포인트

1. 각종 버그 픽스 및 기존 코드 리팩토링
   1. 서적 기반으로 테스트코드
   2. MVC 패턴 기반 + 싱글톤 패턴화
2. 웹서버 연동으로 사용 가능한 서비스로 디벨롭

2-1. 사진 업로드를 통해 모델 사용, 디텍션 하도록

2-2. 업로드 된 사진으로 모델 추가 학습 가능

1. 보안 이슈 컨트롤

3-1. 트래픽 컨트롤 디도스 예시 같은거 만들어서 서버에서 과트래픽 어떻게 처리할 수 있을지

3-2. CSRF 방지 방법

진행사항 25/3/9

모델 생성 코드, 데이터 있음.

개발

FE: 기능은 2-에 따라서 최소 기능 FE 구현

BE:

1. 모델 사용 API
2. 모델 재학습 API

쓰는 기술

Django, Black

UI /Done 25.03.09?

DB / 이거 필요함?

모델 /Base model 25.03.10 / 이슈! 03/12 모델 출력 시 Class가 동적이어야함. 현재 10갠데 N개로 추가학습 하면 어떻게 처리해서 구현할것인가

여기까지 나온 API 업로드

모델 재학습 알고리즘

재학습 API 업로드

\*Rest API를 사용하지 않은 전통적인 Django 방식.

\*웹뷰는 가능하면 HTML 로 간단하게 구현함

\*이 프로젝트의 포인트는 인공지능과 서비스의 혼합적 개발이 가능함

\*가장 기본적인 부분

알게된 것

CSRF 토큰 이슈 : <https://codevang.tistory.com/282>

해결 방법: **✅ 1. HTML에 CSRF 토큰 추가**

html

복사편집

<meta name="csrf-token" content="{{ csrf\_token }}">

**✅ 2. JavaScript 수정**

javascript

복사편집

// CSRF 토큰 가져오는 함수

function getCSRFToken() {

return document.querySelector('meta[name="csrf-token"]').content;

}

// 파일을 서버로 전송하는 공통 함수

function sendImage(endpoint) {

const fileInput = document.getElementById("fileInput");

if (fileInput.files.length === 0) {

alert("먼저 이미지를 업로드하세요!");

return;

}

const formData = new FormData();

formData.append("image", fileInput.files[0]);

fetch(endpoint, {

method: "POST",

body: formData,

headers: {

"X-CSRFToken": getCSRFToken() // CSRF 토큰 추가

}

})

.then(response => response.json())

.then(data => alert(data.message || "오류 발생"))

.catch(error => console.error("Error:", error));

}

**✅ 3. Django 뷰 (views.py)에서 CSRF 확인**

뷰 함수 위에 @csrf\_exempt를 붙이지 않으면 CSRF 토큰을 요구하게 돼.

\*\* 좀더 확실하게 설계할 필요가 있음

0313 – Utils가 4종류 있는데 (ImageHandle, ModelHandle, SearchUtils, LearnUtils) 이게 서로 독립성 유지가 쉽지 않음

일단 지금 생각은 ImageHandle과 ModelHandle은 독립적으로 가는데

SearchUtils랑 LearnUtils에서 Image 처리는 ImageHandle에서 Model은 ModleHandle에서 가져오게 하고 싶음.

근데 이게 안되면 Handler로 하나로 합친다음에 독립성을 유지하는 방법이 있음