

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	최적화된 옥외 광고를 위한 광고 효과 분석 솔루션
팀 명	DRM
문서 제목	캡스톤 디자인I 종합설계 프로젝트 4조 중간보고서

Version	1.5
Date	2024-03-31

팀원	이동국 (20192282) (조장)
	허수빈 (20201320)
	이강희 (20181653)
	황솔희(20203168)
	최지원(20213091)
	장수미(20193216)
지도교수	김형균 교수

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 Do you READ me?를 수행하는 팀 DRM의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 DRM의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	중간보고서-최적화된 옥외 광고를 위한 광고 효과 분석 솔루션.doc
원안작성자	이동국, 이강희, 장수미, 최지원, 허수빈, 황솔희
수정작업자	최지원, 허수빈

수정날짜	대표 수정자	Revision	추가/수정 항목	내 용
2024-03-26	장수미	1.0	최초 작성	프로젝트 개요 작성
2024-03-27	이동국	1.1	추가	수행 내용 및 중간 결과 작성
2024-03-28	이강희	1.2	추가	AI 파트 작성
2024-03-29	황솔희	1.3	추가	모델 서버 및 아키텍처 파트 작성
2024-03-30	최지원	1.4	수정	각 파트 별 문장 수정 및 이미지 첨부
2024-03-30	허수빈	1.5	수정	밀첨자 추가 및 출처 기재, 맞춤법 검사

목 차

1. 프로젝트 개요
 - 1.1. 프로젝트 소개
 - 1.2. 서비스 소개
 - 1.2.1. 서비스 주요 기능
 - 1.3. 서비스 목표
 - 1.4. 기대 효과 및 활용 방안

2. 수행 내용 및 중간 결과
 - 2.1. 초기 계획
 - 2.2. 변경 사항
 - 2.2.1. 변경 사유
 - 2.2.2. 변경 사항
 - 2.3. 수행 내용 및 추진 사항
 - 2.3.1. 프로젝트 전체
 - 2.3.2.
 - 2.3.3. 서버
 - 2.3.4. 모델

3. 향후 추진 계획
 - 3.1. 프로젝트 전체
 - 3.2. 클라이언트
 - 3.3. 서버
 - 3.4. 모델

4. 참고 자료

1. 프로젝트 개요

1.1. 프로젝트 소개

광고주는 자신의 광고가 실제로 어떤 효과를 내는지 알고 싶어 한다. 온라인 광고처럼 오프라인 옥외 광고도 분석이 가능하다면 광고주는 더욱 효율적으로 광고의 효과를 알 수 있을 것이다. 또한 광고주는 소비자들이 정말로 자신의 광고를 관심 있게 보는지에 대한 의문이 있다. 이러한 니즈를 충족시키기 위해 오프라인 광고 분석 서비스 *Do you READ me*의 개발을 시작하였다.

팀 DRM은 2세대 광고 대행사를 만들고자 한다. 기존의 광고대행사들은 단순히 디스플레이에 광고를 띄우는 데 그치는데, 이는 광고의 효과를 정확히 파악하는 것이 불가능하다. 우리는 이러한 한계를 극복하고, 광고 송출에 광고 분석 서비스를 추가하여 오프라인 광고의 실제 효과를 정확히 측정하고자 한다.

서비스명 *Do you READ me?*에서 사용되는 *read*는 사전적 의미로, 단어나 상징을 보고 그 의미를 이해하는 것(*to look at words or symbols and understand what they mean*)을 의미한다. 서비스에서 *read*는 두 가지 의미를 갖는다. 첫 번째, 서비스는 소비자가 광고를 실제로 인지하고 주목하는지를 알려준다. 두 번째, 서비스는 광고주가 광고의 효과를 제대로 이해하도록 대시보드를 제공한다. *Do you READ me?*를 통해 사용자는 온라인 광고뿐만 아니라 오프라인 옥외 광고도 광고 분석 결과를 제공받을 수 있다. 서비스의 가치는 *read*의 의미와 일치한다고 말할 수 있다. 프로젝트의 소스 코드와 진행 상황은 해당 레포지토리([kookmin-sw/capstone-2024-04](https://github.com/kookmin-sw/capstone-2024-04))에서 확인할 수 있다.

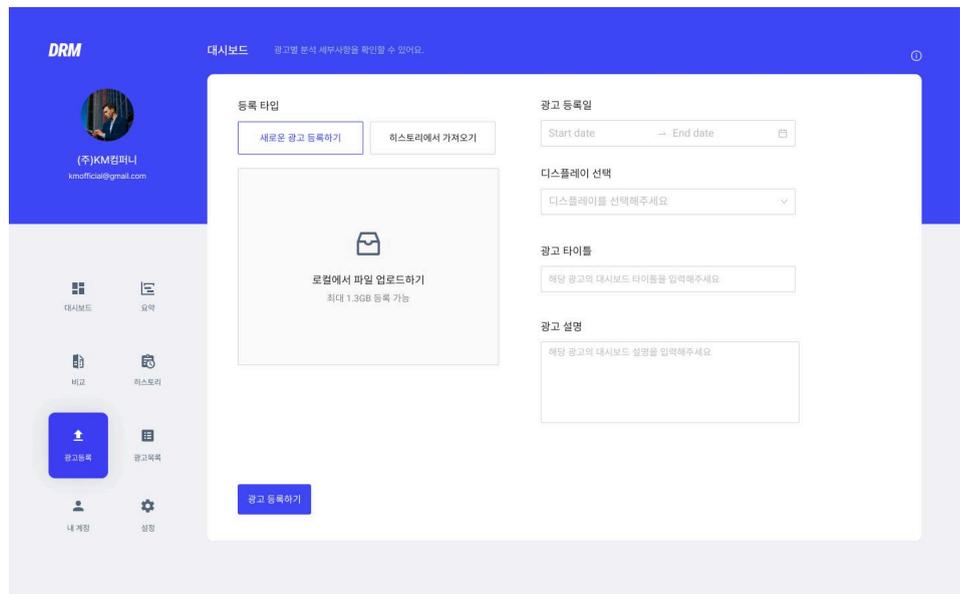
1.2. 서비스 소개

*Do you READ me*는 데이터 분석을 통해 옥외 광고의 효율성을 극대화하는 웹 서비스이다. 해당 서비스는 디스플레이 근처의 유동 인구와 타겟층 정보를 분석하여 광고주가 시각적으로 쉽게 이해할 수 있는 대시보드를 제공한다. 광고주는 대시보드를 통해 사람들의 관심도와 반응을 분석하고 이를 바탕으로 최적의 광고 전략을 수립할 수

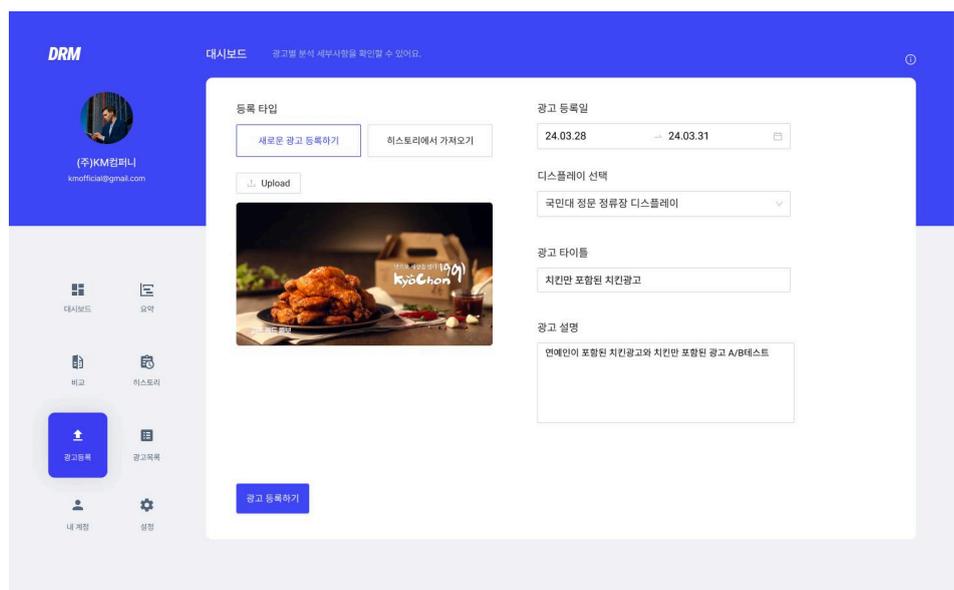
있다. 또한 관심을 가진 대상의 나이 대, 성비 등과 같은 정보를 제공받아, 이를 활용하여 예상 타겟층과 실제 관심을 가진 대상을 비교함으로써 광고의 적절성을 파악할 수 있다. 서비스는 광고주에게 데이터 기반의 솔루션을 제공함으로써 광고의 효과를 극대화하고 광고 캠페인의 ROI를 향상시키는 데 기여한다.

1.2.1. 서비스 주요 기능

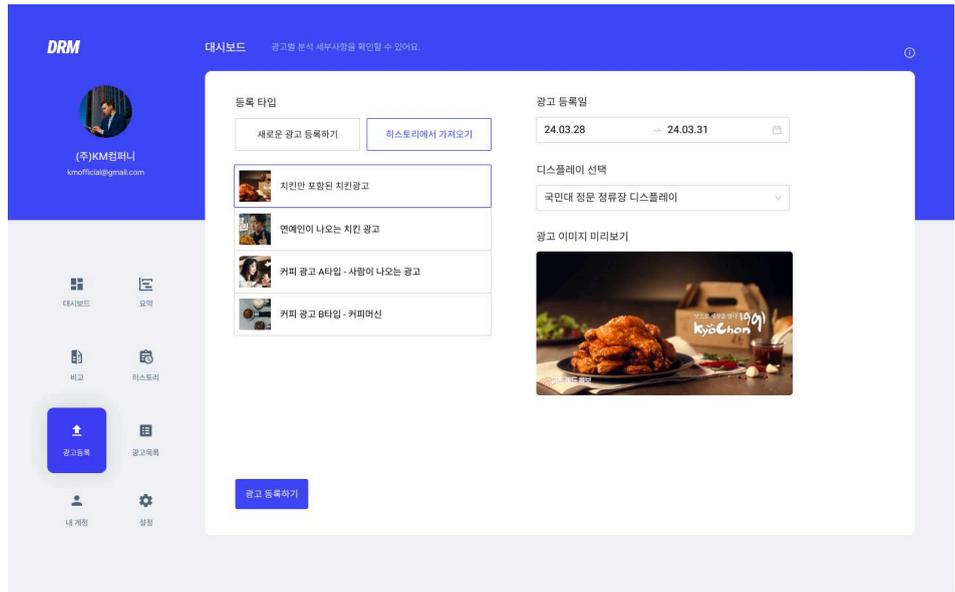
- 고객 유저
- 광고 등록



<광고 등록 default>



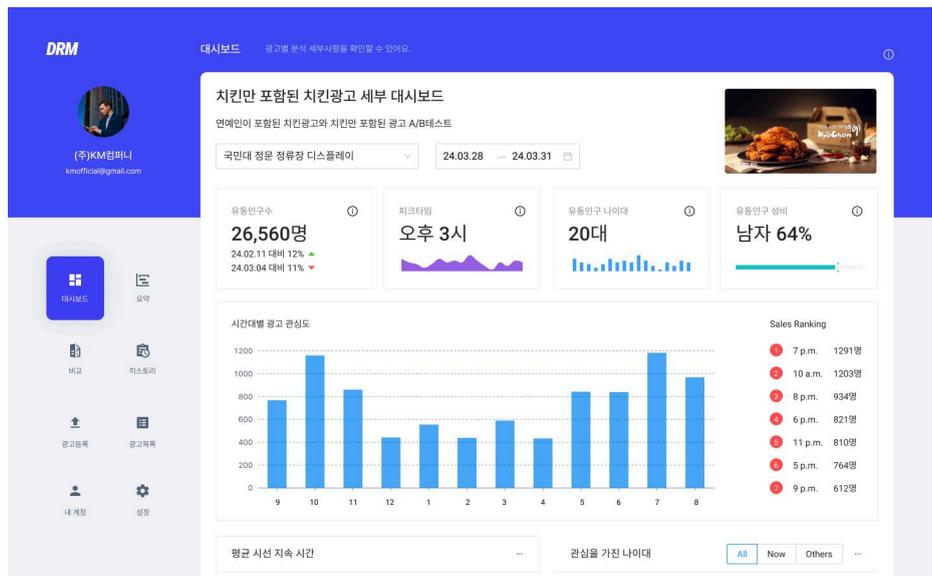
<새로운 광고 등록하기>



〈히스토리에서 가져오기〉

사용자는 '광고 등록' 기능을 통하여 신규 광고 및 이전에 등록했던 광고를 디스플레이에 송출할 수 있다. 사용자는 송출하고자 하는 광고를 업로드 혹은 서비스 히스토리에서 불러온 뒤, 광고 등록일, 송출 디스플레이, 타이틀, 설명 등의 세부 사항도 설정할 수 있다.

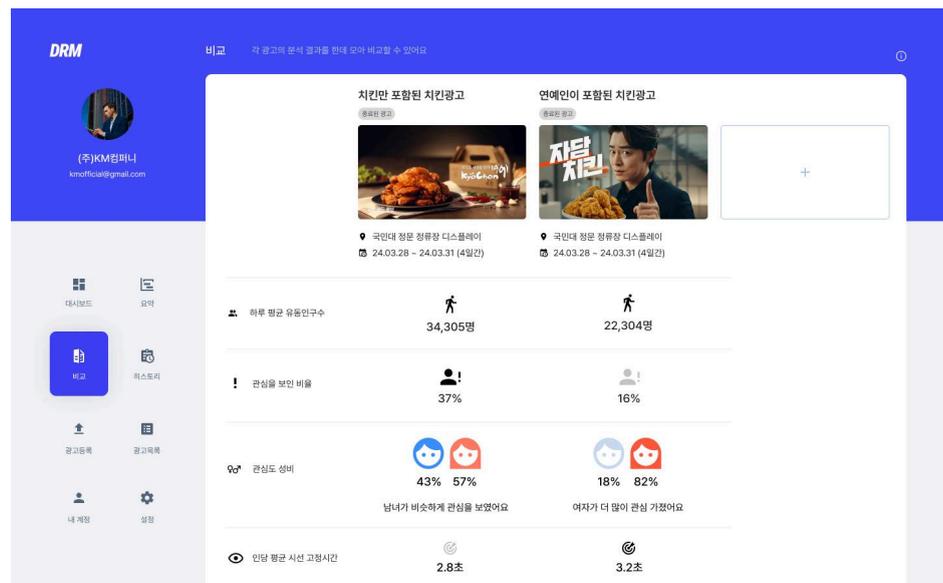
- 광고 분석 결과 대시보드



<광고 분석 결과 대시보드>

사용자가 등록한 광고에 대한 분석 결과를 대시보드 형태로 제공한다. 대시보드에서 제공되는 정보는 유동인구수와 피크 타임(절정기), 광고 소비자의 관심도와 성비, 나이대 등이 있다.

- 광고 분석 결과별 비교 기능



<광고 분석 비교>

사용자가 등록한 2개 이상의 광고를 비교할 수 있는 기능으로 일평균 유동인구수, 광고에 대한 관심도와 성비, 평균 시선 고정 시간 등을 간단하게 비교할 수 있다.

1.3. 서비스 목표

광고대행사 DRM은 디스플레이를 소유하고 있다고 가정하며, 광고 송출과 동시에 광고 분석이 가능한 시스템을 구축한다.

- 객체 분석 기술을 활용하여 광고 디스플레이에 노출된 사람들의 데이터를 정확하게 분석하고 시각화하여 대시보드 형태로 제공합니다.

- 소비자들이 광고를 실제로 관심 있게 보는지에 대한 데이터를 수집하고, 이를 통해 광고의 효과를 평가합니다.
- 이 시스템을 통해 광고주들이 온라인 광고와 유사하게 오프라인 광고 또한 분석할 수 있도록 지원합니다.
- 광고주는 광고 분석 결과를 통해 광고 효과를 이해하고 최적화할 수 있도록 돕습니다.

1.4. 기대 효과 및 활용 방안

- 광고주들은 실제 광고 효과를 정확히 파악하고 최적화할 수 있게 됩니다. 이를 통해 광고 예산을 효율적으로 운용하고 효과적인 광고 전략을 수립할 수 있습니다.
- 지역 소상공인들은 이 서비스를 활용하여 분석된 내용을 반영한 최적화된 광고 전략을 구사할 수 있습니다. 이는 지역 경제 발전에도 도움이 될 것입니다.
- 지역 소상공인들과의 협업을 통해 우리의 기술을 계속 발전시키고 광고 분석 분야의 연구와 발전에 기여할 수 있습니다.

2. 수행 내용 및 중간 결과

2.1. 초기 계획

초기 계획은, 디지털 사이니지(옥외 광고)에 영상형 광고가 송출된다는 것을 가정했다. 디지털 사이니지에서도 영상 광고가 집행되지만 모바일 광고의 특성과 차이가 있다. 모바일에서 집행되는 광고들의 경우, 사용자의 반응, 관심, 이탈 정보를 모두 수집하기에 이 데이터 기반의 광고 집행이 가능하다. 하지만 옥외 디스플레이용 광고의 경우, 유동인구 및 지나가는 사람에 대한 데이터 분석이 없어, 정량적인 광고 집행이 어렵다.

따라서 해당 프로젝트는, 옥외 광고 디스플레이에 카메라가 부착됐을 때를 가정하며, 카메라가 찍어온 유동 인구의 데이터를 분석해서 다음의 가치 창출을 목표로 했다.

- 1) 디스플레이가 위치한 지역의 광고 타겟인 유동 인구의 정보를 수집한다.
- 2) 사람들의 광고에 대한 관심도 정보를 통해 옥외 디스플레이용 광고의 최적화 의사결정용 데이터 제공한다.

2.2. 변경 사항

처음에는 위에서 언급한 “영상” 광고 기반의 데이터 분석 솔루션에 대한 기획을 진행했다. 이 경우에는 카메라를 통해 송신한 사람들의 관심도 정보를 영상형 광고와 동기화 시킬 필요가 있었다. 하지만 이때 적절한 대시보드 디자인 및 데이터 규격을 정의하기 어려웠다. 따라서 송출되는 광고가 이미지라고 가정하고, 카메라에서 송신 받는 사람들의 관심도 데이터 분석 후 대시보드 제공으로 기획을 변경했다.

이처럼, 이미지 광고를 가정하고, 이 광고 디스플레이 인근 카메라에서 수신 받는 사람들의 관심도 데이터 관리 솔루션을 목표하도록 방향을 수정했다.

추후에, 위의 이미지 광고용 데이터 솔루션이 모두 완성되고 난다면, 추가 확장 과제로써 사진이 아닌, 영상형 광고용 데이터 솔루션으로 기획을 확장해서 서비스 제공을 고려하고 있다.

2.3. 수행 내용 및 추진 사항

2.3.1. 프로젝트 전체

2.3.1.1. 팀 소개

팀 DRM은 ‘최적화된 옥외 광고를 위한 광고 효과 분석 솔루션’ 프로젝트를 진행하는 팀으로, 백엔드 개발자 2명, AI 전문가 2명, 프론트엔드 엔지니어 1명, 그리고 기획 및 디자이너 1명으로 구성되어 있다.

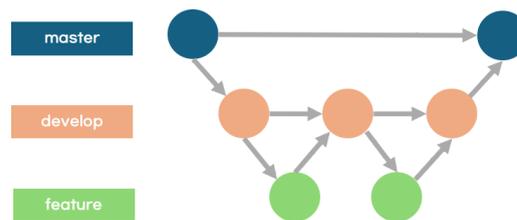
서로 다른 배경과 기술을 가진 팀원들이 협력하여 프로젝트를 이끌어가고 있으며 새로운 기술과 아이디어를 탐구하며 최선의 해결책을 제시하기 위해 노력하고 있다.

팀의 목표는 협업을 통해 최상의 사용자 경험을 가진 서비스를 제공하는 것이다. 각자가 자신의 분야에서 전문가가 되도록 노력하며, 사용자들에게 가치 있는 서비스를 제공하기 위해 노력하고 있다.

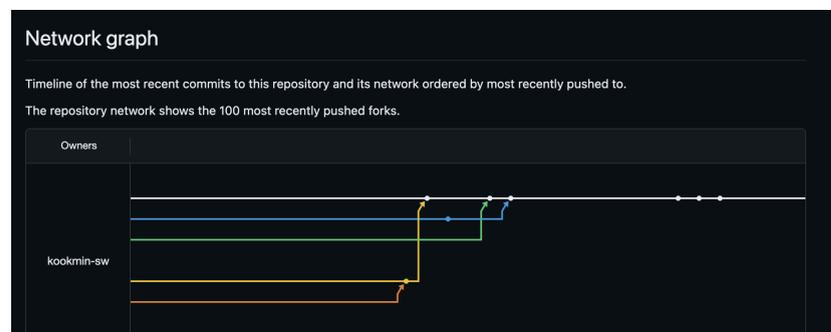
2.3.1.2. 협업 방법

팀원 간 효율적인 협업을 위하여 브랜치 전략으로 GitHub Flow를 개발 방법론으로는 애자일 방법론을 적용하였다. 이러한 접근으로 프로젝트의 유연성을 높이고 빠른 개발 주기를 가지는 것으로 프로젝트의 빠른 완성에 중점을 두었다. 추가로 GitHub Issue Template과 PR Template을 도입하여 작업의 표준화와 효율성을 증진에 노력하였다.

1. GitHub Flow¹



master 브랜치를 기반으로 새로운 기능이나 수정 사항을 위한 브랜치를 생성하고, 작업 완료 후 master 브랜치로 병합하도록 하였다. DRM의 작업 영역을 크게 client, server, model로 구분됨을 고려하여 각각 별도의 develop 브랜치와 feature 브랜치를 생성하도록 하여 프로젝트의 유지 관리를 간소화하였다.

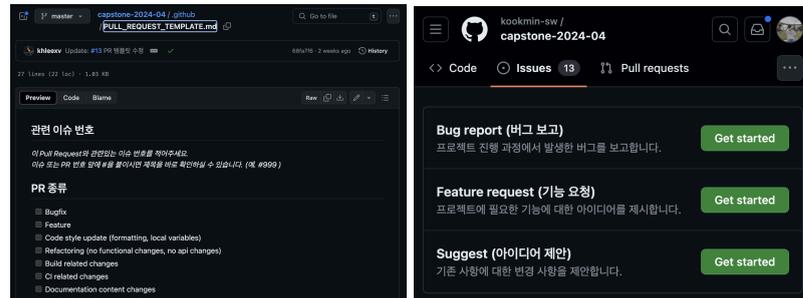


2024.03.30 기준으로 실제 프로젝트에 적용된 모습은 위와 같으며, 저장소 내부 네트워크 그래프를 통하여 확인할 수 있다.

2. 애자일 방법론

세부 기획 및 디자인 등의 변화에 유연하게 대응하고 지속적인 요구사항을 반영할 수 있도록 애자일 방법론을 채택하였다. 프로젝트 내 개발 요구 사항은 하나의 태스크(이슈) 단위로 생성하고 각 팀원에게 할당하여 프로젝트 개발을 진행하였다.

3. Issue template², PR template



프로젝트 참여 팀원들에게 일관된 형식의 Issue와 Pull request 제공 목적으로 도입하였다.

2.3.2. 클라이언트

서비스의 사용자(광고주)가 주로 관심을 가질 법한 데이터를 보다 쉽게 접근하고 이용할 수 있는 사용자 인터랙션들을 추가하고 광고 개별 분석, 광고 비교 분석 등 광고주가 ROI를 한눈에 알아볼 수 있는 시각적인 대시보드를 제공하는 것을 목적으로 진행 중이다.

특히 사용자 경험을 좌우하는 속도, 사용성, 반응성 향상과 빠르고 효율적인 개발에 초점을 맞추어 사용 기술 스택 및 라이브러리를 결정하였고 주요 기술 스택 및 라이브러리는 아래와 같다.

1. Vite(React)

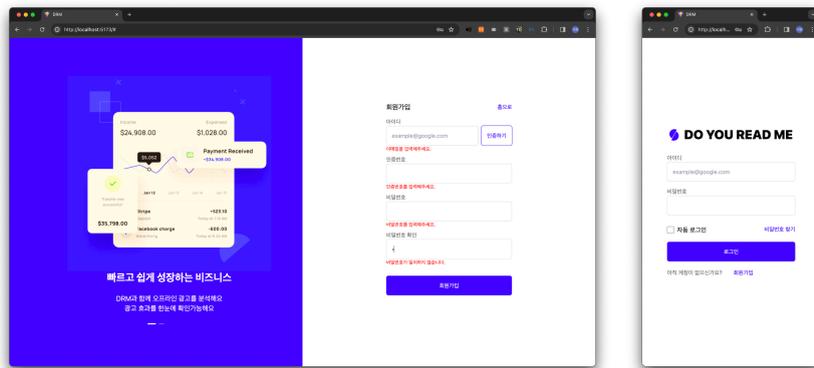
SPA(Single Page Application)을 사용하는 것으로 속도 사용성, 반응성 향상 등의 이점을 가져가는 것으로 좋은 사용자 경험을 제공하기 위하여 채택하였다.

2. TypeScript

타입 체킹을 도입시키는 것으로 컴파일 단계에서의 오류 방지 및 실행 속도 개선을 목적으로 채택하였다.

3. Ant Design

대시보드 UI component 제공 용도이며 세부 디자인 결정되기 이전에 간단히 사용하기 위하여 사용하고 있다.



현재 클라이언트는 초기 화면(로그인, 회원가입 화면) UI와 입력 유효성 검사, 그에 대한 에러 텍스트 노출 처리가 완료된 상황이다. 추가로 다양한 디바이스에서 서비스를 이용할 것을 고려하여 tailwind css를 사용하여 반응형 디자인을 적용하였다.

2.3.3. 서버

1. 목표

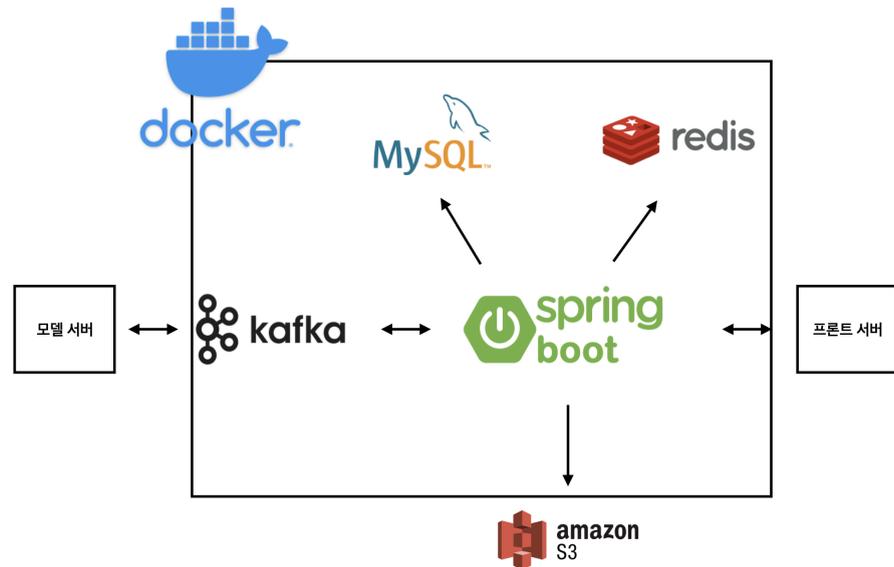
모델 서버와 대시보드 화면을 연결하는 역할로 양쪽의 요청을 처리하며 데이터를 저장하는 역할을 한다. 유저 타입이 관리자와 광고주 두 타입이며 두 타입별 서비스 로직 구현을 목표로 하고 있다. 광고주의 경우 자신의 광고를 등록하고 관리하며 모델에서 들어온 분석 데이터를 대시보드로 볼 수 있는 서비스를 제공한다. 이때 대시보드에 보일 데이터를 시간별 통계, 연령대, 성별 등 다양한 형태의 데이터를 제공하기 위한 로직을 작성한다. 관리자는 광고주가

요청한 광고들을 승인 및 관리하고 이를 분석할 수 있게 광고를 화면에 띄우는 서비스를 제공한다.

프론트 대시보드와 서버의 연결은 HTTP 프로토콜 이용하여 통신하며 모델 서버에서 들어오는 객체 데이터의 처리는 메시지 큐를 이용한다. 여러 모델 서버에서 한 번에 대용량 데이터가 들어오는 경우를 생각해서 카프카를 이용해 비동기 처리를 하여 서버가 원하는 때 데이터를 받아 부하가 되지 않게 하며 대용량의 데이터를 조회 할 때의 최적의 쿼리를 제공한다.

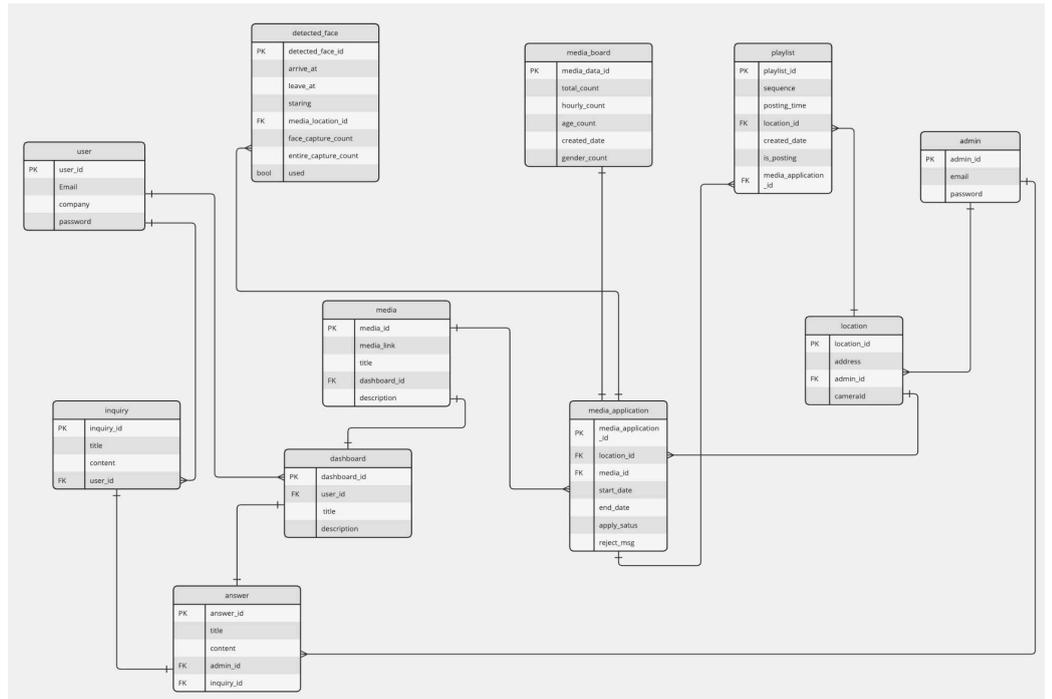
모델 서버와 프론트 서버에서 api를 사용하기 위해 도커 컨테이너로 배포를 진행하며 카프카, 스프링, mysql, redis 등 다양한 기술 스택에 대한 버전을 유지하고 테스트하기 쉬운 형태를 제공한다.

2. 아키텍처



- kafka: 모델 서버에서 보내는 대량의 데이터 저장
- springboot : 모델 서버에서 보내는 데이터와 프론트 서버에서 보내는 데이터에 대한 로직 처리 및 대시보드에 보여질 데이터 로직 제공
- mysql : 모델에서 넘어오는 데이터 저장, 대시보드 관련 데이터 저장
- redis : 로그인 시 토큰 저장, 이메일 인증 번호 저장을 위한 데이터베이스
- s3: 광고 콘텐츠를 저장하는 스토리지
- docker : 카프카, 스프링 서버, mysql, redis 를 컨테이너화하여 배포

3. ERD



4. 기술 설명

1) kafka 사용

1초에 여러명의 사람이 포착되고 이러한 카메라가 여러대라고 가정했을때 이를 매번 서버에 요청을 보내면 서버에 부하가 생길거라 예상되어 비동기적으로 처리하기 위해 메시지 큐 방법을 채택했다. 카프카에서 데이터를 가져올 때의 방법을 아래의 세가지 방법을 고민하고 있으며 각 방법의 장단점을 나눠 선택할 예정이다.

주기별 저장 :

- a) 리스너를 통해 토픽안에 데이터가 변경될 때 특정 주기동안 리스트에 저장후 한번에 DB에 저장한다.
- b) 특정 주기마다 카프카에서 전에 읽었던 데이터 이후부터 pull을 해와서 한번에 DB에 저장한다.

용량 별 저장:

리스너를 통해 변경될 때 리스트에 담아놓고 일정 버퍼가 차면 한번에 DB에 저장한다.

2) redis를 이용한 토큰 및 이메일 인증번호 관리

redis는 nosql db로 "유저 id값 : 리프레시 토큰" 형태로 리프레시 토큰을 관리하는 용도로 사용한다. redis는 In-memory 방법을 사용하기 때문에 RDB 비해 성능이

빠른편이다. 리프레시 토큰의 경우 액세스 토큰을 재발급하기 위해 자주 호출 되기때문에 성능이 빠른 redis를 선정하여 관리한다. 또한 redis는 유효기간을 지정할 수 있기 때문에 토큰 만료일로 유효기간을 설정할 시 토큰이 자동으로 삭제 되어 토큰 관리에 용이하다. 이메일 인증 번호의 경우 인증이 끝나면 더이상 필요없는 일회성 데이터이기 때문에 redis를 이용하여 관리한다.

3) 이메일 인증

초기 회원가입 시 이메일 검증을 위해 SMTP를 이용하여 인증 번호를 보낸다. 개발 비용을 줄이고 보안을 위해 google에서 제공하는 Gmail SMTP 사용한다. 사용자가 인증번호 요청 시 우리 서버에서 인증 코드 생성하여 고객 이메일에 보내는 동시에 redis에 저장하고 이를 통해 사용자가 요청한 인증번호 대조하는 서비스를 제공한다.

4) 도커 컨테이너를 이용한 배포

kafka, spring, redis, mysql 등 여러 기술 스택을 docker-compose 파일에 한번에 관리하고 각 서비스의 의존성을 간단하게 정의할 수 있어 배포에 용이하며 일관된 배포가 가능하다.

5) swagger를 이용한 API 명세서 관리

API 엔드 포인트, 매개변수, 요청 및 응답 형식, 인증 및 권한 등 세부 정보를 작성한 코드대로 자동화시켜 생성한다. 이는 프론트 개발자와 불필요한 소통을 줄이고 API를 테스트하는데 도움을 준다.

5. API

Apply

- DELETE** /api/v1/dashboard/media/{mediaId}/apply/{applyId} 신청 취소
- GET** /api/v1/dashboard/media/{mediaId}/apply 신청 리스트 조회
- PATCH** /api/v1/dashboard/media/{mediaId}/apply/{applyId} 신청 수정
- POST** /api/v1/dashboard/media/{mediaId}/apply 광고 신청

auth 인증이 필요한 api

- POST** /api/v1/auth/signup 회원가입
- POST** /api/v1/auth/signin 로그인
- POST** /api/v1/auth/email/verification 인증번호 확인
- POST** /api/v1/auth/email/verification-request 인증번호 전송

Dashboard 대시보드 관련 API

- GET** /api/v1/dashboard 전체 대시보드 리스트 조회
- GET** /api/v1/dashboard/{dashboardId} id별 대시보드
- GET** /api/v1/dashboard/media/{mediaId}/board 특정 광고에 대한 대시보드 리스트
- GET** /api/v1/dashboard/media/{mediaId}/board/{boardId} 특정 광고 신청 단위별 대시보드

detected-data-controller

- POST** /api/model

Location

- GET** /api/v1/location 디스플레이 장소 조회

Media 광고 등록 관련 api

- DELETE** /api/v1/media/{mediaId} 광고 삭제
- GET** /api/v1/media 등록된 광고 리스트
- GET** /api/v1/media/{mediaId} id 별 광고 조회
- PATCH** /api/v1/media/{mediaId} 광고 수정
- POST** /api/v1/media 광고 등록

swagger link : <http://175.45.203.201:8080/swagger-ui/index.html#/>

2.3.4. 모델

1. AI 개발에서의 목표

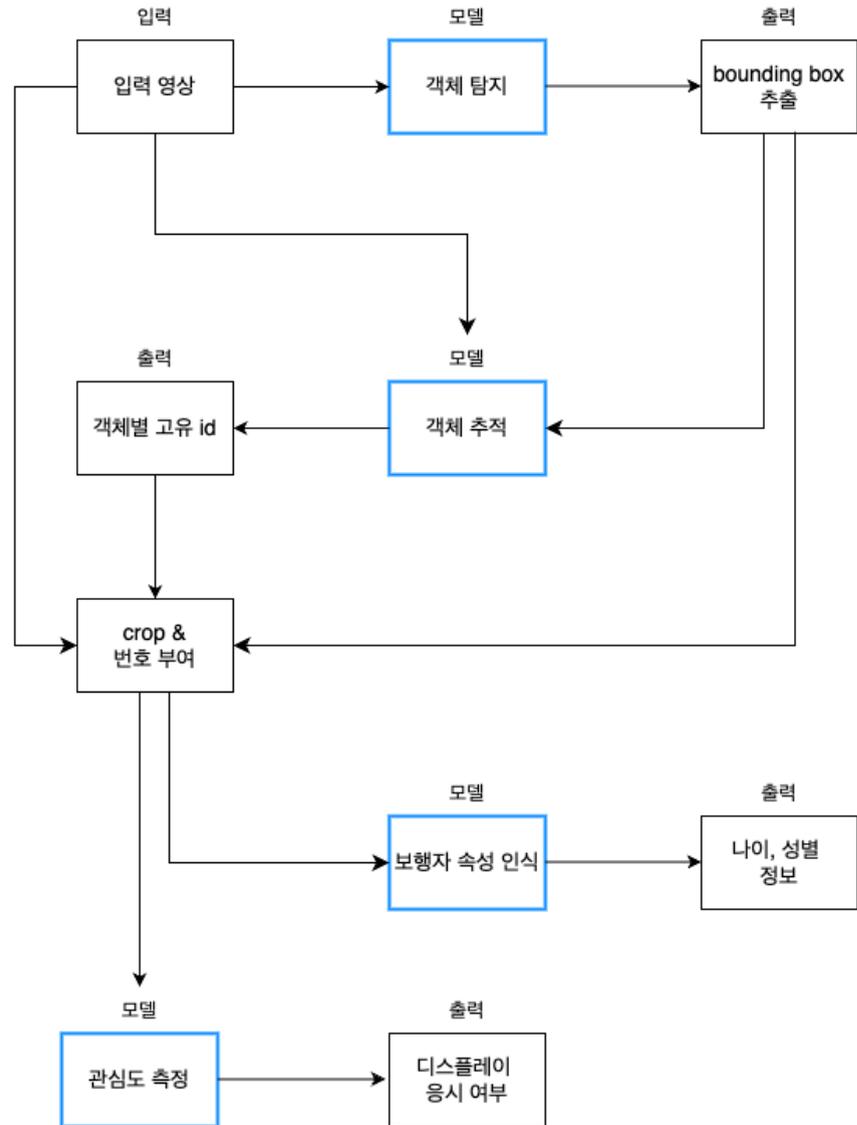
디스플레이에서 송출되는 광고에 대한 사람들의 관심도를 파악하고, 동시에 유동인구의 나이대나 성별 등의 정보를 수집하고자 한다. 이를 위해서 디스플레이 상단의 카메라로 유동인구의 영상을 수집하여 모델의 입력으로 사용하고, 유동

인구의 관심도와 속성 파악을 위해 모델 파이프라인을 구성한다. 우선 디스플레이에 송출되는 광고에 대한 사람들의 관심도를 파악하기 위한 관심도 지표를 제시한다. 디스플레이를 바라보고 있는지를 판단하기 위해서 헤드 포즈 모델, 시선 추적 모델 또는 아이 컨택 모델을 사용할 예정이며, 실험 결과에 따라 광고에 관심을 가졌다고 판단할 정확한 수치나 상황을 정하고자 한다. 또한, 광고에 관심을 가지는 사람들에 대한 나이 대나 성별 정보를 함께 제공하고자 한다.

광고에 대한 사람들의 관심도를 측정하고, 이들의 속성을 파악하기 위해서 크게 세 가지 모델을 사용한다. 첫번째는 영상에 등장하여 움직이는 사람들을 추적할 수 있는 객체 추적 기술이다. 두번째는 영상 프레임에 등장하는 소비자들의 연령대, 성별 정보를 획득하기 위한 보행자 속성 인식 기술이다. 마지막으로 소비자들이 실제 디스플레이를 바라보고 있는지를 판단하기 위해서 헤드 포즈, 시선 추적 또는 아이 컨택 모델을 사용하고자 한다.

각 모델을 거치며 수집된 데이터는 사람 단위로 가공되며 최종적으로는 시각적으로 쉽게 이해할 수 있는 대시보드가 광고주에게 제공된다. 이를 통해 광고주들은 디스플레이 근처의 유동 인구에 대한 정보와 이들의 관심도를 수치적으로 파악하여 광고 전략을 수립함으로써 옥외 광고의 효과를 극대화할 수 있다.

2. 모델 단에서의 전체적인 워크플로우



입력 영상이 들어오면, 먼저 사람 객체를 인식하여 bounding box를 추출한다. 영상에서 각각의 사람들을 시간에 따라 추적하고, 각 객체에 고유한 ID를 할당하기 위하여 객체 추적 모델을 사용한다. 이 단계에서 인식된 객체의 영역을 자르고 이후 모델들의 입력으로 사용된다. 먼저 객체의 속성을 인식하고 분류하기 위해서 보행자 속성 인식 모델을 사용한다. 이 단계에서는 사람들의 나이 대, 성별 정보를 얻을 수 있다. 다음으로는 객체의 영역에서 얼굴을 인식하는 모델을 사용하여 얼굴에 해당하는 bounding box를 추출하여 이를 헤드 포즈 또는 시선 추적 모델의 입력으로 사용한다. 해당 모델에서는 객체의 헤드 포즈 또는 시선 방향을 분석하고, 출력값을 이용하여 사람이 디스플레이에서 송출되는 광고에 얼마나 관심을 가지고 있는지를 판단한다.

3. 모델 상세 설명

1) 객체 추적 모델

- 목적

공공장소에 설치된 디스플레이를 지나치는 사람들을 추적하여, 각 사람들이 광고판 앞에서 보내는 시간을 측정하고 움직임 분석을 위해서 사용된다. 이 정보는 디스플레이에서 송출되는 광고가 얼마나 효과적으로 유동인구의 관심을 끄는지를 평가하는 데 핵심적인 기능이 된다. 해당 모델에서의 입력은 디스플레이 상단의 카메라로 촬영한 유동인구의 영상이며, 출력은 탐지한 사람을 박스 형태로 잘라낸 결과와 각 객체에 할당된 고유 ID이다.

- 세부 설명

- 사용 모델 : 객체 탐지 기반의 추적 모델 BoT-SORT³ 사용
- 동영상에서 객체 추적에는 다양한 접근 방법이 있으나, 객체 탐지 결과를 기반으로 객체를 추적하는 SORT(Simple Online and Realtime Tracking) 기반의 모델 BoT-SORT 를 사용할 예정이다. 이 모델은 최근 다중 객체 추적 벤치마크 데이터셋에서 가장 높은 성능을 달성했으며, 객체 검출, 객체 추적 모델이 분리되어 있는 구조이기 때문에 추후 각 모델을 다른 모델로 변경하기에도 용이하다는 장점이 있다.
- 데이터 처리 : 입력된 영상에서 객체를 감지하고, 각 객체에 고유한 ID를 할당한 후, 이들의 움직임을 추적한다. 이때, 다음 프레임에서 같은 사람이 등장하는 경우 동일한 아이디를 부여하는 재식별 기능을 사용하여 각 사람에 대한 정확한 정보를 수집하는 동시에, 객체 탐지의 정확도를 높이고자 한다.
- 객체 추적 모델에서 검출된 사람들 중 일정 시간 이상 체류한 사람들을 선별하여 해당 사람들을 박스 형태로 잘라낸다. 이 정보는 보행자 속성 및 관심도 측정 모델과 결합하여 관심도를 측정하는 데 사용된다.

2) 보행자 속성 인식 모델

- 목적

각 사람들의 나이 대, 성별 등의 속성을 식별하여 광고주가 타겟 층에게 더욱 적합한 광고를 제공할 수 있게 돕는다. 이 정보는 광고주가 광고를 송출할 위치와 시간에 대한 전략을 세우고 더 나아가 광고의 효과를 분석할 수 있게 한다. 해당

모델에서의 입력은 객체 추적 모델에서 잘라낸 박스 형태의 이미지와 고유 id 값이며, 출력은 사람의 나이 대와 성별 등의 보행자 속성 정보이다.

- 세부 설명

- 보행자 속성 인식 데이터셋 Market-1501에서 사전 학습된 ResNet50 기반의 모델⁴을 사용하고 있다. 해당 모델은 보행자의 외형 이미지를 기반으로 성별, 머리 길이, 복장 정보, 연령대 등 27가지 정보를 추론할 수 있다. 이 중 필요한 정보를 수집 및 가공하여 제공할 예정이다.

3) 관심도 측정 모델

- 목적

광고 대상의 실제 관심도를 측정하기 위해 헤드 포즈, 시선 또는 카메라를 바라보는지를 추적한다. 이 모델은 보행자가 디스플레이를 바라보는 각도와 시간을 분석하여, 관심도를 수치화하여 제공한다. 해당 모델에서의 입력은 잘라낸 사람의 박스 형태의 이미지이며, 관심도를 수치화한 값을 출력하고자 한다.

- 세부 설명

- 관심도 측정을 위해서 논의된 방법은 크게 3가지이며, 각 방법을 설명한다.
- 헤드 포즈 모델은 가장 단순한 방식으로, 사람의 머리가 어떤 3D 자세를 갖고 있는지 측정하여, 머리의 3D 좌표와 회전 정보를 바탕으로 해당 사람이 광고 디스플레이에 관심을 갖고 있는지 측정할 수 있다. 단순한 만큼 측정 값이 정확하지 않을 것으로 예상된다.
- 시선 추적 모델은 헤드 포즈가 아닌, 실제 사람의 시선이 어디를 향하는지를 추론한다. 좀 더 정확한 관심도 측정이 가능할 것으로 예상되며, 더 나아가 단순히 광고에 대한 관심도 뿐만 아니라 디스플레이의 응시 지점에 대한 정보까지 획득할 수 있을 것으로 예상된다. 하지만, 기존 시선 추적 연구들을 살펴보면 한 사람만 가까운 거리에서 등장하는 등 제한된 상황에서 이뤄진 것으로 확인된다. 움직이는 여러 사람에 대한 연구는 최근에 조금씩 발표되고 있는 것으로 보인다. 디스플레이 응시 지점까지 정확히 추론하기에는 한계가 있을 것으로 예상된다.
- 아이 컨택 모델은 사람이 카메라를 향해 관심을 갖고 있는지 여부를 측정한다. 관심도 측정이라는 목적과 부합하는 모델이며, 출력 값 또한 0,

1로 이진 분류기에 활용하기에도 가장 쉬울 것으로 예상된다. 하지만 많은 연구가 이뤄지지 않은 분야이기 때문에 구현 시 어려움이 있을 것으로 예상된다.

3. 향후 추진 계획

3.1. 프로젝트 전체

- 이미지 광고를 대상으로 1차 프로토타입을 완성한다. 이후 여건이 된다면 동영상 광고라고 가정하고 추가적인 서버, 대시보드 로직을 추가한다.

3.2. 클라이언트

- 디바운스(Debounce) 및 스로틀(Throttle)을 사용하여 사용자의 악의적인 동작을 막고 서버로의 요청을 최소화한다.
- Storybook 라이브러리를 도입하여 재사용 가능한 UI 요소들을 기반으로 하여 UI component를 구축하고 효율적으로 관리한다.
- Figma에서 정의된 디자인 시스템을 프로젝트 내부 theme으로 작성하여 변화에 유연하게 대응한다.
- netlify를 활용하여 임시배포를 진행한 후, AWS를 통해 본격적인 배포를 진행한다.
- GitHub CI/CD를 추가하여 웹 배포를 자동화한다.
- 더욱 다양한 디바이스에 대응할 수 있는 Responsive UI를 적용한다.

3.3. 서버

- 모델 서버에서 들어오는 데이터를 대시보드에서 보여주기 위한 저장 타입을 확장성 있게 고려한다.
- 모델 서버에서 동시 요청되는 대용량 데이터들을 처리하기 위한 부하 테스트를 진행하고 이를 감당할 수 있도록 kafka 구조를 재정의한다.
- 서버의 데이터베이스 접근 및 조회 속도를 개선하기 위하여 쿼리를 최적화한다.

- CI/CD를 이용해 서버 배포를 자동화한다.
- 현재 사진으로 가정하고 개발 중인 광고가 영상으로 확장하여 기능을 제공할 때의 촬영 중인 영상과의 싱크 맞추는 로직 개발한다.

3.4. 모델

- 관심도를 측정하기 위해서 어떤 지표를 사용할지 결정한다. 헤드 포즈 정보, 시선 정보, 카메라를 바라보는지 등 관심도에 대한 지표가 될 수 있는 정보를 수치화하기 위해서 모델들을 실험하고 조사한 뒤, 최종적으로 사용할 관심도 지표를 정한다.
- 각 모델의 성능 개선을 위한 실험을 진행한다. 실험과 더불어 환경 설정 방법에 대해 조사하고 논의한다.
- 모델의 성능을 평가하기 위한 데이터셋을 수집하고, 모델 성능 평가 방안을 조사한다.
- Kafka를 사용하여 백엔드와의 통신 기능을 구현한다.

4. 참고 자료

[1] *GitHub flow*. 24.3. GitHub Docs.

<https://docs.github.com/en/get-started/using-github/github-flow>

[2] *Configuring issue templates for your repository*. 24.3. GitHub

Docs.<https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/configuring-issue-templates-for-your-repository>

[3] Aharon, Nir, Roy Orfaig, and Ben-Zion Bobrovsky. "BoT-SORT: Robust associations multi-pedestrian tracking." arXiv preprint arXiv:2206.14651 (2022).

[4] <https://github.com/Yukun-Huang/Person-Attribute-Recognition-MarketDuke>