Safezone

11 June 2024

CloseAI 이상헌 김유철 박준혁 이정훈

목차 table of contents

1

Overview

프로젝트 목표

특정 상황에서 안전 관리에 도움을 주는 모델 개발

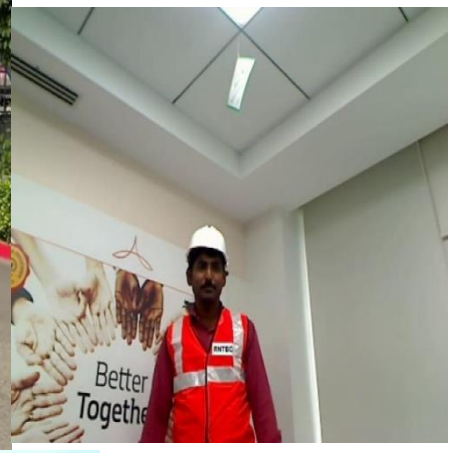
사전에 정의한 각종 위험한 상황을 탐지 및 경고

관리자가 모니터링 하고 싶은 객체를 입력 받아 탐지

2

Data

활용 데이터



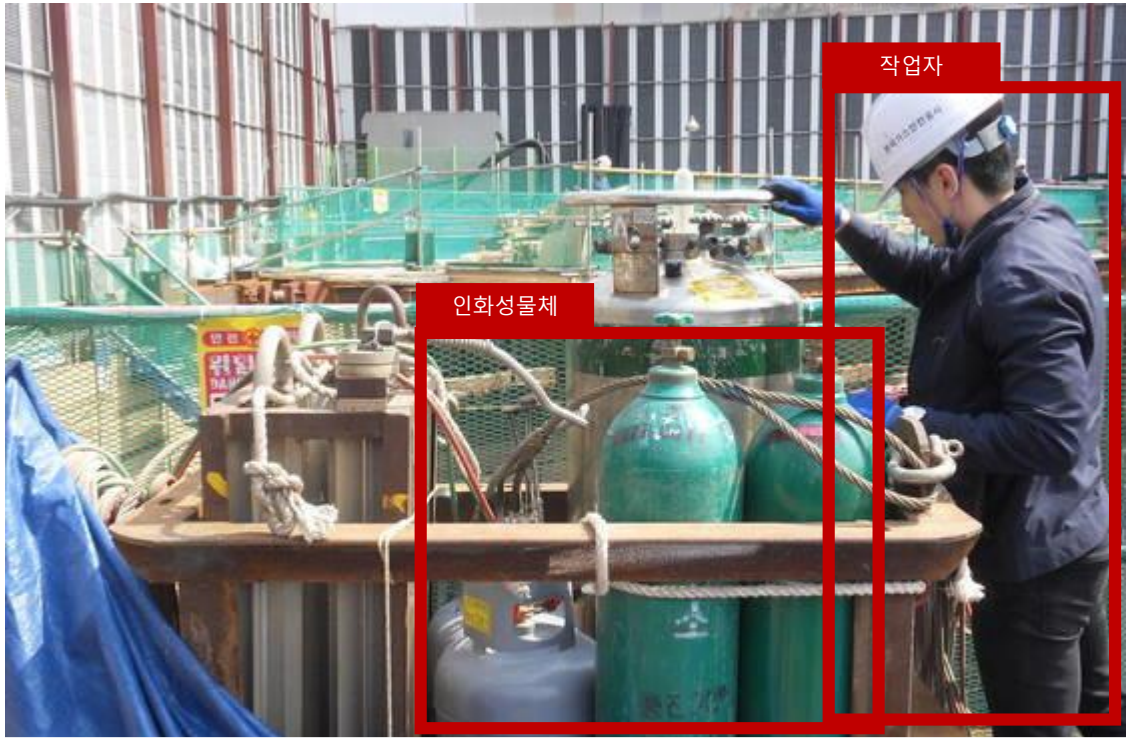
Roboflow 데이터셋 수집
AI허브 안전장비 데이터셋
자체 데이터셋

출처:https://universe.roboflow.com/search?q=construction+images%3E5000+model%3Ayolov8

3

Model

Case 1



사전에 정의한 특정 이벤트 발생시 경고

문자 수신
<인화성 물질에 작업자 접근. 확인 요망>

Case 2

사용자가 원하는 특정 상황을 입력 받아 탐지 가능

예시) 입력 : 안전모를 착용하지 않은 사람을 찾아줘

YOLO v8

모델 선정 이유

1. 높은 정확도와 속도
2. 다양한 객체 탐지
3. 간편한 학습 및 튜닝
4. 고해상도 이미지 처리

다른 모델과 비교

YOLO 버전 비교

버전	구조와 효율성	성능
YOLO6	경량화와 속도 최적화에 중점	작은 객체 탐지에서 강점, 빠른 속도
YOLO7	효율적 구조와 훈련기법으로 성능 향상	다양한 벤치마크에서 높은 성능
YOLO8	최신 기술 통합과 모듈화로 성능 극대화	최신 기술 적용으로 최고성능

LangChain

- LangChain은 대규모 언어 모델(LLM)을 활용하여 자연어 처리 및 다양한 작업을 수행할 수 있는 프레임워크입니다.

- Prompt Template
- Few-Shot Prompt Template

LangChain: Prompt Template


```
from langchain.prompts import PromptTemplate, ChatPromptTemplate

#프롬프트 템플릿을 통해 매개변수 삽입 가능한 문자열로 변환
string_prompt = PromptTemplate.from_template("tell me a joke about {subject}")

#매개변수 삽입한 결과를 string_prompt_value에 할당
string_prompt_value = string_prompt.format_prompt(subject="soccer")

#채팅LLM이 아닌 LLM과 대화할 때 필요한 프롬프트 = string prompt
string_prompt_value
```

```
StringPromptValue(text='tell me a joke about soccer')
```

LangChain: Prompt Template


```
from langchain.prompts.prompt import PromptTemplate

template = """
너는 요리사야. 내가 가진 재료들을 갖고 만들 수 있는 요리를 추천하고, 그 요리의 레시피를 제시해줘.
내가 가진 재료는 아래와 같아.

<재료>
{재료}

"""
prompt_template = PromptTemplate(
    input_variables = ['재료'],
    template = template
)
```

```
print(prompt_template.format(재료 = '양파, 계란, 사과, 빵'))
```

```
너는 요리사야. 내가 가진 재료들을 갖고 만들 수 있는 요리를 추천하고, 그 요리의 레시피를 제시해줘.
내가 가진 재료는 아래와 같아.

<재료>
양파, 계란, 사과, 빵
```

LangChain: Few-Shot Prompt Template


```
from langchain.prompts.few_shot import FewShotPromptTemplate
from langchain.prompts.prompt import PromptTemplate

examples = [
  {
    "question": "아이유로 삼행시 만들어줘",
    "answer":
"""
아: 아이유는
이: 이런 강의를 들을 이
유: 유가 없다.
"""
  },

  {
    "question": "김민수로 삼행시 만들어줘",
    "answer":
"""
김: 김치는 맛있다
민: 민달팽이도 좋아하는 김치!
수: 수억을 줘도 김치는 내꺼!
"""
  }
]
```

```
example_prompt = PromptTemplate(input_variables=["question", "answer"], template="Question: {question}₩n{answer}")
```

LangChain: Few-Shot Prompt Template


```
prompt = FewShotPromptTemplate(
    examples=examples,
    example_prompt=example_prompt,
    suffix="Question: {input}",
    input_variables=["input"]
)
```

```
Question: 아이유로 삼행시 만들어줘

아: 아이유는
이: 이런 강의를 들을 이
유: 유가 없다.


Question: 김민수로 삼행시 만들어줘

김: 김치는 맛있다
민: 민달팽이도 좋아하는 김치!
수: 수억을 줘도 김치는 내꺼!
```

LangChain: Few-Shot Prompt Template


```
print(davinch3.predict("호날두로 삼행시 만들어줘"))
```

호날두 떠나면 혼자 늘 외로워
내 앞에는 다른 누구도 없는데
나는 가끔 슬플 때도 있어
그때는 네가 언제나 곁에 있던 걸

호날두 내 곁에 있던 기억이 나
어제 이렇게 같이 웃었던 걸
호날두 내게 넌 다시 오지 않을까
널 기다려 할 땐 마음 너무 힘들어

난 널 보고 싶어 널 보고 싶어
사랑하는 너를 다시 만나고 싶어
널 보면 마음이 흔들려 흔들려
잊지 못할 추억이 떠오르네

호날두 내 곁에 있던 기억이 나
어제 이렇게 같이 웃었던 걸
호날두 내게 넌 다시 오지 않을까
널 기다려 할 땐 마음 너무 힘들어

비가 오면 내 마음이 슬프게 변하네
나의 가슴 속에 널 찾아가네
난 널 보고 싶어 널 보고 싶어
사랑하는 너를 다시 만나고 싶어

난 널 보고 싶어 널 보고 싶어
사랑하는 너를 다시 만나고 싶어
널 보면 마음이 흔들려 흔들려
잊지 못할 추억이 떠오르네

내게 다시 너를 보내줘

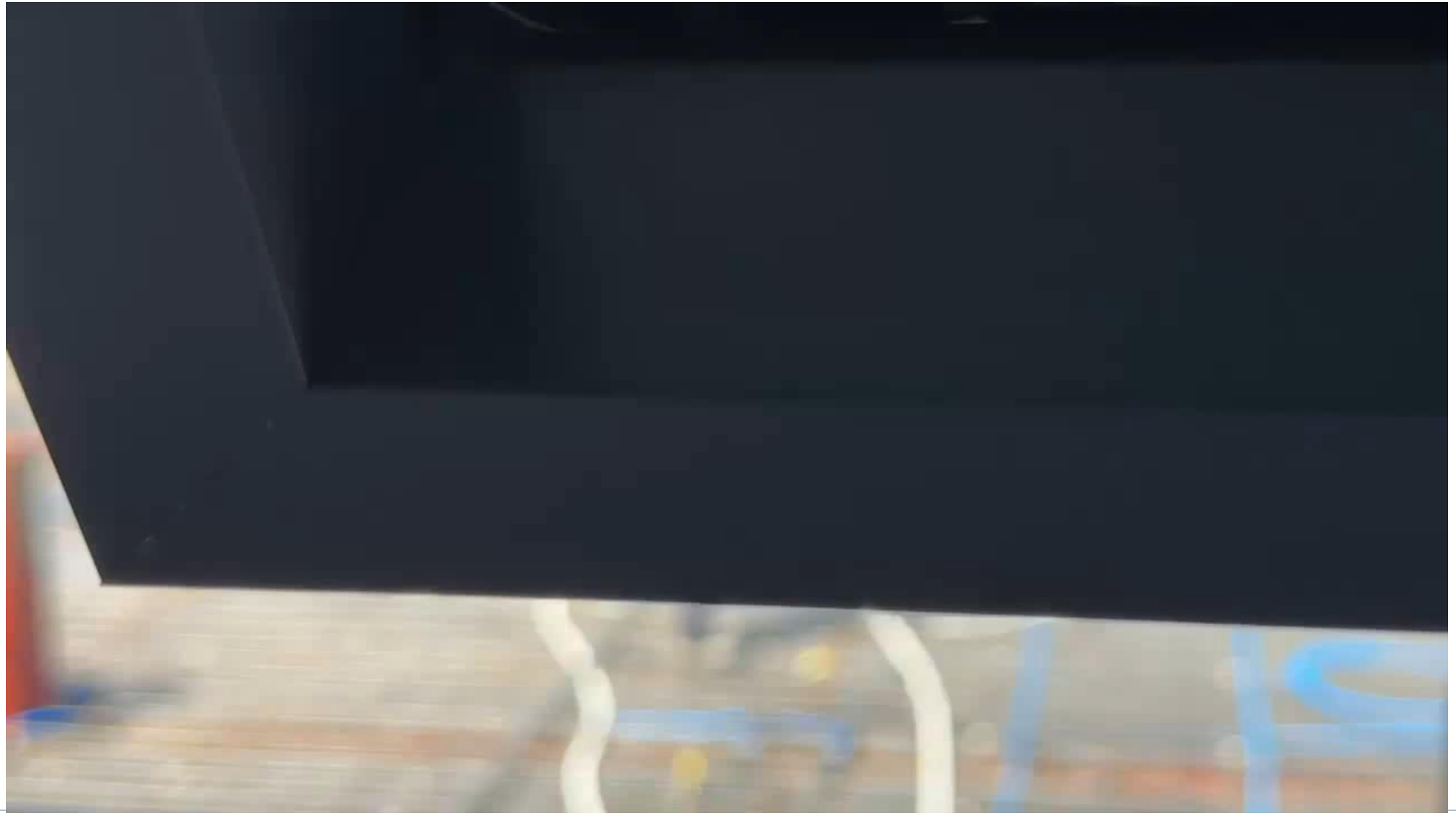
```
print(davinch3(
    prompt.format(input="호날두로 삼행시 만들어줘")
))
```

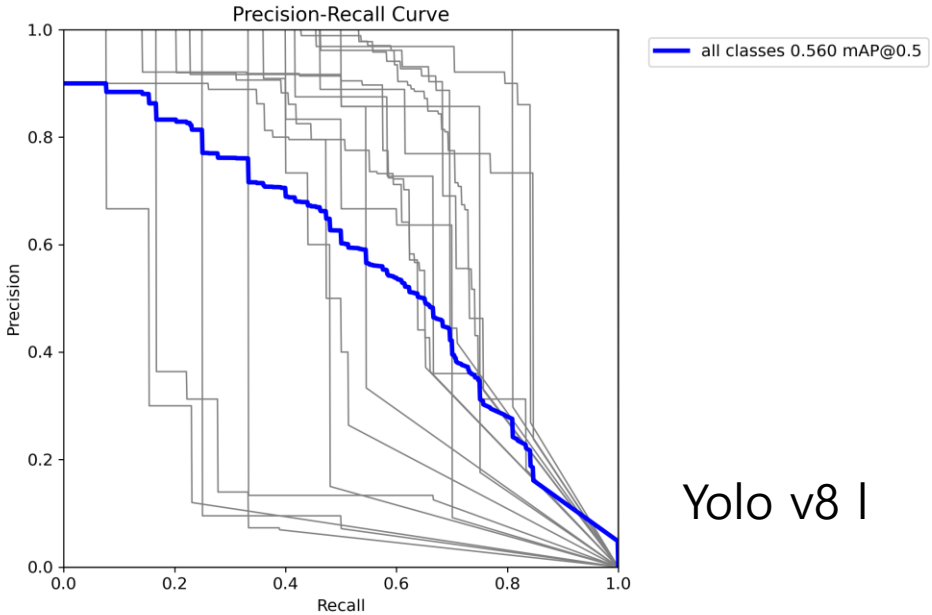
호: 호주에서 왔어
날: 날개가 펄럭이고
두: 두근두근 마음이 뛰어대는 날

4

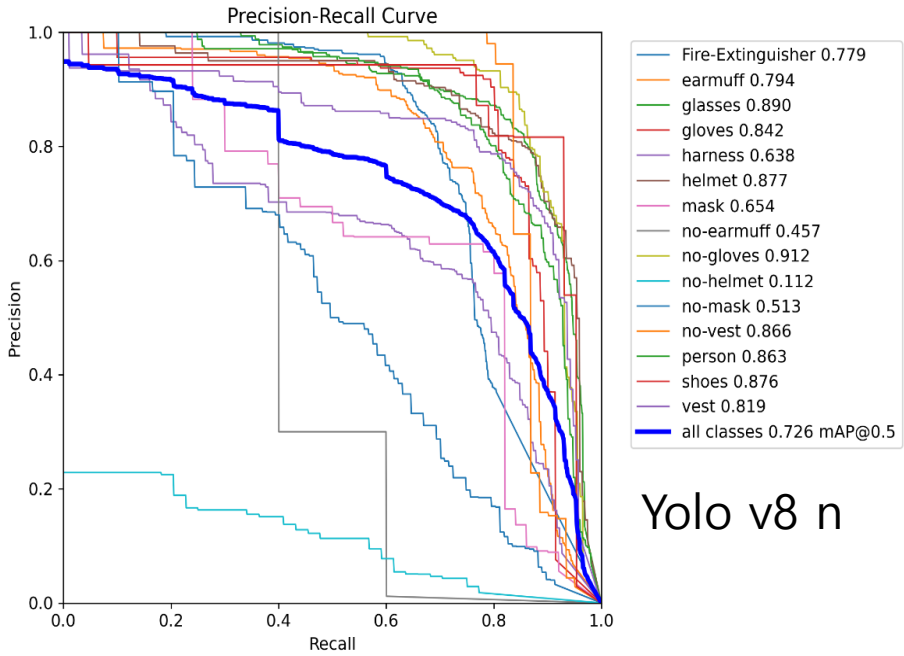
프토로타입

YOLO v8 구현 결과

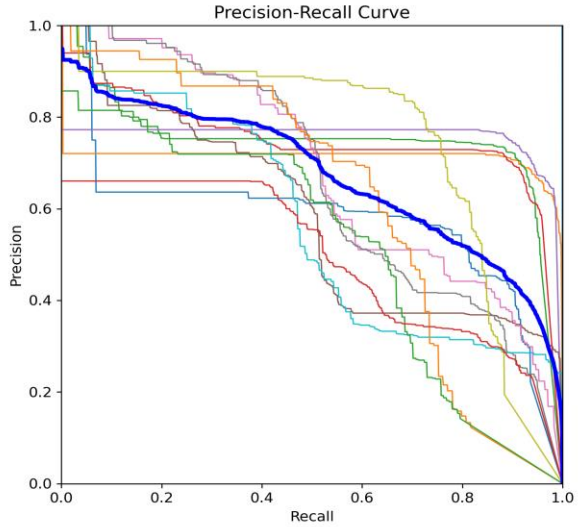




Yolo v8 l



Yolo v8 n



Yolo v8 m

LangChain 구현 결과


```
하이바미착용이 탐지되었습니다. 해당 객체의 좌표는 [115, 521], [388, 321], [434, 259]입니다.

하이바미착용이 여러 곳에서 탐지되었습니다. 해당 객체의 좌표는 [118, 553], [368, 331], [433, 255] 입니다.
```

5

핵심 코드

핵심 코드

```python
from ultralytics import YOLO



# Load a model
model = YOLO("yolov8l.yaml")  # build a new model from scratch
model = YOLO("yolov8l.pt")  # load a pretrained model (recommended for training)

# Use the model
model.train(data="data.yaml", epochs=500, batch=-1, patience=100)  # train the model
metrics = model.val()  # evaluate model performance on the validation set
# results = model("https://ultralytics.com/images/bus.jpg")  # predict on an image
# results = model("/home/saka/바탕화면/123.jpg")  # predict on an image

path = model.export(format="onnx")  # export the model to ONNX format
```

핵심 코드

```python
import cv2
from ultralytics import YOLO
import numpy as np
from PIL import ImageFont, ImageDraw, Image

# Load a model
model = YOLO("best.pt")  # load a pretrained model

# 비디오 파일 경로 설정
video_path = "/home/saka/문서/카카오톡 받은 파일/bbb.mp4"
output_path = "/home/saka/yolov8/output/annotated_video_123.mp4"

# 비디오 캡처 객체 생성
cap = cv2.VideoCapture(video_path)

# 비디오 저장 객체 설정
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
fps = cap.get(cv2.CAP_PROP_FPS)  # 원본 비디오의 프레임 속도 가져오기
slow_fps = fps / 2  # 비디오 속도를 절반으로 줄이기
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
out = cv2.VideoWriter(output_path, fourcc, slow_fps, (width, height))
```

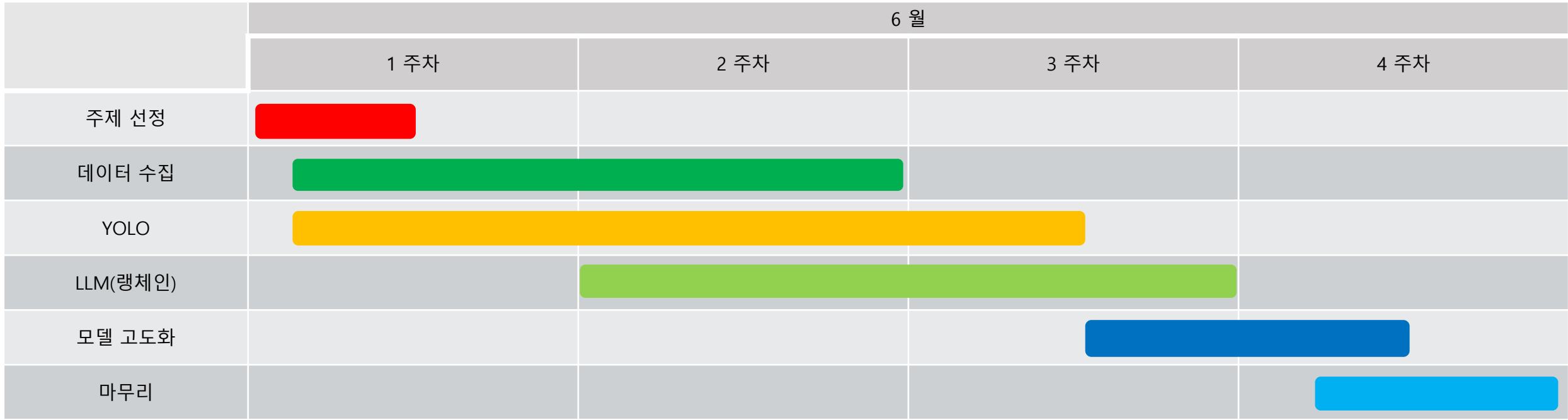
핵심 코드

```python
from langchain.chains import LLMChain
from langchain.schema import BaseOutputParser
import os



class CommaSeparatedListOutputParser(BaseOutputParser):
    """LLM 아웃풋에 있는 ','를 분리해서 리턴하는 파서."""
    def parse(self, text: str):
        return text.strip().split(", ")

template = """
너는 공사장에서 객체의 상태를 파악하고 객체의 위치를 알려주는 AI야.
여러가지 객체의 이름과 좌표를 입력할테니, 내가 요구한 <특정 객체>가 탐지되면 좌표를 알려줘.
탐지된 <특정 객체>가 여러개라면, 콤마(,)로 구분해서 알려줘

특정 객체:"""
system_message_prompt = SystemMessagePromptTemplate.from_template(template)
human_template = "{사용자가 찾는 객체}"
human_message_prompt = HumanMessagePromptTemplate.from_template(human_template)

chat_prompt = ChatPromptTemplate.from_messages([system_message_prompt, human_message_prompt])

chain = LLMChain(
    llm=ChatOpenAI(),
    prompt=chat_prompt,
    output_parser=CommaSeparatedListOutputParser()
)
chain.run("하이바미착용")
```

6

Time Line

Time Line



7

사용 기술

사용 기술


LangChain


TensorFlow


PyTorch


OpenCV


ultralytics
YOLOv8


scikit
learn


matplotlib


NumPy

8

Reference

Reference

Real-Time Flying Object Detection with YOLOv8 (Dillon Reis, 22 May 2024 https://arxiv.org/pdf/2305.09972**)**

You Only Look Once: Unified, Real-Time Object Detection (Joseph Redmon, 9 May 2016 https://arxiv.org/pdf/1506.02640**)**

9

QnA