

사랑회요

6조

20160700 안세혁

20170902 이주언

20190541 방준식

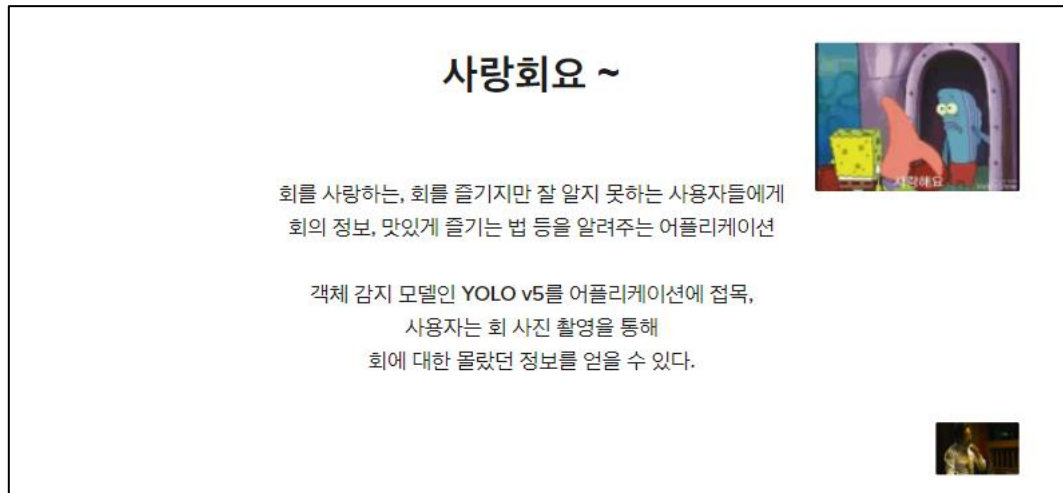
20191250 현소담

목차

1. 프로젝트 개요.....	3
1.1 개발 동기.....	
1.2 개발 목표.....	
2. 요구사항 분석.....	4
2.1 유스케이스.....	
2.2 아키텍처 다이어그램.....	
3. 설계내용	5
3.1 모듈 설계.....	
3.2 주요 알고리즘 설계	
3.3 DB 설계.....	
3.4 UI 구체화.....	
4. 개발 과정 및 구현 방법.....	8
4.1 개발 환경.....	
4.2 사용 라이브러리 및 오픈소스	
4.3 개발 일지.....	
5. 구현 내용 및 최종 결과물	13
5.1 최종 결과물	
5.2 시연.....	
5.3 소스 소개.....	
5.4 테스트 과정 및 결과.....	
5.5 사용 매뉴얼	
5.6 목표 달성 여부	
6. 기타	24
6.1 한계 및 고찰.....	
6.2 발전 방향 및 향후 계획.....	
6.3 참고문헌	

1. 프로젝트 개요

1.1 개발 동기



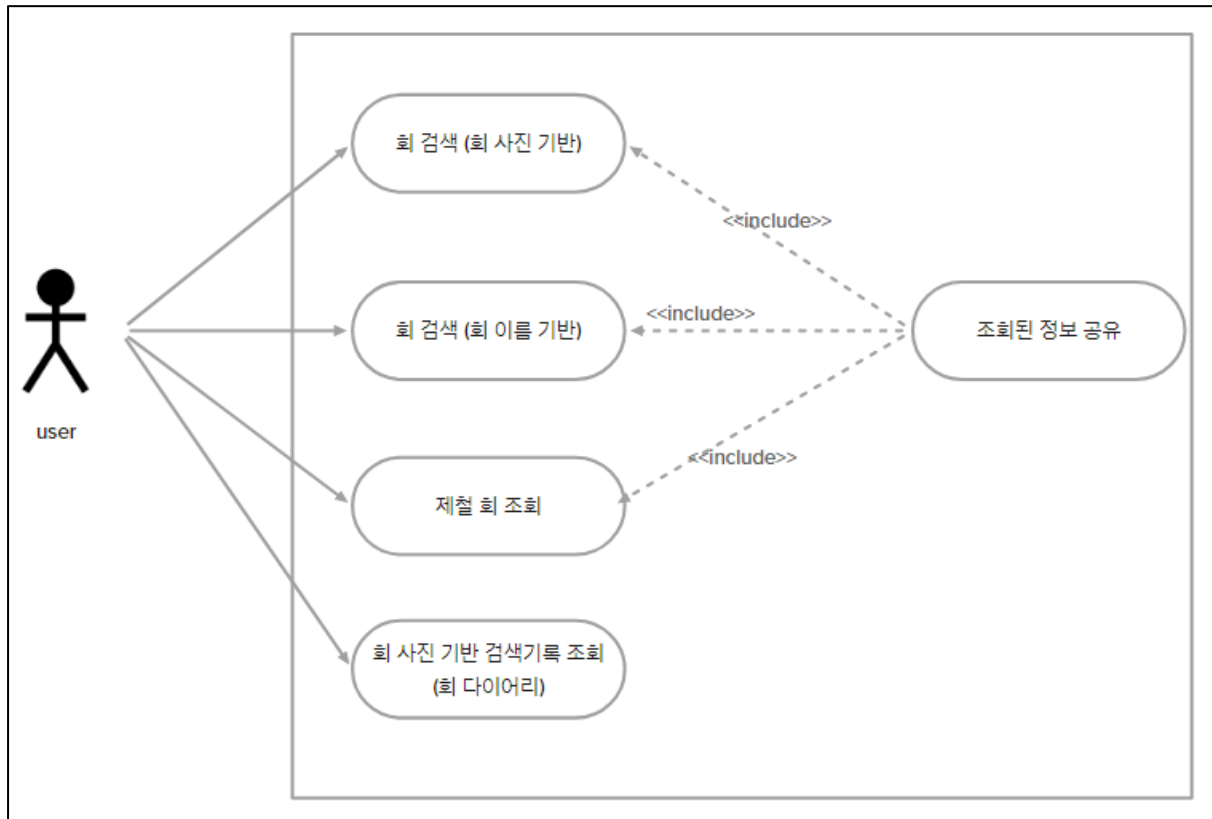
- 평소 회를 좋아하나 모듬회를 시키고 자신이 어떤 회를 먹고 있는 지도 모르는 채 회를 먹는 사람들이 있다. 이러한 사람들에게 회 이름과 더불어 제철 정보, 맛있게 먹는 방법 등을 함께 제공하여 단지 회를 좋아하기만 하는 사람이 아닌, 회잘알(회를 잘 아는 사람)에 가까워지도록 돕고자 하였다.

1.2 개발 목표

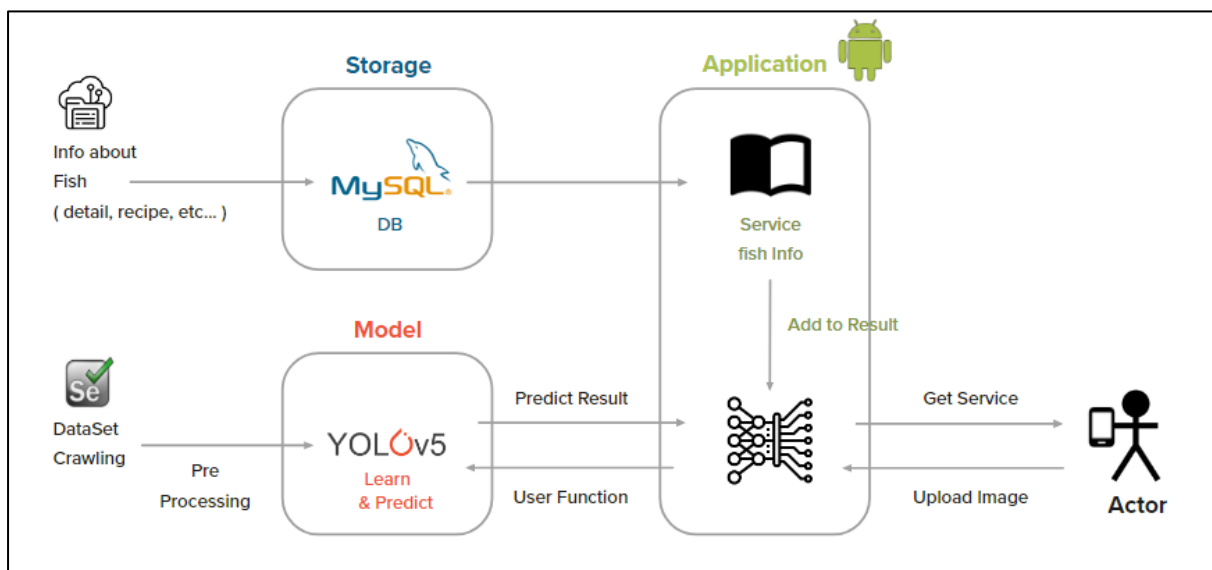
1. 학습 데이터 수집 - 학습을 위한 회 이미지 데이터 수집	2. 모델 학습 - 회를 인식 및 분류할 모델 학습
3. db 구축 - 회 종류별 사용자에게 제공할 데이터 구축	4. 어플 제작 - 일반인들이 사용하기 쉽도록 해당 서비스를 어플로 제작

2. 요구사항 분석

2.1 유스케이스



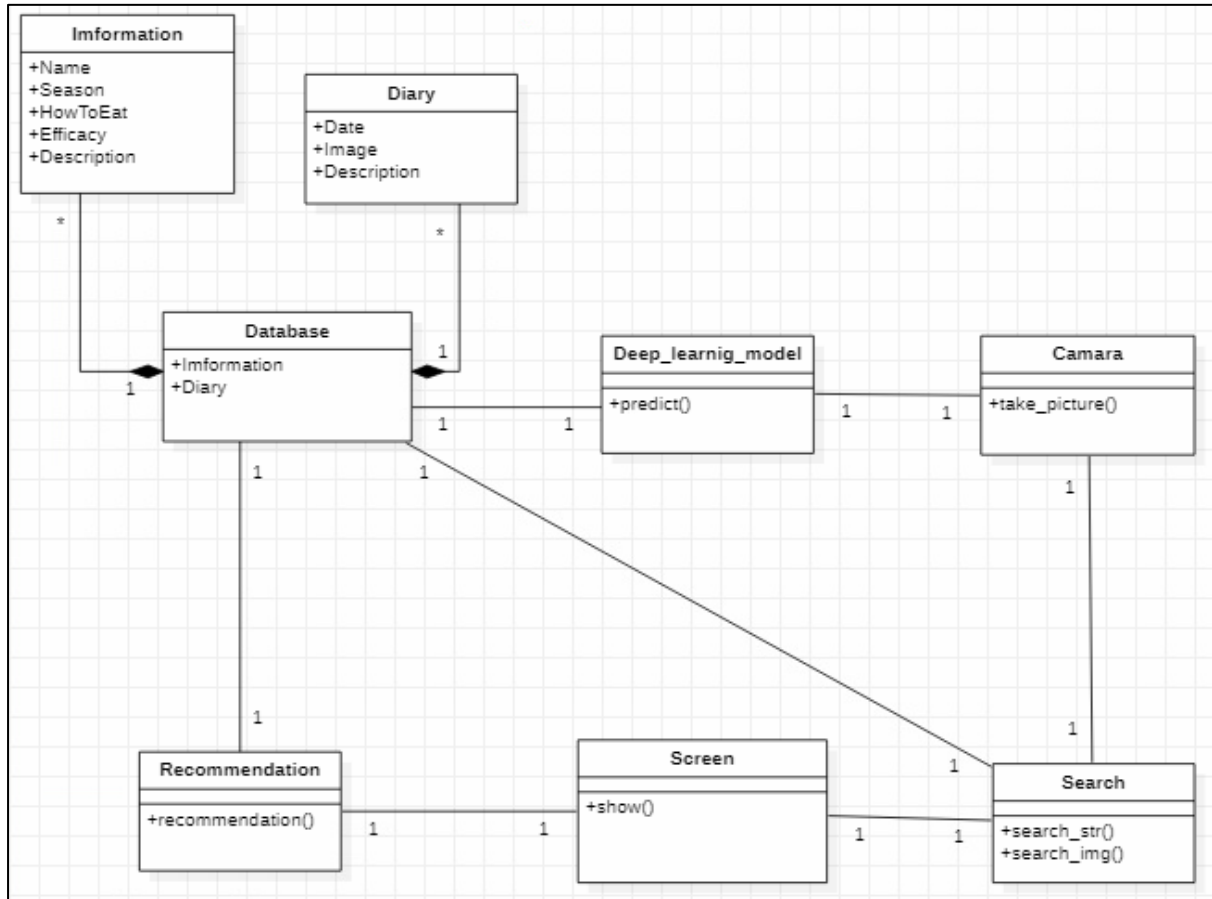
2.2 아키텍처 다이어그램



- DB는 개발 과정에서 Firebase로 변경되었음

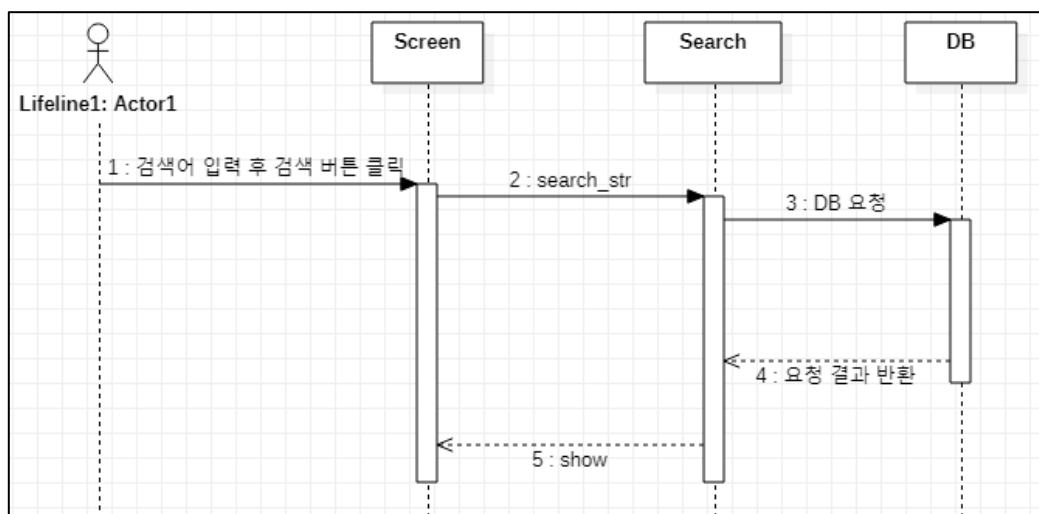
3. 설계 내용

3.1 모듈 설계 (클래스 다이어그램)

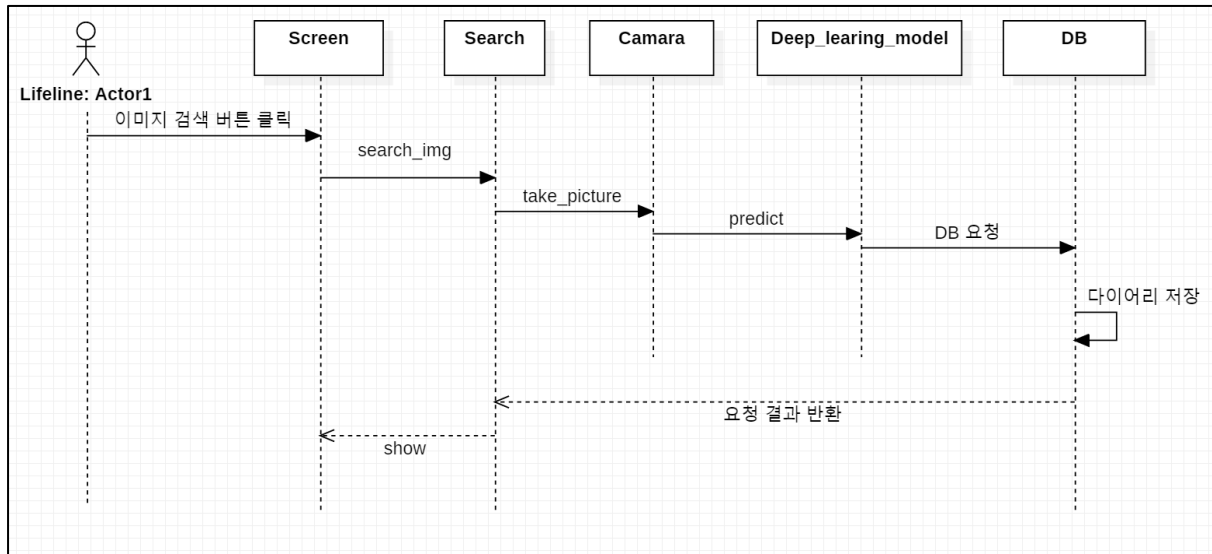


3.2 주요 알고리즘 설계 (시퀀스 다이어그램)

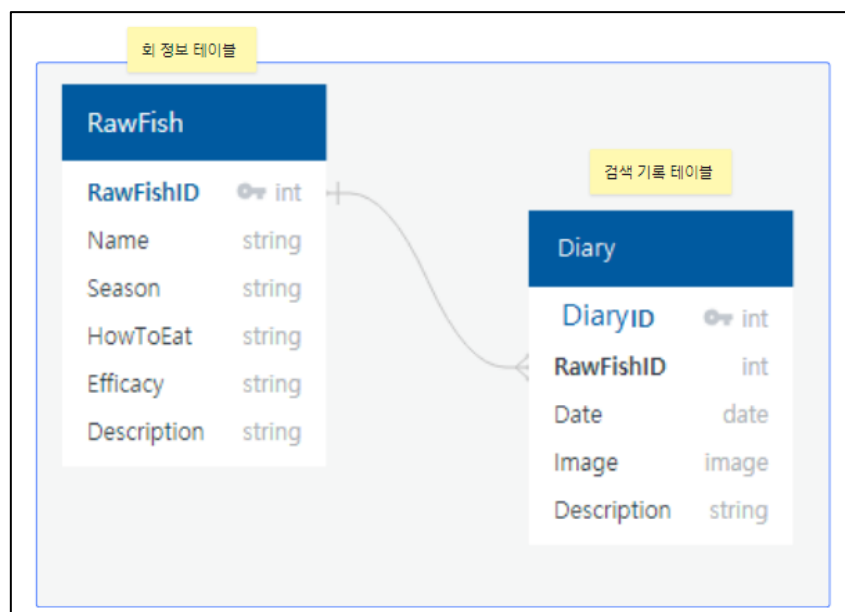
1) 회 이름 기반 검색



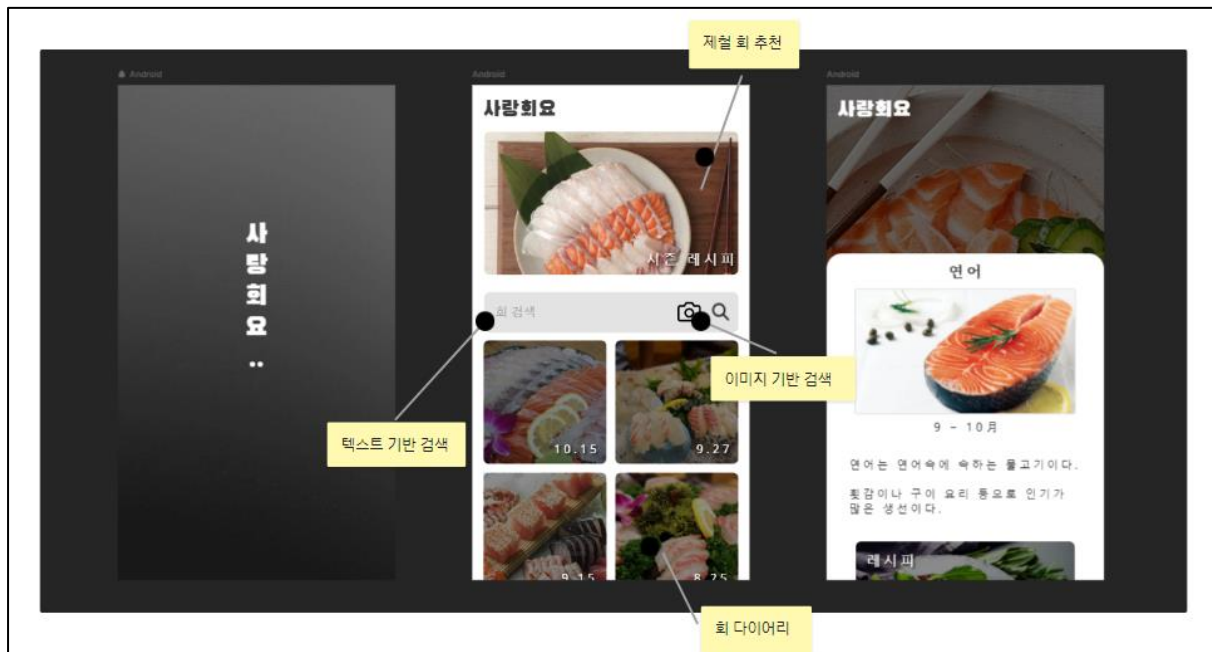
2) 회 이미지 기반 검색



3.3 DB 설계



3.4 UI 구체화



4. 개발 과정 및 구현 방법

4.1 개발 환경

- 어플 제작: Android Studio 사용
- 딥러닝 모델 서버 환경: Ubuntu 18.04.6 LTS, Jupyter Notebook, 1080

4.2 사용 라이브러리 및 오픈소스

1. 학습 데이터 수집 - 셀레니움 (파이썬 크롤링 라이브러리)	2. 모델 학습 - Yolo v5 (객체 탐지 딥러닝 모델)
3. db 구축 - Firebase	4. 어플 제작 - 안드로이드 스튜디오 사용




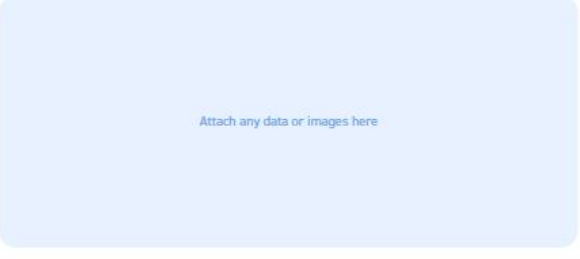

4.3 개발 일지

1) ~11/18



딤러닝 모델 구축 (이주연)	DB 구축 (현소담)
	
<p>Date ~11/18</p>	<p>Date ~11/18</p>
<p>Summary</p> <ul style="list-style-type: none"> • 크롤링 코드 완성 • 광어, 우럭, 말치, 참치, 연어 이미지 수집 • 필요 없는 이미지 삭제 • 딤러닝 구동 환경 설정 • 참고: https://github.com/dami138/loveSashimi.git 	<p>Summary</p> <ul style="list-style-type: none"> • 깃헙 프로젝트 생성 https://github.com/dami138/loveSashimi.git • 사용자에게 제공할 DB 구축 완료 • mysql → Firebase firestore로 변경 • Android Studio와 연동 테스트 완료

Application Design(안세혁)	앱 UI 1차 (방준식)
	
<p>Date ~11/18</p>	<p>Date ~11/18</p>
<p>Summary</p> <ul style="list-style-type: none"> • App 사용자 편의성에 대한 디자인 • App 디자인 IDE 실사용 • 예제 디자인 생성 및 응용 <p>Framer Design Example : https://framer.com/share/k5VYlm8f2HtETvLsQsaX/AQ4YU5iB4</p>	<p>Summary</p> <ul style="list-style-type: none"> • SplashActivity와 MainActivity로 구성 • 앱 로딩 화면 • '시즌 레시피' 추천 버튼 • 텍스트 기반 검색 창 • (아직 기능은 미구현)

2) ~11/25

딥러닝 모델 학습(이주언)	Yolo v5 → tflite convert 및 android 테스트 (현소담)
 	
<p>Date ~11/25</p>	<p>Date ~ 11/25</p>
<p>Summary</p> <ul style="list-style-type: none"> 모든 이미지에 바운딩 박스 입력 딥러닝 모델 학습 <p>데이터 확인: https://github.com/damit38/loveSashimi/tree/main/data 학습 결과 확인: https://github.com/damit38/loveSashimi/tree/main/yolov5</p> <p>생강을 참치로 인식 → 배경 데이터를 모아 정확도 향상 필요</p>	<p>Summary</p> <ul style="list-style-type: none"> Yolo v5에서 자체적으로 제공하는 tflite converter를 통해 파일 형식을 변환하였으나 android app 테스트를 위해 해당 모델을 예제에 적용하던 중 오류 발생, 원인 파악 중 https://github.com/damit38/loveSashimi/issues/2 정 안되면 서버를 통해 결과를 얻어오는 형식을 취해야 할 것 같음 또는 tflite로 학습을 시켜보는 방안도 있는데, 이 방식을 택할 시 정확도가 잘 나올 지 모르겠음
데이터 전처리(안세혁)	앱 UI 2차 (방준식)
	
<p>Date ~</p>	<p>Date ~11/25</p>
<p>Summary</p> <ul style="list-style-type: none"> . 	<p>Summary</p> <ul style="list-style-type: none"> 이미지 기반 검색 버튼(카메라 버튼) 카메라 연동 후 실행 확인 회 다이얼리(이미지와 날짜 출력) 검색 결과 페이지 구현(MainActivity2) 회 이름, 이미지, 제철, 회 정보 등 출력(회 정보 TextView 스크롤 가능)

3) ~12/02

딤러닝 모델 학습(이주언)	어플에 DB 반영 및 함수 일부 구현 (현소담, 방준식)
	
Date ~12/02	Date ~12/02
Summary <ul style="list-style-type: none"> • 모델 정확도 향상 • 안드로이드에 딤러닝 모델 적용 • 모듬회 인식을 낮음 → 추가학습 필요 	Summary <ol style="list-style-type: none"> 1. 제철 회 안내 기능 구현 <ul style="list-style-type: none"> - 메인 화면에 제철 회가 뜨고, 누르면 상세정보 확인 가능 2. 회 검색 기능 구현 <ul style="list-style-type: none"> - 검색 버튼 누를 시 해당 회의 상세정보 확인 가능 - 검색 결과가 없을 시 이전 검색 결과가 뜨고 있는데.. 결과가 없으면 토스트라도 띄우게 수정해야 할 듯 3. 상세 정보에 설명만 띄우게 되어 있음 <ul style="list-style-type: none"> - Efficacy(효능) Cooking(요리) HowToEat(먹는방법) 추가 필요 <p>• 고민: 딤러닝 수행 시 탐색 결과의 종류가 여러개 나오면 그 개수만큼 두번째 창을 띄울까?</p>

데이터 전처리(안세혁)

Attach any data or images here

Date ~ 12/02

Summary

- 데이터 전처리
- 2가지 회 종류에 대한 이미지 크롤링 및 인덱스 레이블링
- 정확도 향상을 위하여 원인분석

4) ~12/20(최종)

설명	날짜	작성자
main origin/main origin/HEAD Update README.md	20 12 2021 22:47	will980313 <4267
모델, apk 링크 업데이트	20 12 2021 22:45	will980313 <4267
모델 삭제(용량문제로 다른 링크를 통해 다운로드)	20 12 2021 22:30	will980313 <4267
버그 수정	20 12 2021 21:59	will980313 <4267
ui update 4	20 12 2021 19:23	Laika Ahn <83230
사진첩 기능 추가	20 12 2021 17:11	will980313 <4267
ui update 3	19 12 2021 3:40	Laika Ahn <83230
1. 첫 화면 UI 변경 - 기존 텍스트 "제철 레시피"에서 "제철 화: 이름" 으로 변경 2. MainActivity2 찍은 사진이 백그라운드로 뜨게 변경	19 12 2021 1:38	DESKTOP-GTII4S
ui update 2	18 12 2021 23:20	Laika Ahn <83230
ui update test	16 12 2021 22:34	Laika Ahn <83230
detect + UI	16 12 2021 18:15	DESKTOP-GTII4S
검색창 미등록 회 처리, 세부정보 추가	16 12 2021 15:39	Jun_Sik <bjs0104
데이터 추가	9 12 2021 12:49	will980313 <4267
1. 제철 회 안내 기능 구현 - 메인 화면에 제철 회가 뜨고, 누르면 상세정보 확인 가능	2 12 2021 6:26	DESKTOP-GTII4S

- 디자인 최종 완성
- 최종 기능 구현 완료
- 최종 모델 구현 완료

5. 구현 내용 및 최종 결과물

5.1 최종 결과물

- PyTorch로 만들어진 AI 모델을 Android Studio로 제작한 Application에 올려 제작하였고 Application의 전체적인 디자인은 최근 미니멀 디자인 보다는 프로젝트의 주제가 되는 회의 이미지를 돋보이게 하기 위하여 SNS 형태의 레이아웃을 선택하였다.

5.2 시연

- 시연 정보는 최종 동영상과 사용 매뉴얼에 포함

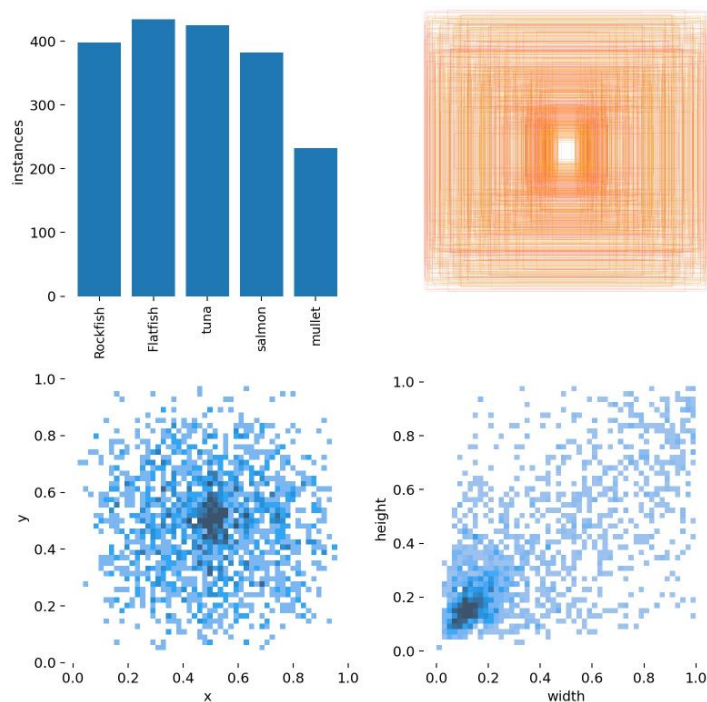
5.3 소스 소개

- 모든 소스코드는 github에 업로드 되어져 누구나 확인, 활용 가능하다.

- <https://github.com/dami138/loveSashimi>

5.4 테스트 과정 및 결과

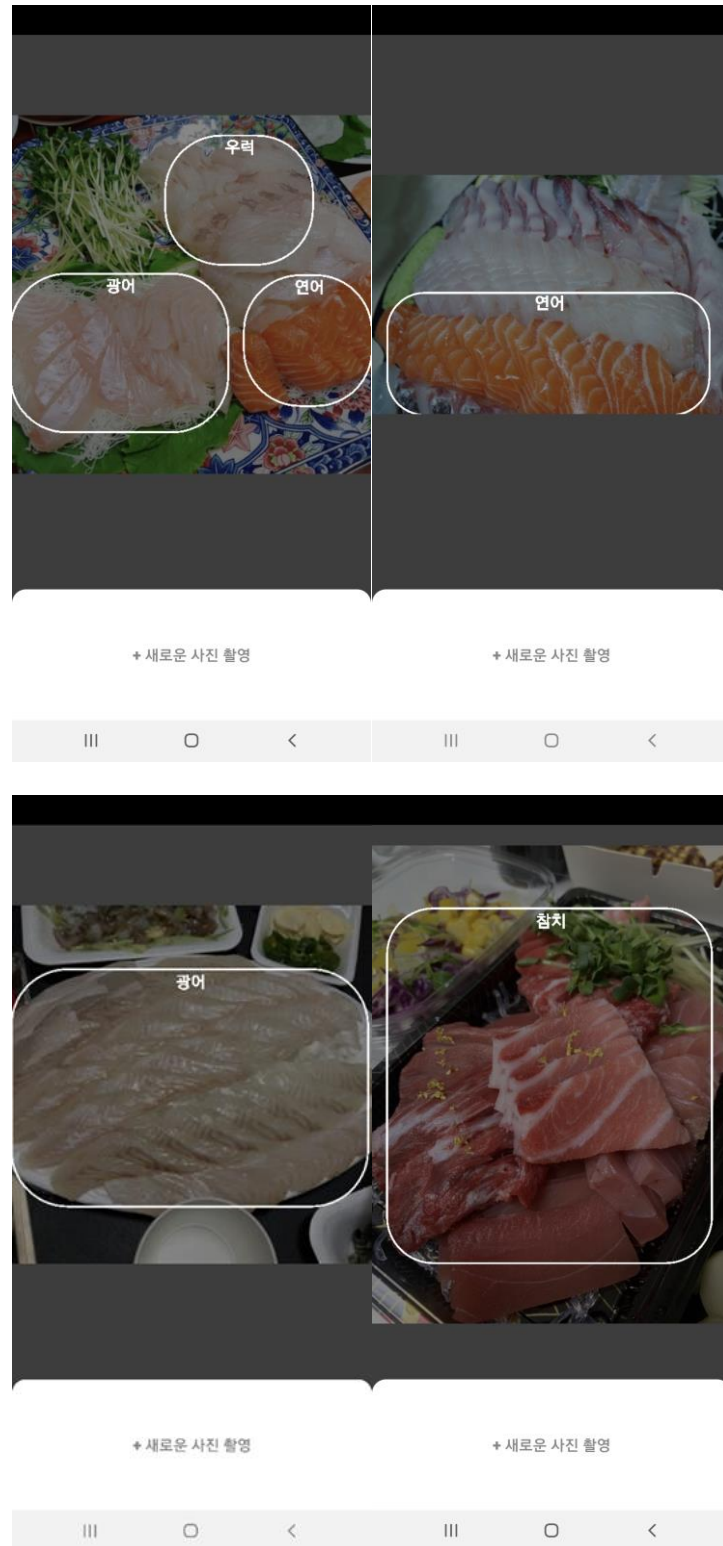
1) 학습 객체 수



2) 학습 결과



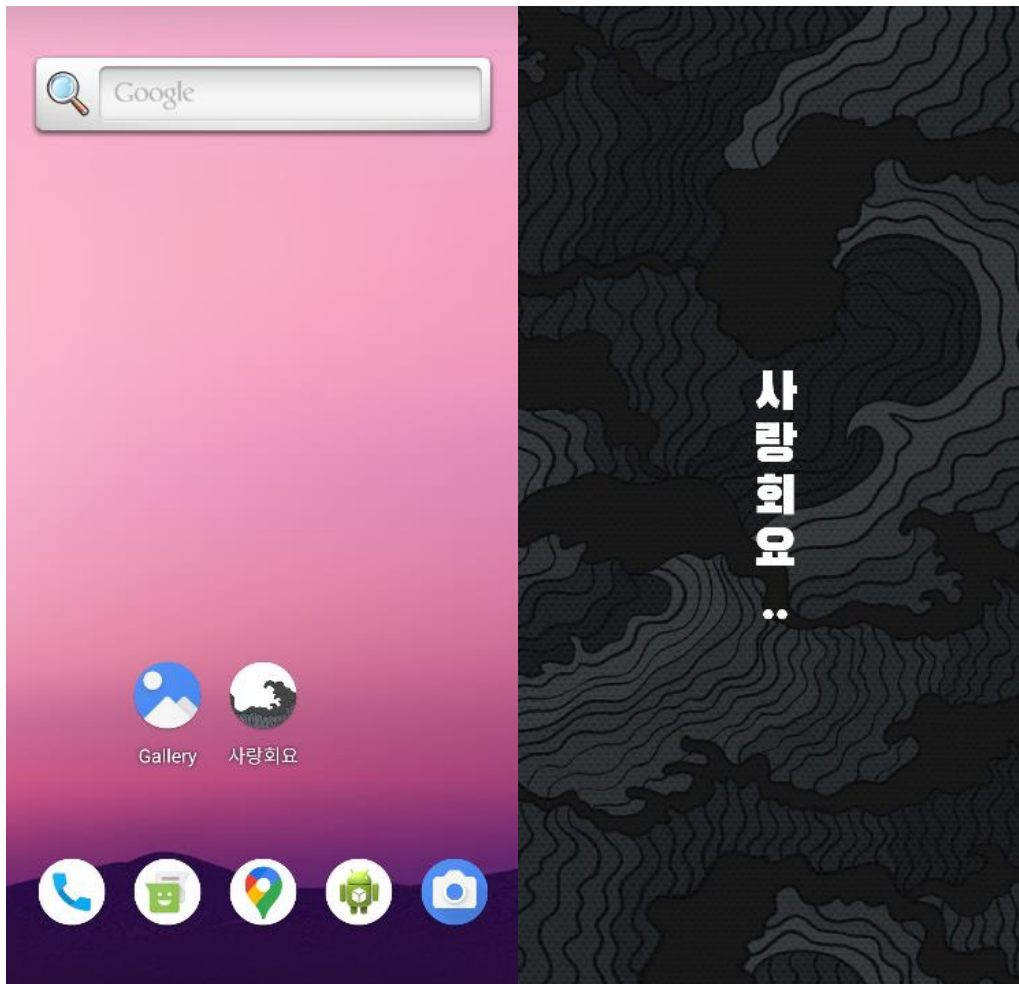
- 사진에 다른 객체들이 나오는 것을 최소화하였을 때 회 인식률을 상당히 높다.
- 우측 상단 이미지(sal4.jpg)는 참치와 연어로 동시에 예측했지만 연어의 점수가 0.7로 더 높으므로 실제 출력 시 연어로 예측한다.



- 모듬회의 인식률은 단일 종류의 회보다 낮지만 양호한 인식률을 보인다. 위 사진들은 모바일에서 yolov5모델을 사용한 장면이다.

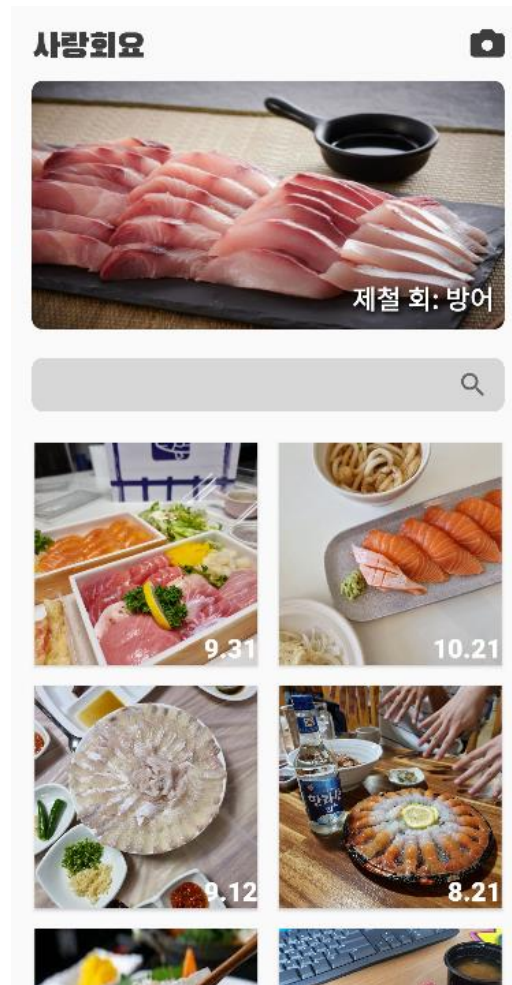
5.5 사용 매뉴얼

1) 아이콘 터치 및 어플 실행



- 좌: '사랑회요'의 Application 아이콘을 터치하여 실행한다.
- 우: 어플 실행 시 Application의 Splash 이미지와 함께 로딩되는 것을 확인 가능하다.

2) 메인 페이지



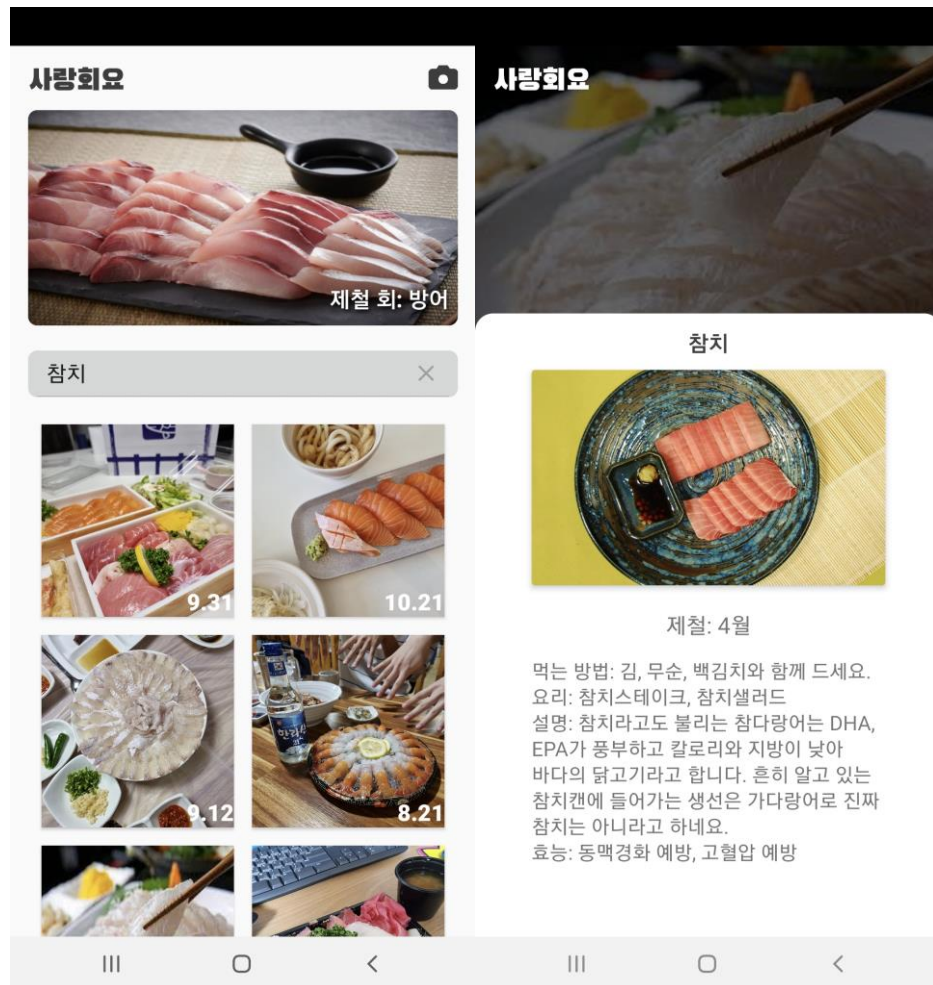
- 중앙 상단의 뉴스 윈도우를 통하여 현재 계절에 맞는 제철 회를 확인할 수 있으며, 그 밑으로 검색 기능과 사용자 회 탐방 다이어리를 바로 확인 가능하다.
- 프로젝트의 메인 기능인 AI 모델을 통한 회 사진 검색 기능은 우측 상단의 카메라 아이콘을 터치하는 것으로 사용 가능하다.

3) 제철 회 정보 조회 화면



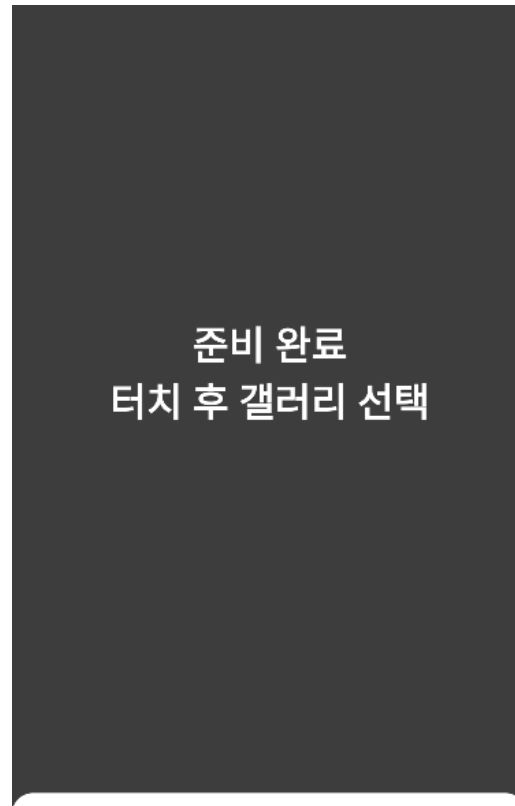
- 메인 화면에서 중앙 상단의 제철 회 윈도우를 누를 시 나타나는 화면이다.
- 사용자는 계절에 맞는 회 추천 정보를 확인할 수 있다.
- 지금은 12월이므로 제철이 12월인 방어의 정보를 확인할 수 있다.

4) 회 이름 검색 및 검색 결과 화면



- 좌: 메인 화면에서 참치를 검색하는 화면
- 우: 검색 결과인 참치의 정보를 확인하는 화면.

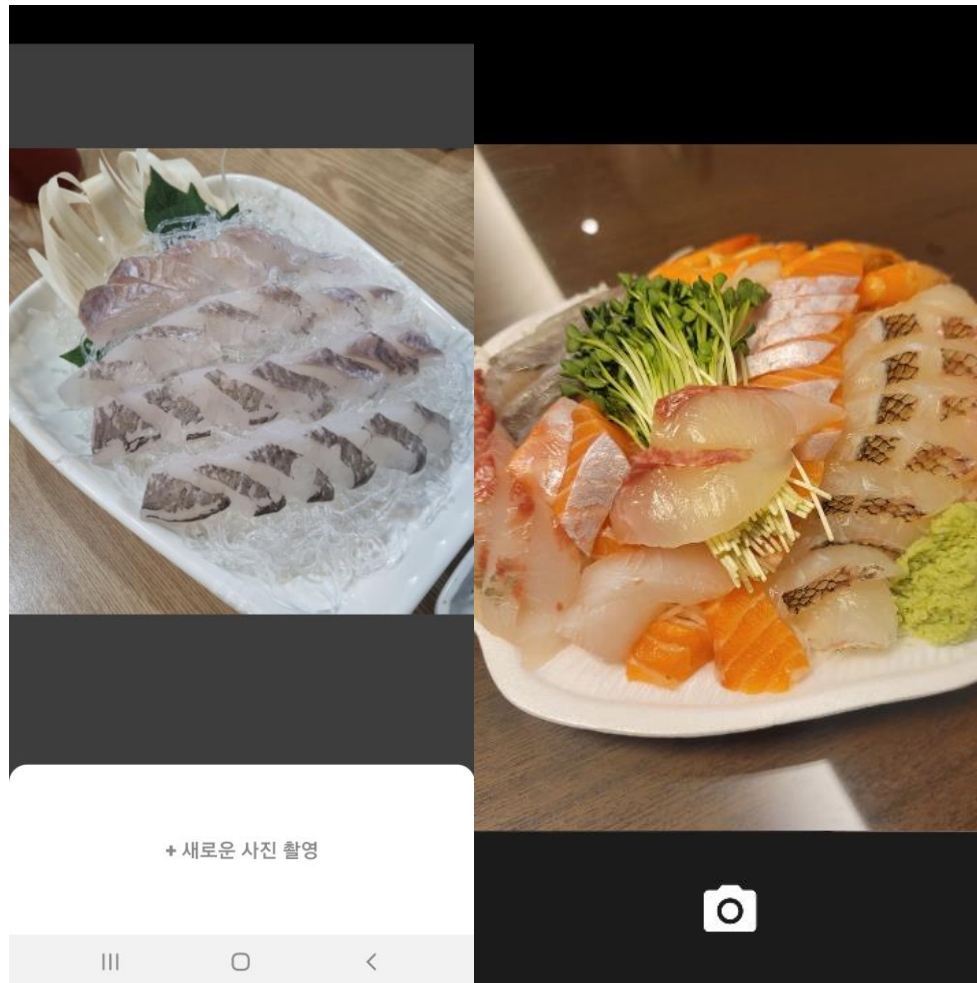
5) Cam 액티비티 화면



+ 새로운 사진 촬영

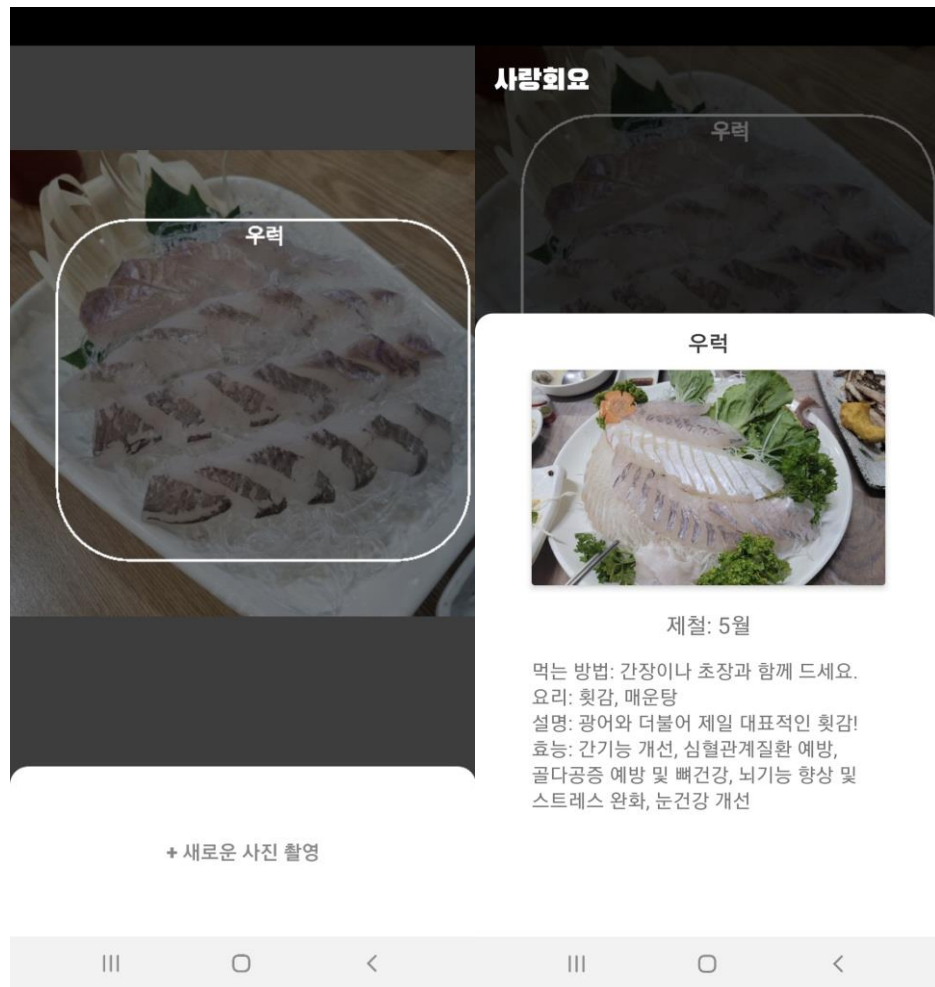
- AI 모델이 준비된 Cam 액티비티 화면이다.
- 사용자는 회색 화면을 터치하여 기존 갤러리의 사진을 불러오거나, 하단의 새로운 사진 촬영을 선택하여 새로운 회 사진을 촬영할 수 있다.

6) 갤러리 사진 불러오기 및 새 사진 촬영 화면



- 좌: 갤러리에서 기존 촬영된 사진을 불러온 모습.
- 우: 새로운 회 사진을 촬영하기 위해 카메라 모듈을 불러온 모습.
- 이 단계에서 잠시 대기하면 AI 모델이 분석 후 결과를 알려준다.

7) AI 분석 결과 화면



- 좌: AI 분석이 완료된 화면, 우럭이 잘 검출되었다.
- 우: Dimming(어둡게 입히는 필터)된 이미지는 사용자가 분석한 회 사진을 보여주며, 하단의 오버레이된 레이아웃에 AI 모델이 분석하여 얻어낸 회의 정보를 출력한다.
모듬회를 찍은 경우와 같이 2개 이상의 회가 검출되었을 때에는 회 종류별로 오버레이를 띄우게 되며, 뒤로 가기를 누르면 오버레이가 하나씩 사라지게 된다.

5.6 목표 달성 여부

1) 학습 데이터 수집

- Selenium을 통해 학습에 필요한 회 데이터들을 적절히 수집하였다.

2) 모델 학습

- YoloV5를 통해 회 인식 및 분류를 위해 정확도 높은 모델을 구현하였다.

3) 데이터베이스 구축

- Firebase를 통해 사용자에게 제공할 회 정보 데이터베이스를 구축하였다.

4) 어플리케이션 제작

- Android Studio 환경에서 안드로이드 어플리케이션을 제작하였다.
- 회 사진 기반 검색, 회 이름 기반 검색, 제철 회 조회 기능을 구현하였다.
- UI를 사용자 친화적으로 구성하였으며, 사용자의 편의성을 잘 고려하였다.

5) 버전 관리 및 소통

- Github을 통해 팀 내 버전 관리를 진행하였다.
- 카카오톡과 알로를 통해 팀원과 항상 원활한 소통을 이루었다.

6. 기타

6.1 한계 및 고찰

1) 기능 관련

- 카메라 모듈을 통해 촬영을 할 시, 촬영된 사진이 내부 저장소(갤러리)에 저장되게 하지 못하여 다이어리 기능은 추후 계획으로 결정하였다.
- 캐시 영역에 저장되게는 할 수 있었는데, 이와 같은 경우 어플 재실행시 초기상태로 되돌아가기 때문에 기능을 제대로 구현할 수 없었다.

2) 학습 관련

- 단일 회 인식은 상당히 정확했으나, 모듬회는 면적이 좁은 경우 낮은 정확도를 보여주었다



- 서버에서 예측한 경우, 결과가 어느 정도 정확했으나 휴대폰에서 실행 시 성능 저하가 있었다.

좌(서버): 상대적으로 면적이 넓은 광어와 연어는 감지되었지만 밀치는 인식하지 못하였다.

우(휴대폰): 모바일 환경에서 실행해 보았으나 아무 객체도 인식하지 못하였다.

6.2 발전 방향 및 향후 계획

- 휴대폰에서 GPU서버로 사진을 인풋하여 결과를 받아오는 방법을 사용하면 될 것이라 생각했다. 하지만 여러 유저가 접속하였을 경우 서버에서 처리할 코드를 작성하여야하고 고화질 이미지 업로드시 시간이 걸리는 등 문제가 있었다.
- 카메라 촬영 시 촬영한 사진이 내부 저장소에 저장될 수 있도록 수정할 계획이다. 내부 저장소에 저장 성공 시 다이어리 기능도 활성화 가능하다. (저장된 기록을 통해 사진과 날짜를 불러와 다이어리에 표시)
- 핸드폰에서도 컴퓨터와 같은 좋은 성능을 나타내도록 최적화할 계획이다.
- 향후 주변 횃집 추천, 배달 기능도 추가하여 서비스를 제공해볼 생각이다.
- 자신이 촬영한 사진과 정보를 SNS에 공유할 수 있는 기능을 추가해볼 계획이다.
- 더 많은 학습을 통해 이미지 인식률을 높이고 다른 회 정보도 추가하여 종류를 늘릴 계획이다.

6.3 참고문헌

Firebase https://firebase.google.com/docs/firestore/query-data/get-data?utm_source=studio

YoloV5 <https://github.com/ultralytics/yolov5>

tfLite <https://www.tensorflow.org/lite/guide/android?hl=ko>

Pytorch <https://pytorch.org/mobile/android/>