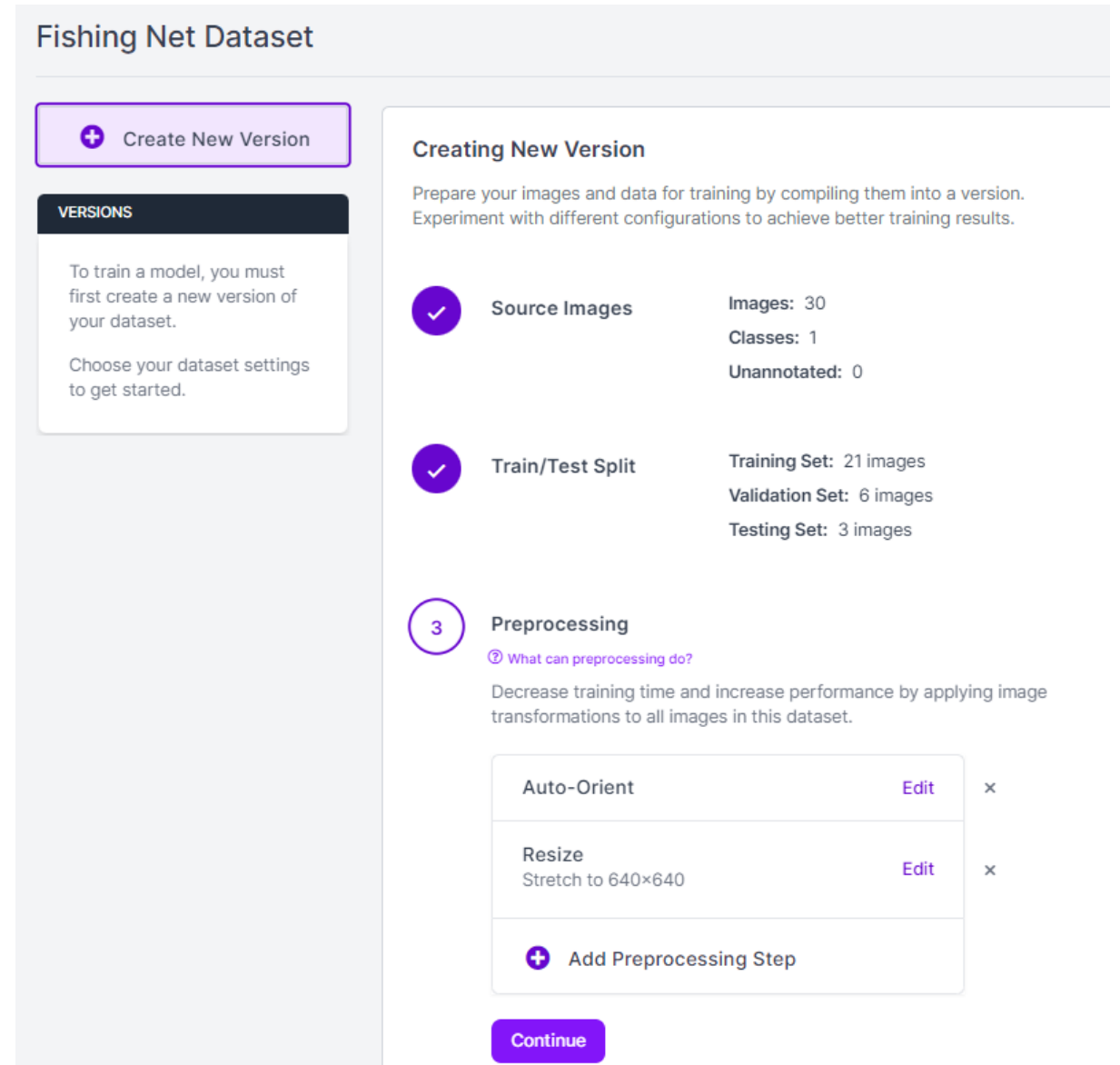
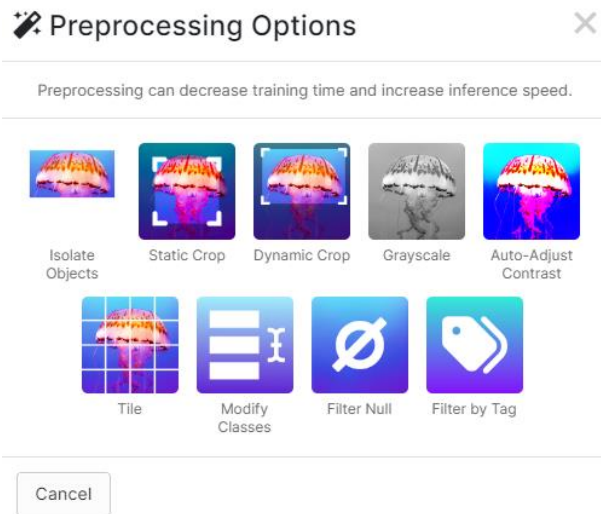
Save and Continue

- 모든 이미지에 대하여 작업이 끝났으면,
- 우상단에 있는 Save and Continue 버튼을 클릭한다
- 훈련 – 검증 – 테스트 데이터 분리 비율을 결정한다
- Continue 버튼을 누르면 파일 업로딩이 시작된다



데이터 전처리

- Continue 버튼을 클릭하여
사이즈 조정 전처리를 수행한다
- Add Preprocessing Step을 선택하면
이미지 Crop 등을 추가로 진행할 수 있다



데이터 증강

- Add Augmentation Step을 눌러 다양한 기법으로 데이터 증강을 한다

4

Augmentation

[? What can augmentation do?](#)

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

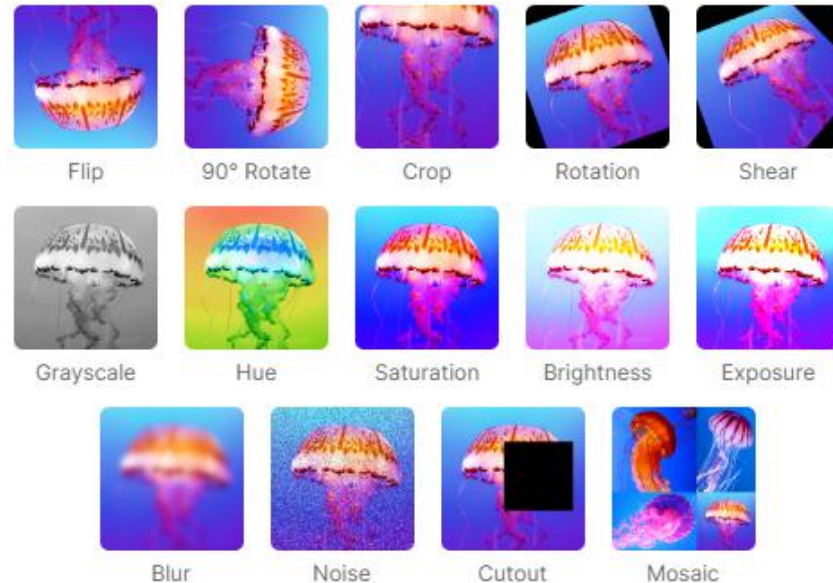
[+ Add Augmentation Step](#)

Continue

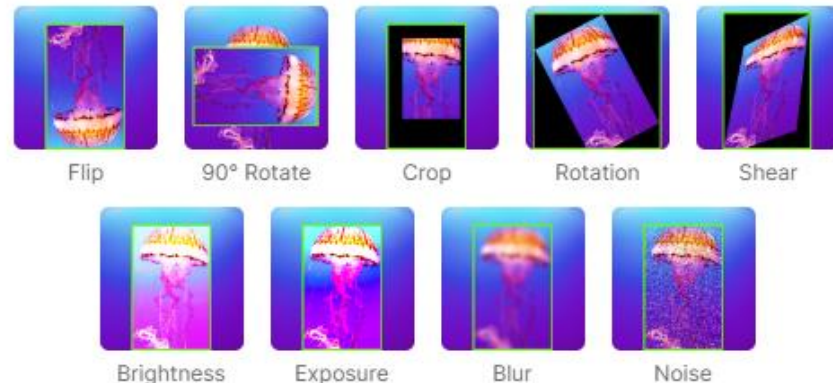
Augmentation Options

Augmentations create new training examples for your model to learn from.

IMAGE LEVEL AUGMENTATIONS



BOUNDING BOX LEVEL AUGMENTATIONS ?



Cancel

데이터 증강 선택 결과

- 준비된 이미지에 맞게 적절한 증강 기법을 선택한다
- Continue

4

Augmentation

[? What can augmentation do?](#)

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

Flip Horizontal	Edit	×
Rotation Between -15° and +15°	Edit	×
Hue Between -15° and +15°	Edit	×
Saturation Between -25% and +25%	Edit	×
Exposure Between -10% and +10%	Edit	×
+ Add Augmentation Step		

Continue

증강 실행

- 선택한 기법으로 데이터를 증강한다
- 유료 버전에서는 50배까지 많은 데이터를 만들 수 있다
- Create

5

Create

Review your selections and select a version size to create a moment-in-time snapshot of your dataset with the applied transformations.

Larger versions take longer to train but often result in better model performance. [See how this is calculated >>](#)

Maximum Version Size

72 images (3x)

Create



5

51 images (2x)
72 images (3x)
93 images (4x - Requires Upgrade)
114 images (5x - Requires Upgrade)
135 images (6x - Requires Upgrade)
156 images (7x - Requires Upgrade)
177 images (8x - Requires Upgrade)
198 images (9x - Requires Upgrade)
219 images (10x - Requires Upgrade)
240 images (11x - Requires Upgrade)
261 images (12x - Requires Upgrade)
282 images (13x - Requires Upgrade)
303 images (14x - Requires Upgrade)
324 images (15x - Requires Upgrade)
345 images (16x - Requires Upgrade)
366 images (17x - Requires Upgrade)
387 images (18x - Requires Upgrade)
408 images (19x - Requires Upgrade)
429 images (20x - Requires Upgrade)
450 images (21x - Requires Upgrade)

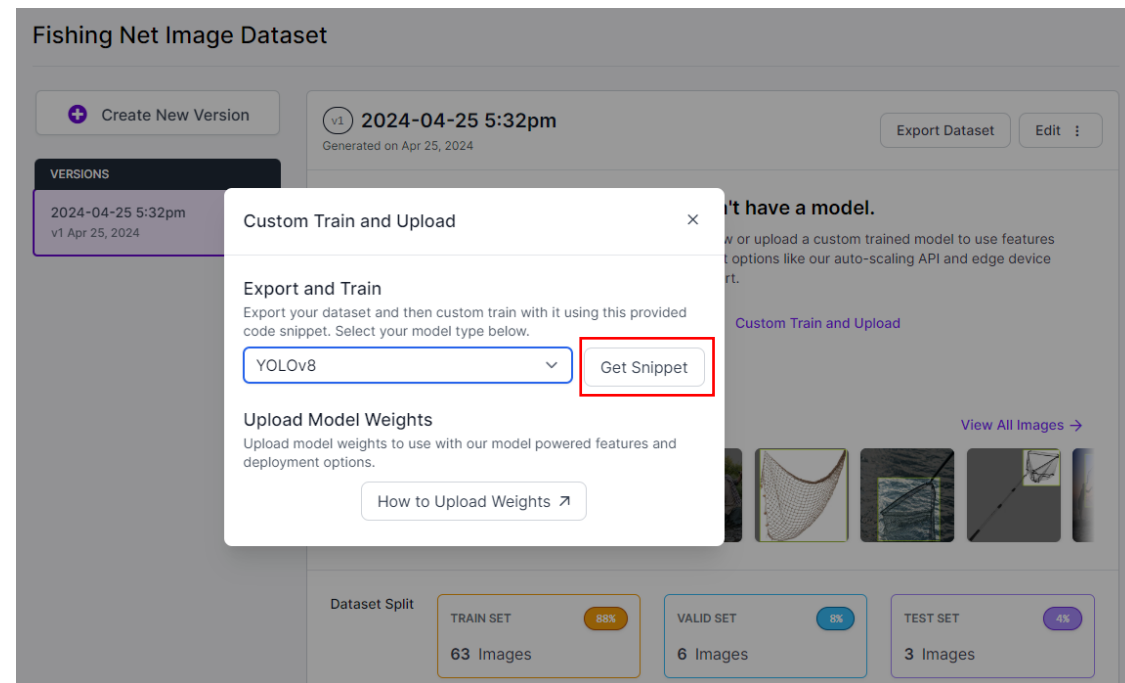
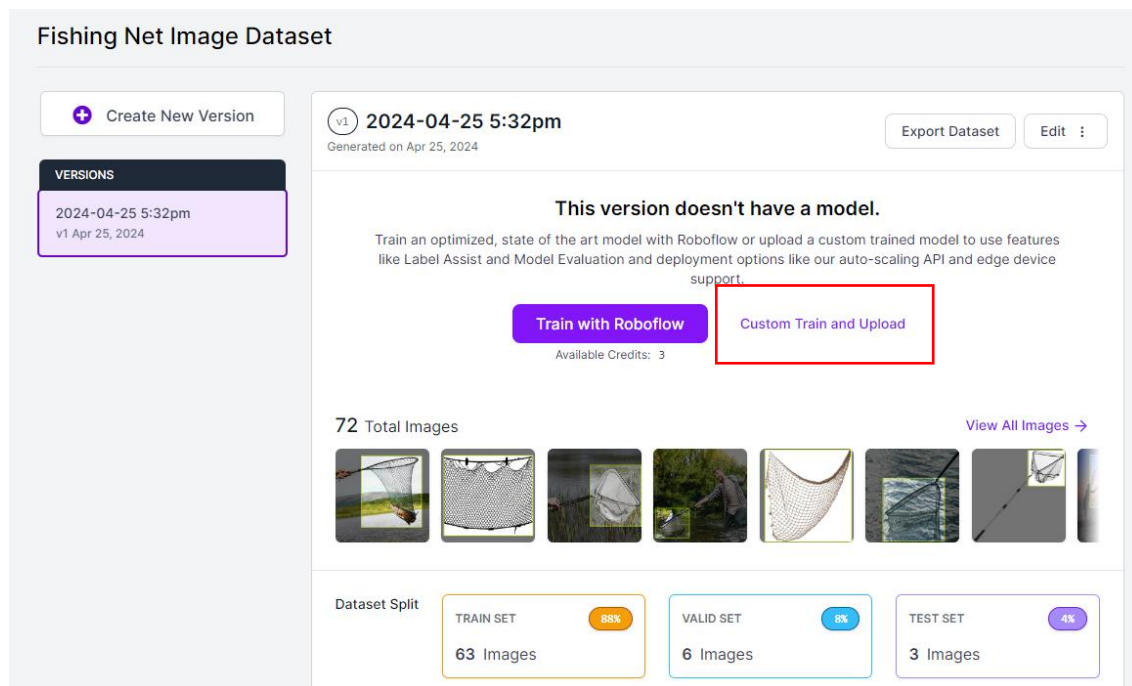
72 images (3x)

Create

between -15° and +15°
-15° and +15°
between -25% and +25%
between -10% and +10%
size to create a moment-in-time
transformations.
to result in better model

Custom Train and Upload

- 우측의 Custom Train and Upload를 선택한다
- Model을 YOLOv8을 선택한 뒤, Get Snippet 버튼을 클릭
 - 좌측의 Train with Roboflow 버튼을 클릭하면 Roboflow를 이용해 자동화된 학습까지 할 수 있다 (무료 3회)



Jupyter Code 복사

- Jupyter 탭의 복사 버튼을 눌러 코드를 복사한다
- 아래쪽의 Copy Snippet and Open YOLOv8 ... 버튼은 클릭하지 않는다

버튼을 클릭하면 아래의 매뉴얼 사이트로 간다

<https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb>

- roboflow 사이트는 달아도 된다

The screenshot shows the Roboflow web application. At the top, there's a header with "Your Download Code" and a close button. Below the header, there are three tabs: "Jupyter" (selected), "Terminal", and "Raw URL". The main content area displays a Python code snippet for installing RoboFlow and downloading a dataset. A red box highlights a copy icon next to the code. Below the code, there is a warning message.

```
!pip install roboflow

from roboflow import RoboFlow
rf = RoboFlow(api_key="[REDACTED]")
project = rf.workspace("yolov8-").project("fishing-net")
version = project.version(1)
dataset = version.download("yolov8")
```

Warning: Do not share this snippet beyond your team, it contains a private key that is tied to your Roboflow account. Acceptable use policy applies.

Colab 열기

- Colab 파일을 연다
- 런타임 유형은 우선 CPU를 선택하여 코드를 작성하고,
- 이후 실행할 때 GPU와 고용량 메모리를 선택한다

런타임 유형 변경

런타임 유형

Python 3

하드웨어 가속기 ?

☐ CPU ☒ A100 GPU ☐ L4 GPU

☐ V100 GPU (deprecated) ☐ T4 GPU

☐ TPU (deprecated) ☐ TPU v2

고용량 RAM ☐

취소

저장

환경 확인

- !pwd 명령어로 현재 경로가 /content 에 있는지 확인한다
- 이 경로에서만 아래의 과정이 잘 진행된다

```
!pwd
/content
```

- !nvidia-smi 명령어로 GPU 자원 사용 여부를 알아본다

```
!nvidia-smi
```

```
Thu Apr 25 10:12:36 2024
```

NVIDIA-SMI 535.104.05 Driver Version: 535.104.05 CUDA Version: 12.2									
GPU		Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC			
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	MIG M.		
0	N/A	35C	P8	12W / 72W	Off	00000000:00:03:0 Off	1MiB / 23034MiB	0%	Default
								N/A	

Processes:							GPU Memory
GPU ID	GI ID	CI ID	PID	Type	Process name		Usage
No running processes found							

ultralalytics 설치

- !pip install ultralytics 명령어로 라이브러리를 설치한다

✓
0초

[1] !nvidia-smi

/bin/bash: line 1: nvidia-smi: command not found

✓
1분

[2] !pip install ultralytics

Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.66.2)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from ultralytics) (5.9.5)
Requirement already satisfied: py-cpuinfo in /usr/local/lib/python3.10/dist-packages (from ultralytics) (9.0.0)
Collecting thop>=0.1.1 (from ultralytics)
 Downloading thop-0.1.1.post2209072238-py3-none-any.whl (15 kB)
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.0.3)
Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (0.13.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralalytics)

YOLO 모델 불러오기

- from ultralytics import YOLO
 - import os
 - from IPython.display import Image
 - from IPython import display
-
- display.clear_output()
 - !yolo checks

이전 출력 내용 지우기

YOLO와 함께 설치된 라이브러리 확인

Ultralytics YOLOv8.2.2 Python-3.10.12 torch-2.2.1+cu121 CPU (Intel Xeon 2.20GHz)
Setup complete (2 CPUs, 12.7 GB RAM, 29.0/225.8 GB disk)

OS	Linux-6.1.58+-x86_64-with-glibc2.35
Environment	Colab
Python	3.10.12
Install	pip
RAM	12.67 GB
CPU	Intel Xeon 2.20GHz
CUDA	None

matplotlib	3.7.1>=3.3.0
opencv-python	4.8.0.76>=4.6.0
pillow	9.4.0>=7.1.2
pyyaml	6.0.1>=5.3.1
requests	2.31.0>=2.23.0
scipy	1.11.4>=1.4.1
torch	2.2.1+cu121>=1.8.0
torchvision	0.17.1+cu121>=0.9.0
tqdm	4.66.2>=4.64.0
psutil	5.9.5
py-cpuinfo	9.0.0
thop	0.1.1-2209072238>=0.1.1
pandas	2.0.3>=1.1.4
seaborn	0.13.1>=0.11.0

데이터셋 다운로드

- 복사한 코드를 실행하여 Roboflow에 있는 데이터셋을 다운로드한다
- 이 과정에서 세션이 다운되면 세션 다시 시작 후 처음부터 다시 실행한다

- !pip install roboflow

- from roboflow import Roboflow
- rf = Roboflow(api_key="YOUR-API-KEY")
- project = rf.workspace("yolov8-YOURS").project("fishing-net")
- version = project.version(1)
- dataset = version.download("yolov8")

세션 다시 시작

WARNING: The following packages were previously imported in this runtime:
[certifi, chardet, cv2, cyclert, idna]

You must restart the runtime in order to use newly installed versions.

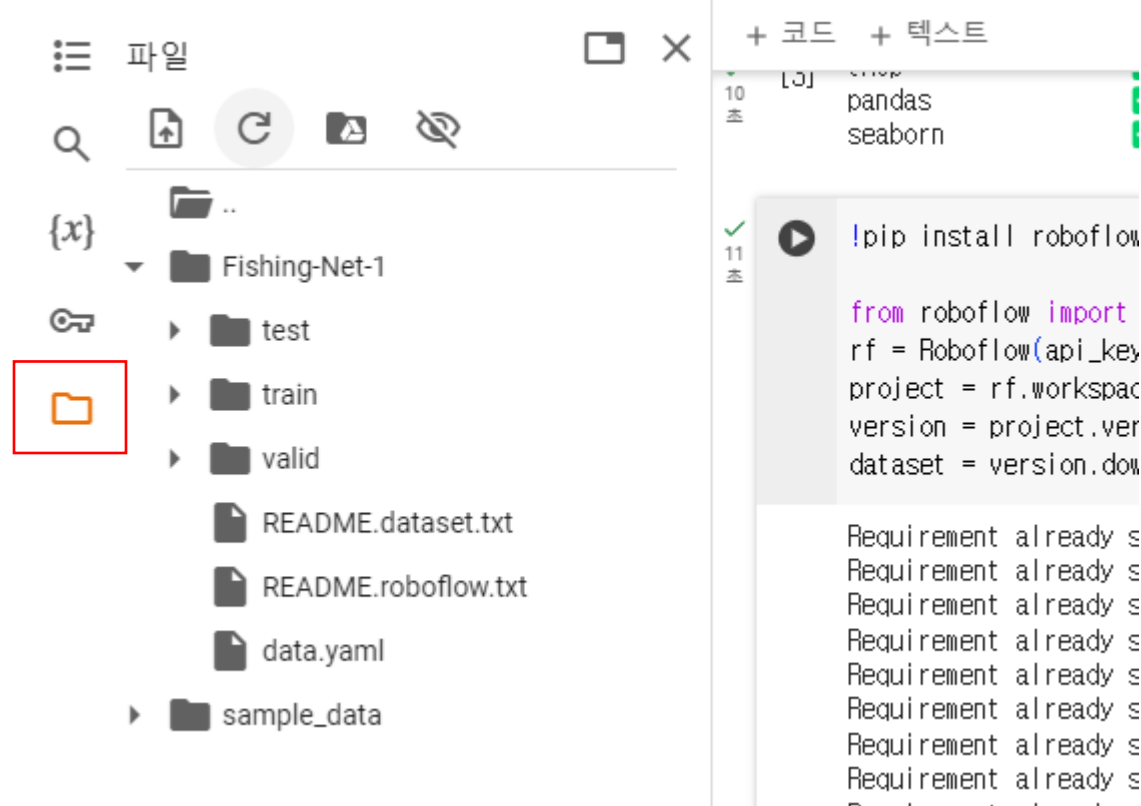
Restarting will lose all runtime state, including local variables.

취소

세션 다시 시작

다음으로 확인

- Colab의 폴더 버튼을 클릭하면 데이터가 다운로드 된 것을 확인할 수 있다
- Fishing-Net-1 아래에 test, train, valid 데이터 폴더가 생성되었다



데이터셋 확인

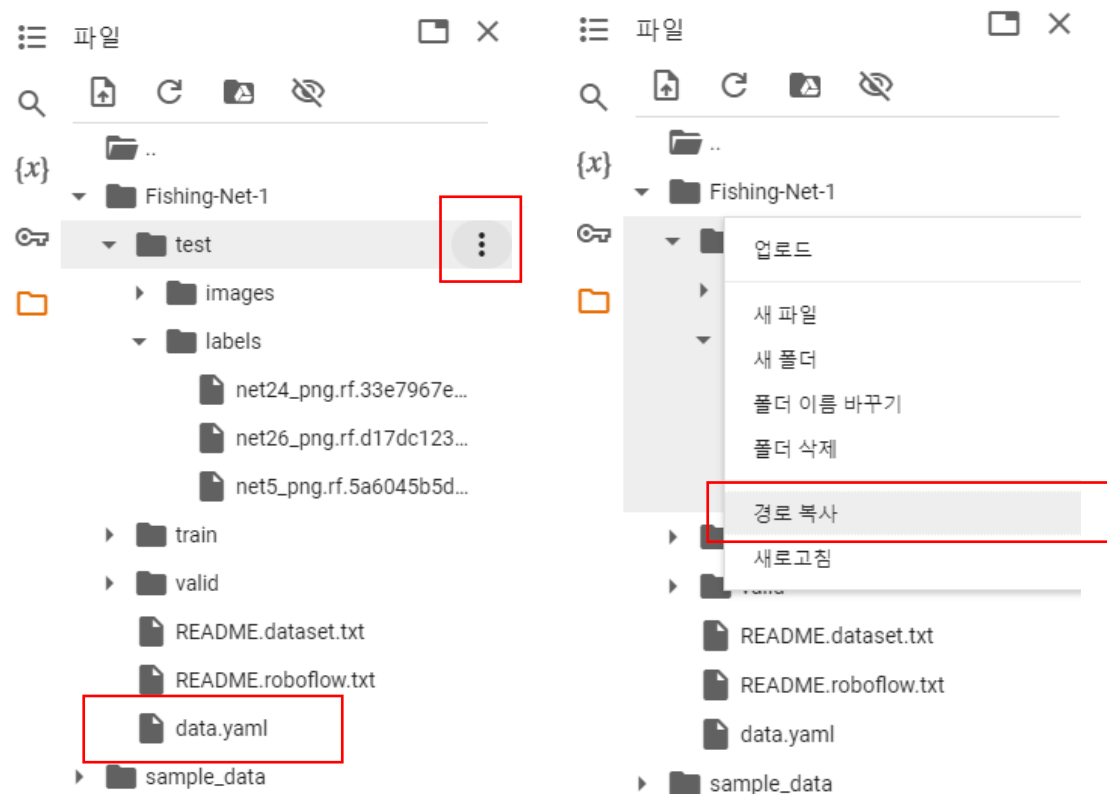
- test 폴더의 images와 labels 폴더를 살펴보면
- 각각 이미지와 바운딩 박스 좌표가 있는 것을 볼 수 있다



data.yaml

- data.yaml 파일은 YOLO 모델 학습에 필요한 구조 정보를 갖고 있다
- 이 파일의 경로 정보를 수정해야 한다

- test 폴더의 우측 버튼을 클릭하여
- [경로 복사] 버튼으로 경로를 복사한다



data.yaml 수정

- data.yaml 파일을 열고
- test 경로 부분을 복사한 내용으로 아래와 같이 수정한다
- 이 작업을 train, val 에도 수행한다
- 파일 저장 Ctrl-s

```
data.yaml x
1 names:
2 - net
3 nc: 1
4 roboflow:
5   license: CC BY 4.0
6   project: fishing-net
7   url: https://universe.roboflow.com/yolov8-1/fishing-net/dataset/1
8   version: 1
9   workspace: yolov8-o
10 test: /content/Fishing-Net-1/test/images
11 train: /content/Fishing-Net-1/train/images
12 val: /content/Fishing-Net-1/valid/images
```


모델 학습

- 다음의 코드로 모델을 학습시킨다
- 학습이 종료되면 runs/detect/train n 폴더에 저장된다
결과는 train 일 수도 있고, train1, train2, train3, ... 일 수도 있다
- !yolo task=detect mode=train model=yolov8m.pt
data={dataset.location}/data.yaml epochs=20 imgsz=640

20 epochs completed in 0.016 hours.

Optimizer stripped from runs/detect/train/weights/last.pt, 52.0MB

Optimizer stripped from runs/detect/train/weights/best.pt, 52.0MB

Validating runs/detect/train/weights/best.pt...

Ultralytics YOLOv8.2.2 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (NVIDIA L4, 22700MiB)

Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	6	18	0.885	0.508	0.664	0.466
fish	6	5	1	0	0.172	0.0559
net	6	6	0.914	0.667	0.927	0.762
person	6	7	0.74	0.857	0.893	0.58

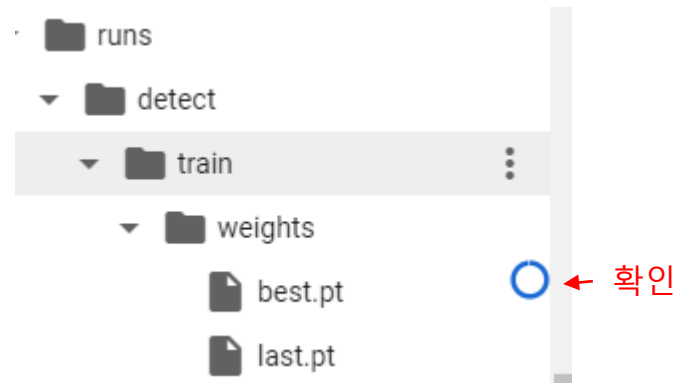
Speed: 0.2ms preprocess, 3.6ms inference, 0.0ms loss, 9.6ms postprocess per image

Results saved to runs/detect/train

💡 Learn more at <https://docs.ultralytics.com/modes/train>

Best 모델 다운로드

- 가장 좋은 성능의 모델은 runs/detect/train~~n~~/weights/best.pt
- 우측 버튼을 클릭하여 다운로드할 수 있다
- 다운로드 시 약 1~2분 정도 소요된다



검증데이터 평가

- 다음의 코드로 검증 데이터를 평가한다
- `!yolo task=detect mode=val`
`model=/content/runs/detect/trainn/weights/best.pt`
`data={dataset.location}/data.yaml`
`/content/Fishing-Net-1`

```
Ultralytics YOLOv8.2.2 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (NVIDIA L4, 22700MiB)
Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs
val: Scanning /content/Fishing-Net-1/valid/labels.cache... 6 images, 0 backgrounds, 0 corrupt: 100% 6/6 [00:00<?, ?it/s]

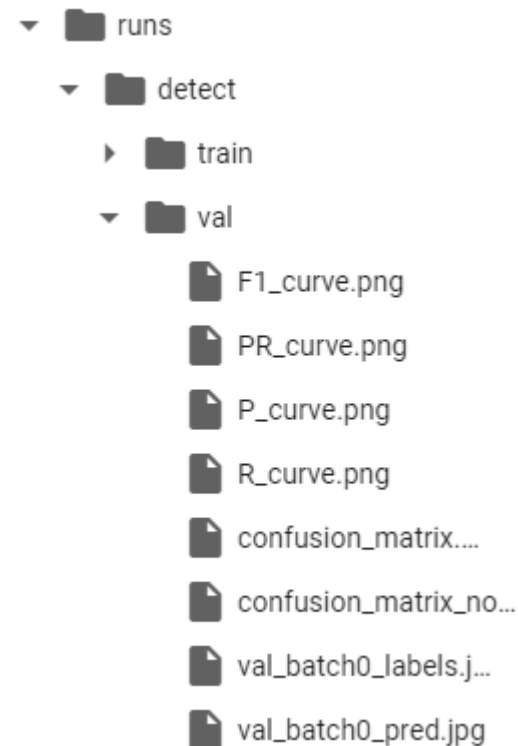
```

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	6	18	0.88	0.508	0.664	0.471
fish	6	5	1	0	0.174	0.0565
net	6	6	0.914	0.667	0.927	0.762
person	6	7	0.725	0.857	0.893	0.594

```
Speed: 0.5ms preprocess, 39.8ms inference, 0.0ms loss, 340.9ms postprocess per image
Results saved to runs/detect/val
🔗 Learn more at https://docs.ultralytics.com/modes/val
```

검증데이터 평가 결과

- runs/detect/val 폴더가 생성되었으며, 결과가 저장되어 있다
- 결과는 다운로드 할 수 있다



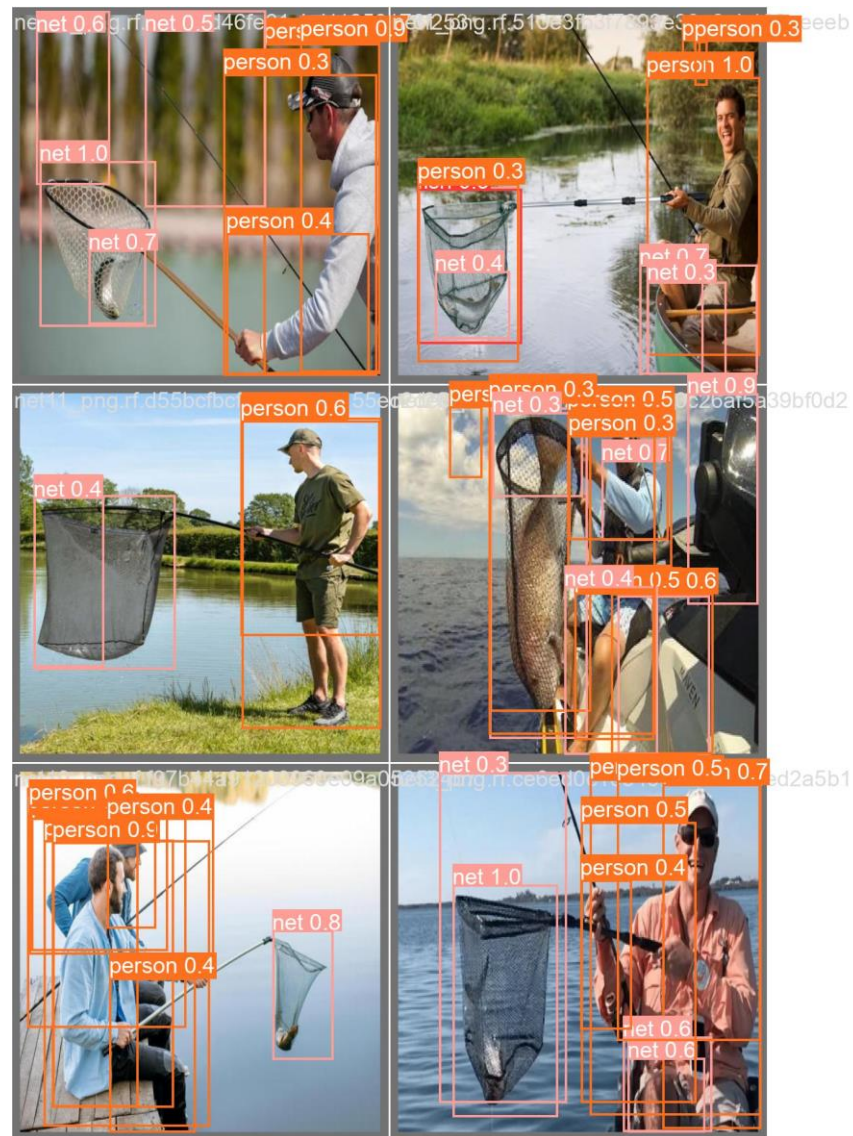
평가 중 이미지

정답



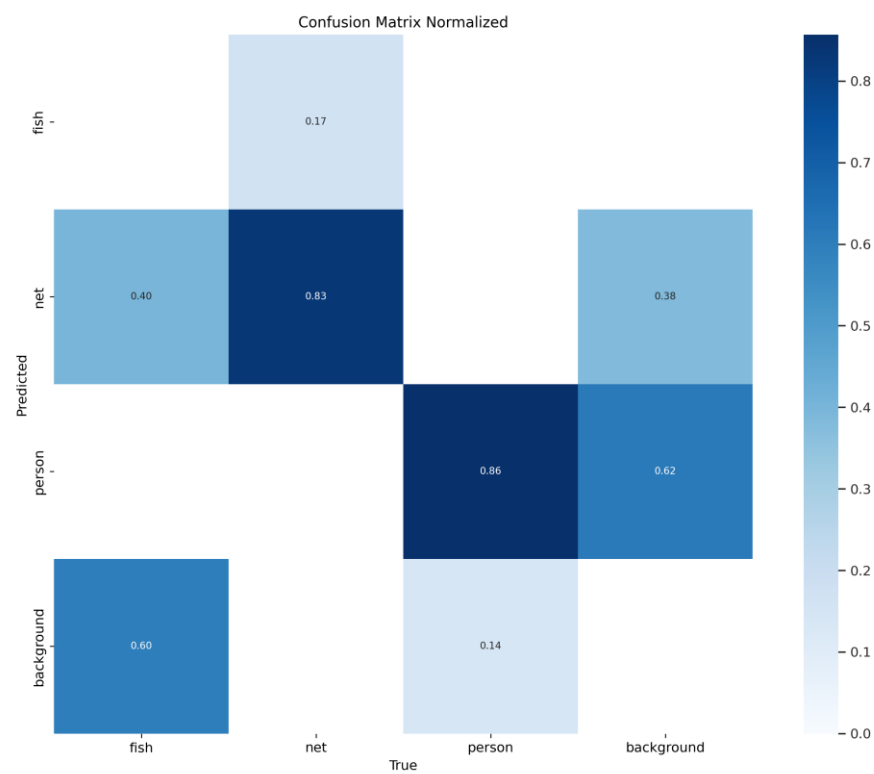
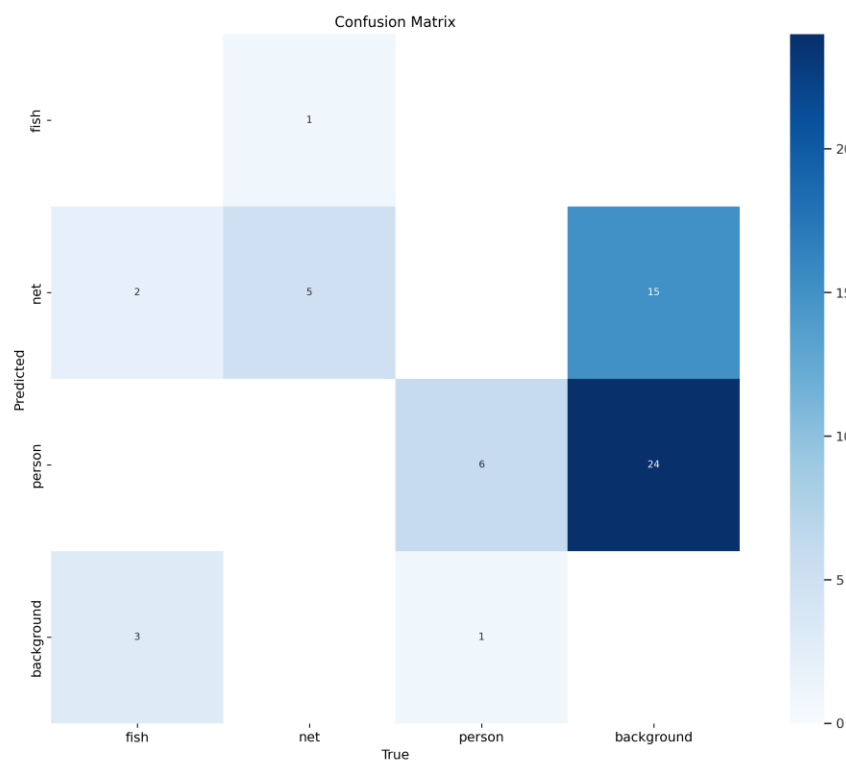
전체 신뢰도 결과를 보여준다

예측 결과



Confusion Matrix

- 배경, net, person, fish의 구분 결과를 보여준다
- 그러나 일반적인 혼동행렬과는 다르고, 이미지로 본 결과와 일치하는지 확인이 필요하다



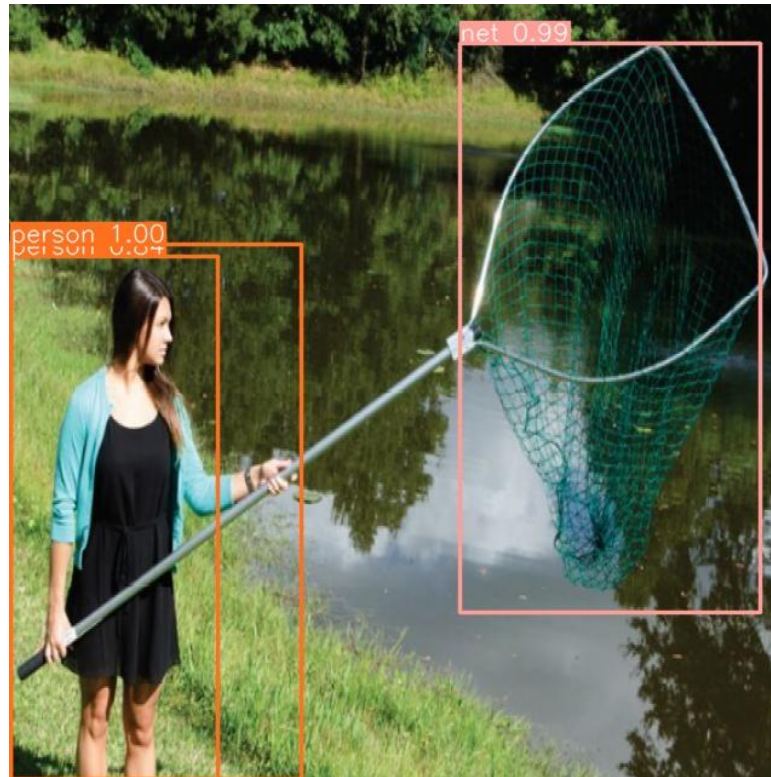
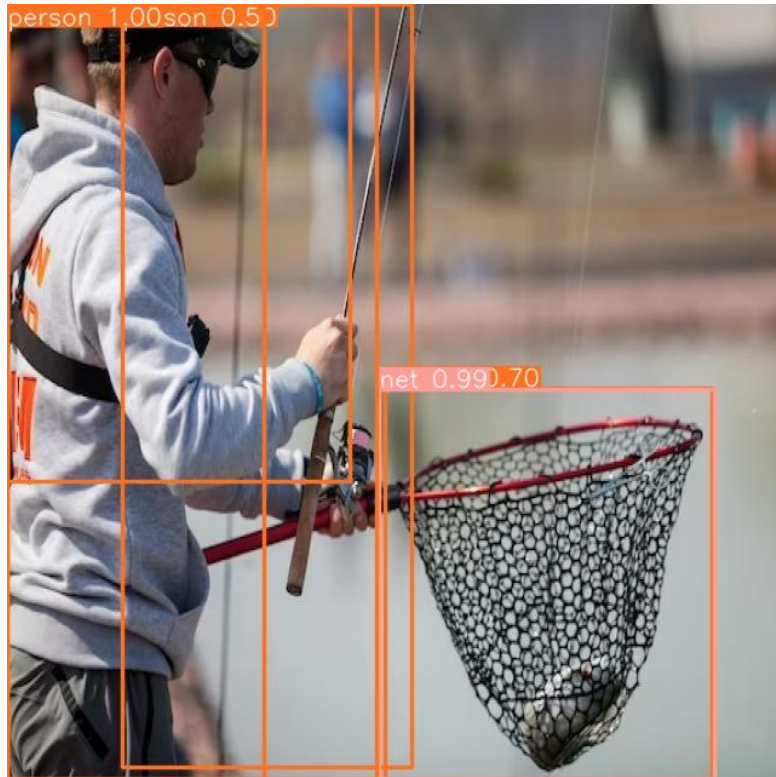
테스트 데이터 평가

- 다음의 코드로 테스트 데이터를 평가한다
- `!yolo task=detect mode=predict
model=/content/runs/detect/trainn/weights/best.pt conf=0.5
source={dataset.location}/testn/images
/content/Fishing-Net-1`

```
WARNING ⚠ 'source' argument is missing. Using default 'source=/usr/local/lib/python3.10/dist-packages/ultralytics/assets'.  
Ultralytics YOLOv8.2.2 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (NVIDIA L4, 22700MiB)  
Model summary (fused): 218 layers, 25840339 parameters, 0 gradients, 78.7 GFLOPs  
  
image 1/2 /usr/local/lib/python3.10/dist-packages/ultralytics/assets/bus.jpg: 640x480 7 nets, 148.8ms  
image 2/2 /usr/local/lib/python3.10/dist-packages/ultralytics/assets/zidane.jpg: 384x640 3 nets, 148.2ms  
Speed: 2.9ms preprocess, 148.5ms inference, 972.9ms postprocess per image at shape (1, 3, 384, 640)  
Results saved to runs/detect/predict  
💡 Learn more at https://docs.ultralytics.com/modes/predict
```

테스트 데이터 평가 결과

- 테스트 데이터 결과는 잘 이루어졌다



동영상 재분석

- 다운로드받은 best.pt 파일을 YOLO 모델 경로에 넣고 예: C:\Users\<사용자계정>\PythonProjects\models
- sample.mp4 데이터를 다시 분석해본다
- 사람은 잘 판정하나, 그물 판정 결과는 매우 좋지는 않다
 - 학습 데이터에 유사한 그물이 많지 않은 것이 이유로 보인다

