

# 9주차 보고서

바운딩박스 좌표(홈플레이트&타자)이용해서 스크라이크존 그리기

```
# 1. YOLO 설치
%cd /content
!git clone https://github.com/ultralytics/ultralytics

%cd /content/ultralytics
!pip install -e .
!pip install roboflow
```

```
# 2. Roboflow 라이브러리 설정 및 데이터셋 다운로드
from roboflow import Roboflow
rf = Roboflow(api_key="JX9qbDijsZEqknj2AfLJ")
project = rf.workspace("chaesubin").project("batter-and-home-
version = project.version(1)
dataset = version.download("yolo11") # yolo11 형식으로 데이터셋
```

```
# 3. YOLOv11n 모델 로드 및 사용
from ultralytics import YOLO
```

```
# YOLOv11n 모델 불러오기 (적절한 모델 가중치 선택, 예: 'yolov8n.pt',
model = YOLO("yolo11n.pt")
```

```
# if 학습시켜야 할 때
model.train(data='/content/ultralytics/batter-and-home-plate-
            imgsz=640,
            batch=32,
            epochs=300,
            name='ball_yolo11_result')

# else imodel 있는 거 쓸 때 (가중치 파일 있을 때)
model = YOLO('/content/ultralytics/yolo11_best.pt') # 경로는 예
```

```

import cv2

# 입력 및 출력 동영상 경로 설정
input_video_path = '/content/baseball.mp4'
output_video_path = '/content/result_baseball_home&batter_yol

# 동영상 파일 로드
cap = cv2.VideoCapture(input_video_path)

# 동영상 파일의 속성 가져오기
fps = int(cap.get(cv2.CAP_PROP_FPS))
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

# 결과를 저장할 VideoWriter 설정
fourcc = cv2.VideoWriter_fourcc(*'mp4v') # 코덱 설정
out = cv2.VideoWriter(output_video_path, fourcc, fps, (width,

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # 모델로 프레임에 대한 예측 수행
    results = model(frame)

    # 프레임에 예측 결과 그리기
    annotated_frame = results[0].plot() # 예측 결과 그리기

    # 그림 그리기
    # cv2.rectangle(image, (x1, y1), (x2, y2), (RGB), 선두께)
    cv2.rectangle(annotated_frame, (1000, 1000), (500, 500),

        # bounding box 좌표 print
        for obj in results[0].boxes.xyxy:
            print(obj)

    # 결과 프레임 저장

```

```

        out.write(annotated_frame)

# 자원 해제
cap.release()
out.release()

```

이제 **홈플레이트와 타자의 바운딩 박스 좌표를 기반으로 스트라이크 존을 그리는 단계**를 진행할 차례

```

# model 불러와서 사용하기
model = YOLO('/content/drive/MyDrive/Colab Notebooks/yolo11_b

import cv2

# 입력 및 출력 동영상 경로 설정
input_video_path = '/content/drive/MyDrive/Colab Notebooks/ba
output_video_path = '/content/resultdaima.mp4'

# 동영상 파일 로드
cap = cv2.VideoCapture(input_video_path)

# 동영상 파일의 속성 가져오기
fps = int(cap.get(cv2.CAP_PROP_FPS))
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

# 결과를 저장할 VideoWriter 설정
fourcc = cv2.VideoWriter_fourcc(*'mp4v') # 코덱 설정
out = cv2.VideoWriter(output_video_path, fourcc, fps, (width,

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # 모델로 프레임에 대한 예측 수행

```

```

results = model(frame)

# 프레임에 예측 결과 그리기
annotated_frame = results[0].plot() # 예측 결과 그리기

print(annotated_frame)
# 그림 그리기

for obj in results[0].boxes.xyxy:
    print(obj)
    annotated_frame = cv2.rectangle(annotated_frame, (int(

#cv2.rectangle(annotated_frame, (1000, 1000), (500, 500),

# 결과 프레임 저장
out.write(annotated_frame)

# 자원 해제
cap.release()
out.release()

```

