

# Tree Flow

```

메인메뉴
├─ 회원가입
├─ 로그인
│   └─ 대시보드
│       ├── 사고발생 횟수 (Time Series Line Chart)
│       ├── 고정일과에 대한 To Do List
│       └─ └─ 매일 지정 시간마다 초기화 (Ex. 06시에 DB에 적재되면서 일자변경 후 초기
화)
│           ├── 미보고 일탈현황
│           ├── 일탈 선택시 보고서 양식 제공
│           ├── DB에 적재했던 일탈 세부 정보 제공
│           └─ 선택사항 ( 자동 로테이션 CCTV 화면 || 추가 아이디어 )
├─ 챗봇
│   ├── 임베딩된 사내문서 기반 검색기능
│   ├── ※ 페이지 내부 요소를 변경하는 방식으로 연결이 끊기지 않게
│   └─ CCTV : 문제 발생시 자동 활성화
├─ CCTV
│   ├── 객체탐지 (객체는 학습 가능한 객체만 선택)
│   ├── └─ 후보군(안전모 착용 유무 , 화재 , 누수 , 스팀)
│   └─ 이상현상 발생시 트리거
│       ├── ChatBot에 알림
│       ├── └─ {어떤화면(어디서), 어떤문제가, 언제, 누가(기계, 사람)}
│       ├── DB에 적재
│       └─ Drag&Drop으로 CCTV배치 편집 가능
└─ 게시판
    └─ 사내문서 조회용 게시판
  
```

## 객체 탐지 기술

### 전체 흐름

( 처리순서 )

#### 1. 프레임 캡처

- 카메라(웹캠, RTSP 스트림 등)에서 실시간으로 한 프레임을 가져온다.
- cv2.VideoCapture를 사용하거나, RTSP를 통해 영상 스트림을 읽는다.

#### 2. 객체 탐지

- 가져온 프레임을 YOLO 모델에 입력한다.
- YOLO는 객체의 위치(바운딩 박스), 클래스 정보(이름), 신뢰도를 반환한다.

#### 3. 탐지 결과 오버레이

- YOLO의 리턴값(바운딩 박스, 클래스 이름, 신뢰도)을 활용해 프레임에 정보를 오버레이한다:
  - 바운딩 박스 그리기: `cv2.rectangle`
  - 클래스 이름과 신뢰도 표시: `cv2.putText`

#### 4. 수정된 프레임 출력

- 수정된 프레임을 `cv2.imshow`를 통해 화면에 실시간으로 출력한다.

#### 5. 동작 방식 요약

1. 클라이언트에서 영상 캡처 → 서버로 전송.
2. 서버에서 YOLO 모델로 객체 탐지 수행.
3. 탐지 결과를 오버레이하여 수정된 프레임 생성.
4. 서버에서 수정된 프레임을 클라이언트로 반환하거나, 바로 화면에 출력.

## 챗봇 기술

### 전체 흐름

(처리순서)

아직 몰라요

### < Yolo-질문 >

Q1. YoloX 혹은 어떤버전? 어떤 버전을 쓰든 weight종류 많던데 어떤걸로 할까? 각각 다른걸 써서 합쳐야 할지? 아닐지 합친다면 어떻게 합칠지?

A1. 객체탐지를 잘 하는 모델일 뿐. 내가 정하면 되는거임 ~논리적으로 가장 잘 맞는 것

Q2. 유튜브 영상이나 실제 라이터 같은걸로 테스트 해야 하는지? 유튜브로 하게 된다면 어떻게 해야될지 기반이 잘 안잡혀 있긴해서 ( 화재 알람같은걸 어떤식으로 줘야할지 몇초마다 한번씩? 그런다고 한번 났다고 한번 보내고 끝?)

A2. 제일 빠르고 좋은 방법으로

Q3. 영상학습도 프레임단위로하고, 실제 객체탐지 받는것도 프레임 단위로 하는게 맞나요? - 사진으로도 학습하고 영상으로도 학습하는것 같던데...

A3. (전처리과정이 중요하다 사이즈나 이런걸 잘 맞춰야 되기 때문) 어차피 프레임 단위로 진행된다

Q4. 안전모 착용 데이터셋 관련 - 우리데이터로 이미 안전모 착용/미착용 상태 이미지를 학습한 best.pt를 검증하고 전이학습처럼 사용해도 될지? 아니면 raw data를 새로 학습한다음 test하는게 더 좋을까요?

A4. 사용하는게 좋아요 재학습을 통한 파인튜닝~

Q5. 전체 흐름에 대한 이해(이게 맞나요?)

- 객체(화재, 안전모) 이미지를 학습(모델 적용 , 학습 실행)
- > 그 결과가 best.pt, last.pt로 저장
  - > 실시간 또는 저장된 영상 데이터를 video\_capture 로 가져와 프레임 단위로 처리
  - > 해당 이미지 데이터들에 best.pt, 적용하여 test
  - > 해당 결과값(class\_id)에 Class\_name 매핑
  - > bounding box에 confidence랑 객체명이 띄워지면서 해당 객체가 리턴됨
  - > 사용자 인터페이스에서 감지 결과 실시간 보여줌
  - > 함수로 후속 로직 수행

A5. 인풋에 putText, rectangle 으로 집어 넣어서

Q6. 몇프레임단위로 보내야될지

A6. 엔지니어링 기법이 들어가야함. 성능은 나와야되고 처리는 빨라야함.

### < 챗봇 질문 >

Q1. chromaDB (LangChain이나 임베딩 데이터 처리속도 문제로 크로마DB사용 예정)

- 현재 DB의 정보를 어떻게 확인해야 할까?
  - 왜냐면 벡터화해서 정보를 적재만 하다보니 잘 하는지 신뢰가 안간다
- 차원에 관해서 걱정이 많음. 의미가 있는 단어의 갯수로 차원 수를 정한다고 하는데 이해가 잘 안간다. 벡터화에 대한 이해가 필요한 것 같기도 하다

A1. 어차피 티안남

Q2. 구현하고자 하는 목표는 뚜렷하나 기술적인 흐름이 감이 안옴

사용자 질의 - 질의 벡터화 - DB에서 유사도 기반으로 추출 - LangChain&RAG 답변생성 - 사용자에게 전송

으로 이해는 하지만 LangChain과 RAG을 기반으로 챗봇이 작용하는 방식이 이해가 부족한듯 하다. 처리 흐름이나 개별 기능&사용법이 궁금하다.

A2. 질의를 벡터화해서 유사도를 기반으로 정답을 뽑아냄, 프롬프트를 기반으로 RAG으로 말투로 나눔

#### < 공통 질문 >

Q1. 두 기능 다 파이썬에서 돌아가는 기능인데, 웹과의 인터랙션은 자바에서 구현할 계획이다. 그렇다면 파이썬으로 어떻게 분석이나 질의 기능을 실행하는가?

Ubicorn- 서버역할 - (FAST API : 함수를 웹 어플리케이션으로 만들어짐.):웹 어플리케이션 톰캣 - 서버역할 - (Spring : 웹어플리케이션으로 만들어주는거거) 함수형태로 돌려야댐!