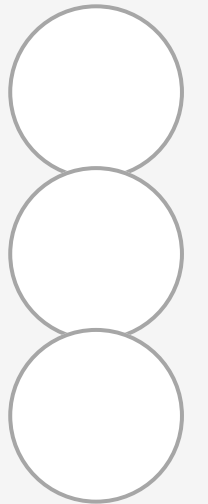


# SAM3

# PARTICLE SEGMENTATION



**박정규**

DeepSharkLab\_학부연구생

# Contents

1

## PipeLine

Segmentation  
Inpainting

2

## SAM3

Segmentation  
Anything  
Model

3

## Result

SAM3

4

## Limit

Particle  
segmentation

5

## Plan

Fine tuning

# PipeLine

1

## Video Separation

영상을 프레임 단위로 분리

2

## Image Segmentation

부유물의 Polygon 외곽선 추출 후  
외곽선 안쪽 픽셀의 RGB 값을  
(0,0,0) 으로 변경

3

## Inpainting

Segmentation 된 부분을  
Inpainting Model을 활용하여  
채우기

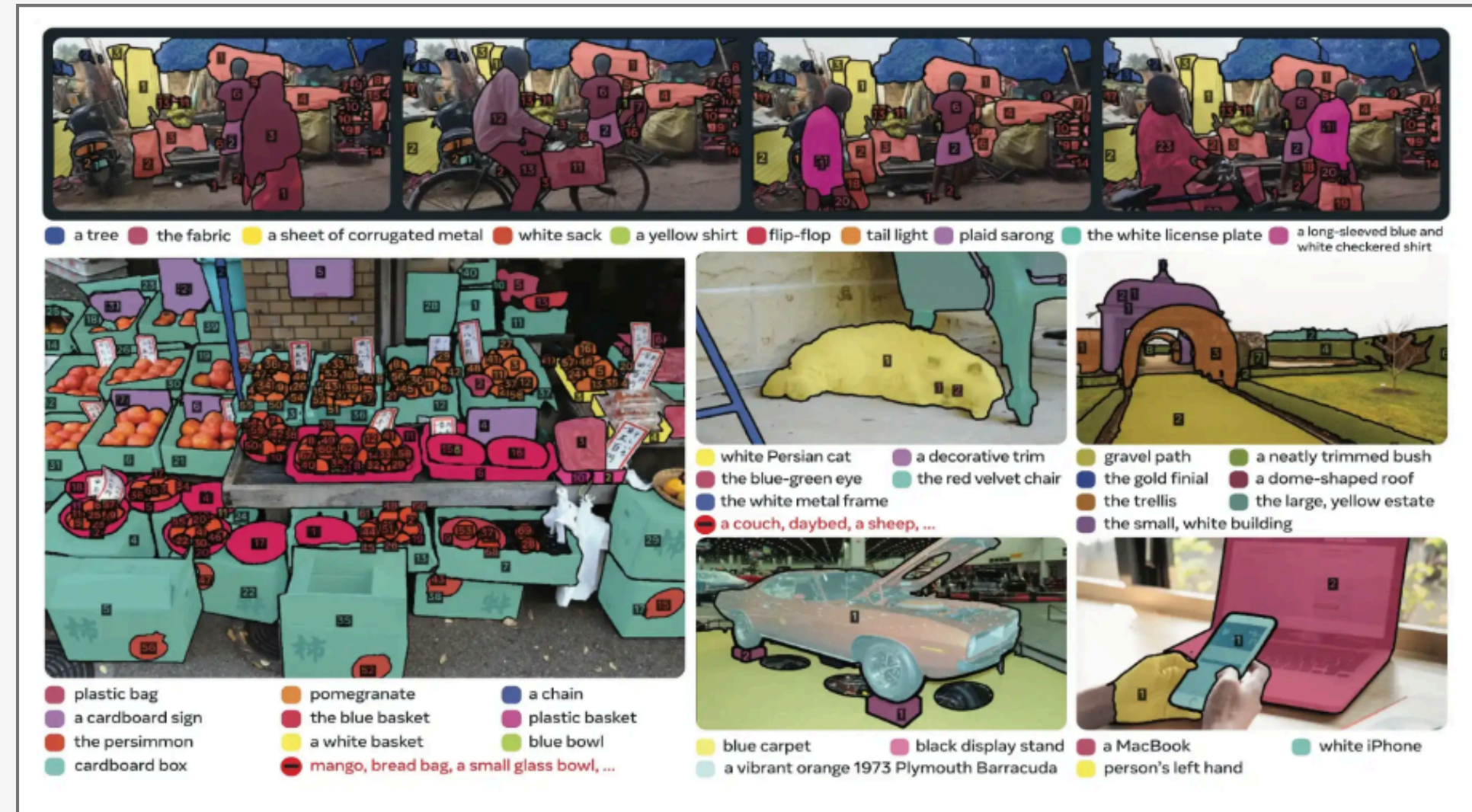
# SAM3

Segmentation Anything Model

Developed by Meta / November 19, 2025

Key Innovations 개념 및 시각적 프롬프트의 통합  
상호작용형 예시 개선

Package Ultralytics에 병합되어 함수와 클래스는  
Ultralytics package에 포함되어있음  
Weight parameter는 다운로드 필요



<https://docs.ultralytics.com/ko/models/sam-3>

# SAM3

Promptable Concept Segmentation

**Method** Text prompt 를 통해 해당 프롬프트와 동일한 모든 객체 segmentation

**Prompt** Open-vocabulary 을 지원하여 학습된 Class 뿐만 아니라 다른 Class도 prompt로 입력 가능  
단순 객체뿐만 아니라 "선글라스를 쓴 사람" 과 같은 묘사적인 표현도 지원

**Example** Prompt를 “person”으로 입력시 segmentation 진행 X

Prompt를 “penguin”으로 입력시 Segmentation 진행



# SAM3

## Promptable Concept Segmentation

```
1 from ultralytics.models.sam import SAM3SemanticPredictor
2 from PIL import Image
3 import numpy as np
4
5 # Initialize predictor with configuration
6 overrides = dict(
7     conf=0.1,
8     task="segment",
9     mode="predict",
10    model= "sam3.pt download path",
11    half=True, # Use FP16 for faster inference
12    save=True,
13 )
14 predictor = SAM3SemanticPredictor(overrides=overrides)
15
16 # Set image once for multiple queries
17 predictor.set_image("image path")
18
19 #Works with descriptive phrases
20 results = predictor(text=["prompt"])
21
```

```
# Query with multiple text prompts
results = predictor(text=["person", "bus", "glasses"])

# Works with descriptive phrases
results = predictor(text=["person with red cloth", "person with blue cloth"])

# Query with a single concept
results = predictor(text=["a person"])
```

# SAM3

Promptable Concept Segmentation

**Method** bbox Prompt를 통해 bbox 객체와 동일한 모든 객체 segmentation

**Prompt** bbox 좌측 상단 ( $X_1, Y_1$ ) / 우측 하단 ( $X_2, Y_2$ ) 좌표

bbox 개수는 여러개 가능

text prompt, bbox 동시 사용 가능

```
1 from ultralytics.models.sam import SAM3SemanticPredictor
2 from PIL import Image
3 import numpy as np
4
5 # Initialize predictor
6 overrides = dict(conf=0.25, task="segment", mode="predict", model="sam3.pt", half=True, save=True)
7 predictor = SAM3SemanticPredictor(overrides=overrides)
8
9 # Set image
10 predictor.set_image("img.PNG")
11
12
13 # Multiple bounding boxes for different concepts
14 results = predictor(bboxes=[
15     [590,114,604,128]
16 ])
17
```

```
1 from ultralytics.models.sam import SAM3SemanticPredictor
2
3 # Predictor 초기화
4 overrides = dict(conf=0.25, task="segment", mode="predict", model="sam3.pt", save=True)
5 predictor = SAM3SemanticPredictor(overrides=overrides)
6
7 # 이미지 설정
8 predictor.set_image("path/to/image.jpg")
9
10 results = predictor(
11     text=["person with a hat"],
12     bboxes=[[100, 150, 300, 500]]
13 )
```

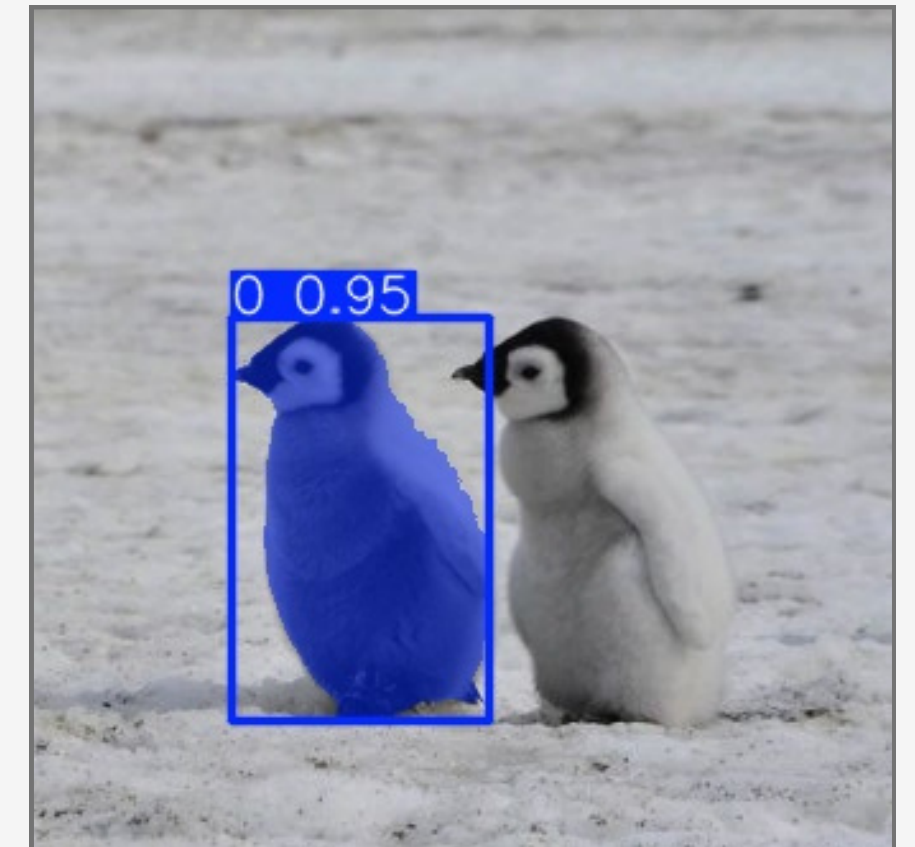
# SAM3

Promptable Visual Segmentation

**Method** Point를 이용하여 특정 객체만 segmentation

**Example** 왼쪽 펭귄 point 지정해주면 왼쪽만 segmentation 진행

```
1 from ultralytics import SAM
2
3 # SAM 3 모델 로드 (가중치 파일이 작업 디렉토리에 있어야 함)
4 model = SAM("sam3.pt")
5
6 # 포인트 프롬프트를 사용하여 세그멘테이션 수행
7 # points: [x, y] 좌표, labels: 1 (긍정 포인트)
8 results = model.predict(source="image.png", points=[[119,194]], labels=[1])
9
10 # 결과 시각화 및 저장
11 results[0].show()
12 results[0].save(filename="segmentation_result.jpg")
13
```



# Result

Text Prompt



Prompt

```
results = predictor(text=["particles"])
```

Result

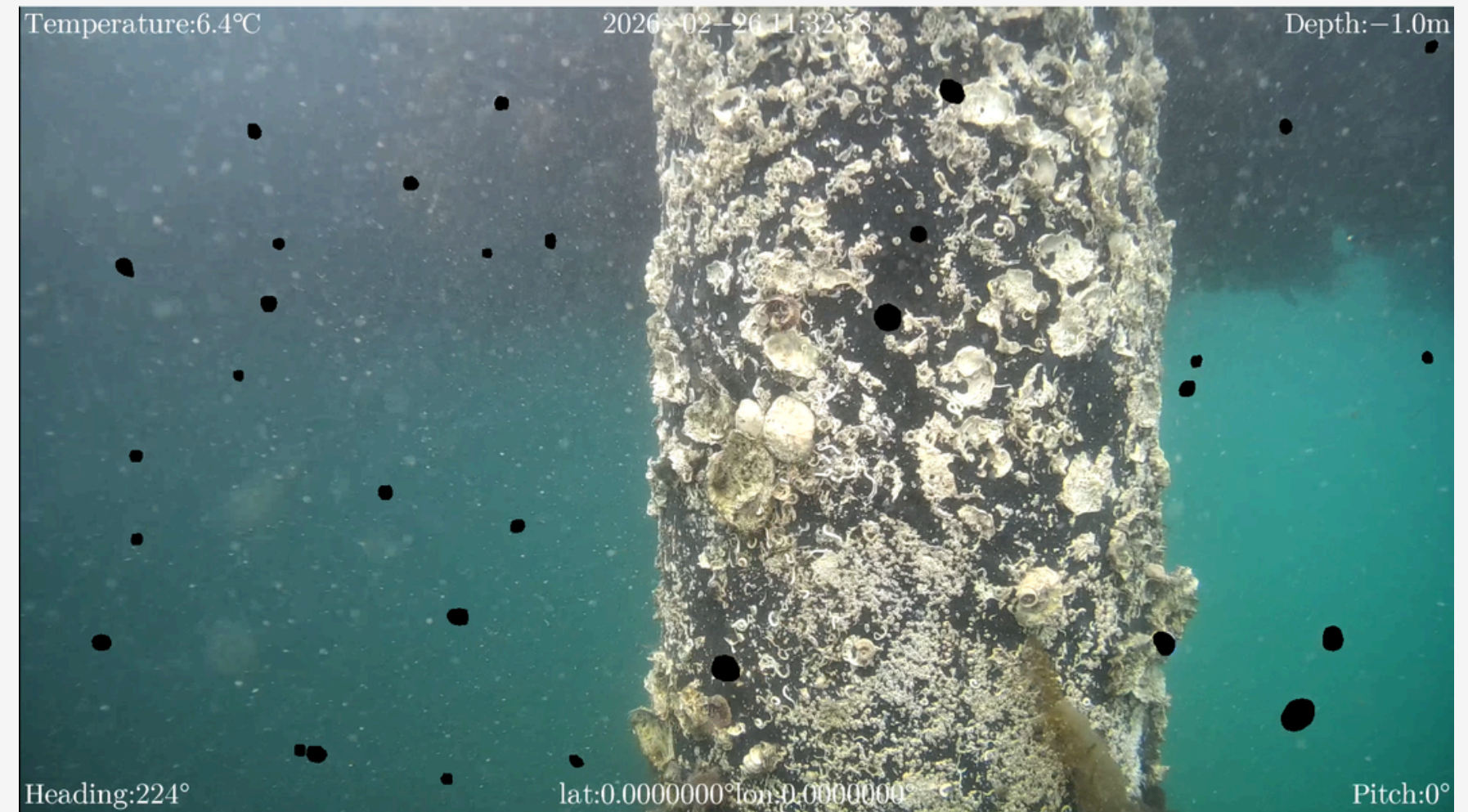
탐지한 객체가 없음

Fail Factor

“particle” prompt는 형태가 불분명하고 희미하기 때문에 text prompt 만으로는 탐지하기 어려움

# Result

bbox Prompt



Prompt

```
results = predictor(bboxes=[  
    [590,114,604,128],  
    [300,359,316,375],  
    [688,931,698,941],  
    [1564,140,1574,150]  
])
```

Result 31개 객체 탐지

# Result

bbox Prompt

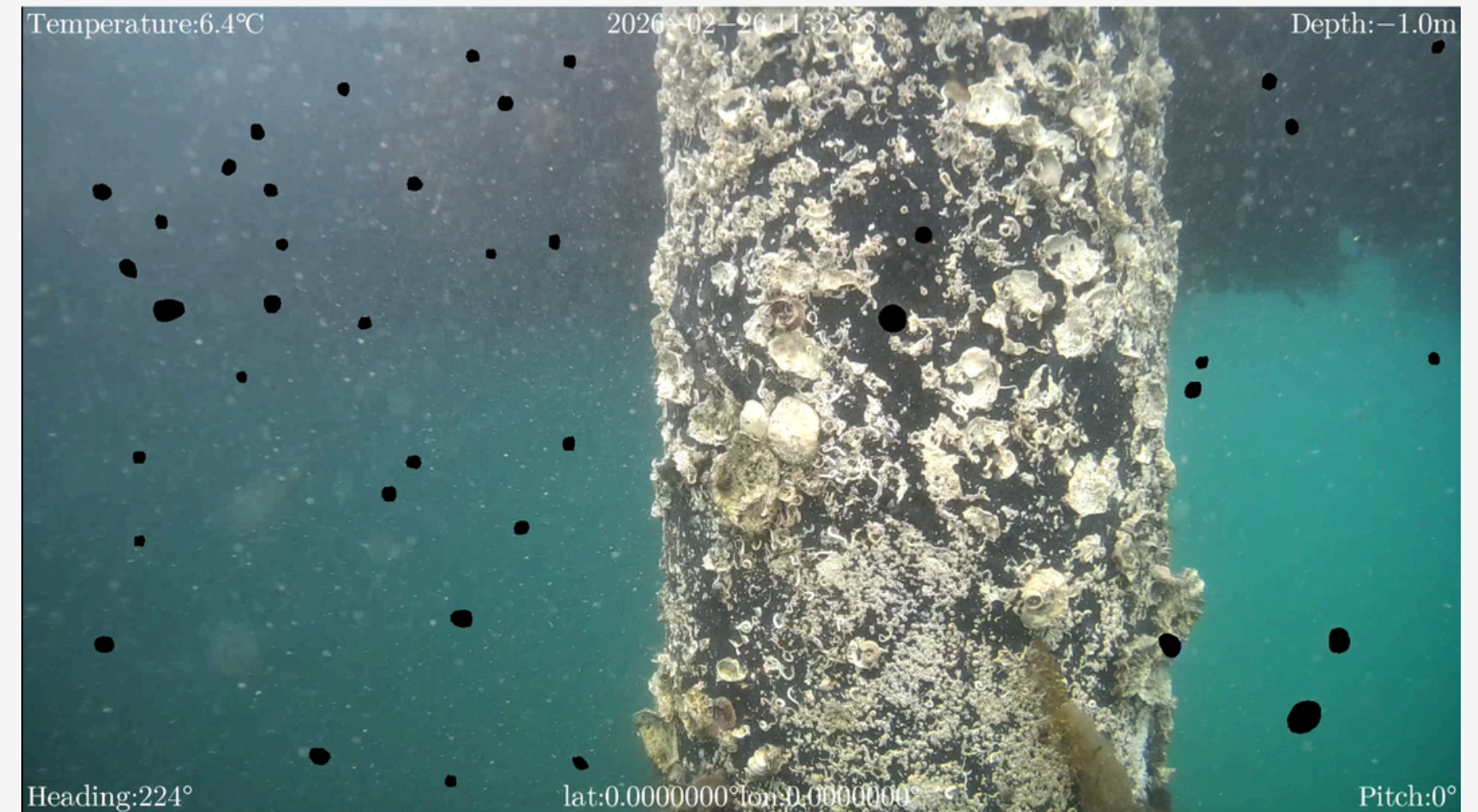
Prompt

```
results = predictor(bboxes=[  
  [590,114,604,128],  
  [300,359,316,375],  
  [688,931,698,941],  
  [1564,140,1574,150],  
  [1455,432,1465,442],  
  [1618,769,1648,799],  
  [277,145,302,168],  
  [299,354,321,380],  
  [167,258,178,273],  
  [124,313,139,335],  
  [589,114,608,124],  
  [674,59,686,81],  
  [481,210,494,225]  
])
```

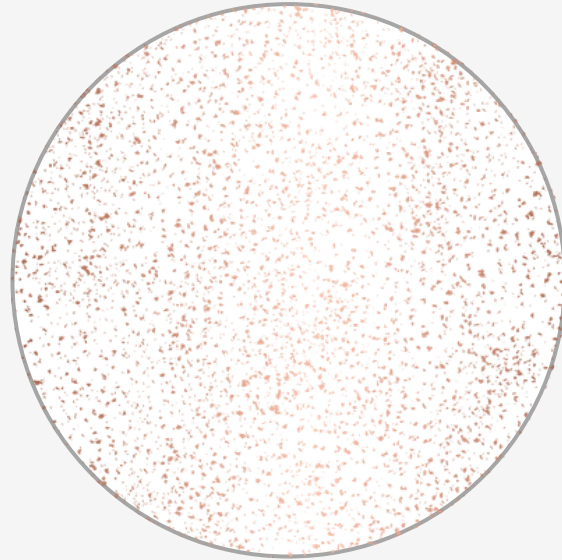
Result

약 10개의 부유물 추가 탐지, 오탐 약 2개 감소

추가된 bbox 갯수랑 비교시 큰 변화는 없음



# Limit



## 작은 객체 탐지

희미하거나 너무 작은 객체는 탐지 정확도가 낮음



## 불분명한 class

부유물이라는 특성상 따개비, 빛번짐등과 같은 노이즈에 약함



## Open-vocabulary

Open-voca의 장점은 범용성이지만, colosed-voca에 비해 전문성이 떨어져 부유물 탐지 작업같은 특수한 객체 탐지에는 오히려 약함



## FineTune 제한

SAM3는 FineTuning 하기에 적합한 모델이 아님

# Limit

## Hard Finetuning Factor

### Zero-shot

단순히 개, 고양이를 학습한게 아니라, “노란색옷을 입은 사람” 같은 복잡한 개념을 이해하도록 설계되어 소규모 데이터로 튜닝하면 모델의 기존 학습 체계가 깨져버릴 수 있음

### Multimodal

텍스트와 이미지를 하나의 공간에서 매핑시켰기 때문에 단순히 이미지 데이터셋으로만 학습시키면 이미지와 텍스트의 연관성에 대한 개념이 비틀어질 수 있음

### Hard Negatives

단순히 객체를 찾는 법 뿐만 아니라, 비슷한 오답을 걸러내는 방법을 같이 학습하여, 부정적 예시가 부족한 일반적 데이터 셋으로 학습할 시 모델의 변별력이 오히려 저하될 수 있음

# PLAN

1

## Sample image collect

수중드론 채집 영상을 일정 간격으로 프레임 추출

2

## Particle Polygon Labeling

직접 부유물의 Polygon을 따서  
DataSet 수집

Tool : Roboflow

3

## Model Finetuning

FineTuning이 가능한 모델에  
수집한 DataSet으로 FineTuning

# PLAN

Roboflow

AI를 이용해 클릭으로 Labeling 가능

## Particle num per Image

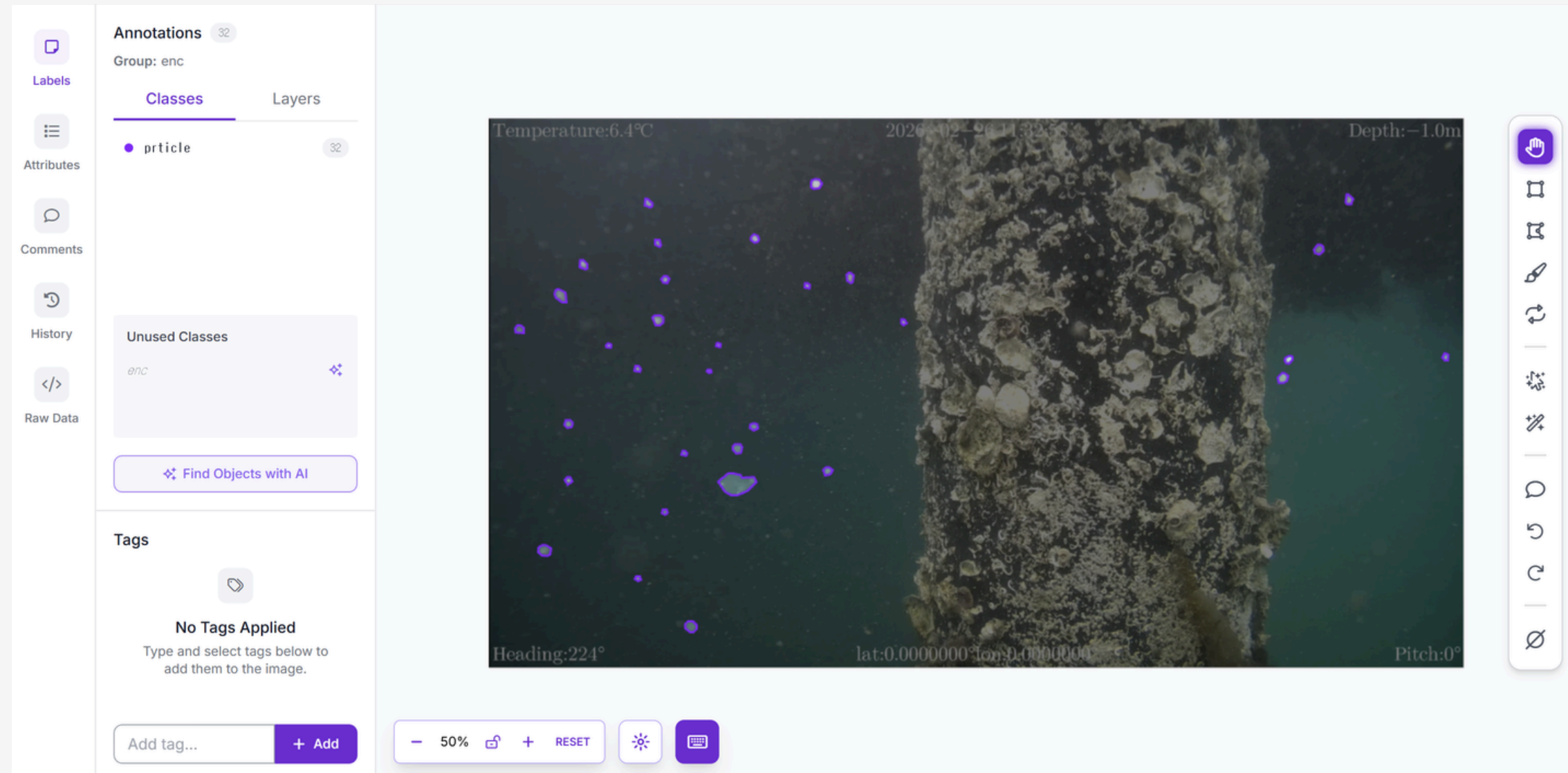
사람눈에 확실한 particle의 Poligon 가능한 많이 Labeling  
Image당 최소 100개 예상

## Number of Image

수집한 영상 총합 약 30분  
2초 간격으로 Frame 수집  
4K 화질(2160, 3840, 3) 이므로 학습시 GPU 메모리 부족할 수 있기 때문에 타일링 작업 필요  
전처리할 Image 갯수 : 약 900 \* N

## Credit Usage

5000 Image per Credit  
100 AI-Labeling per Credit



# PLAN

Roboflow

DataSet을 프로그램 내에서 Train, Test, Valid Set으로 분리 가능

DataSet 처리와 설정이 완료되면 모델별 학습 가능한 형태로 다운로드 가능

DataSet 처리가 완료되면 모델별 학습 가능한 형태로 다운로드 가능

### Add Images To Dataset

Add 12 images to dataset

[? What's Train, Valid, Test?](#)

Method

Split Images Between Train/Valid/Test

**Train** 78%    **Valid** 22%    **Test** 0%

Image Distribution

Train: 9 images  
Valid: 3 images  
Test: 0 images

You are about to add 12 images to the dataset

0 images will be sent back as part of a new job

**Add Images**

COCO Segmentation

SAM 2

YOLOv5 Oriented Bounding Boxes

YOLOv8 Oriented Bounding Boxes

Convert to Object Detection

COCO

COCO-MMDetection

CreateML

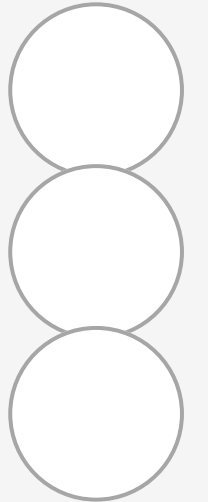
**COCO Segmentation**

COCO JSON with polygon segmentation masks.

Cancel

**Export as**

# THANK YOU



**박정규**

DeepSharkLab\_학부연구생