

2023-1 AI 융합학부 캡스톤 디자인

매일 건강해질 나를 위한 기록 ———

# 다일일지 혈당관리앱

2023학년도 1학기 캡스톤 디자인 프로젝트 중간 보고서

AI 융합학부 건강최고당

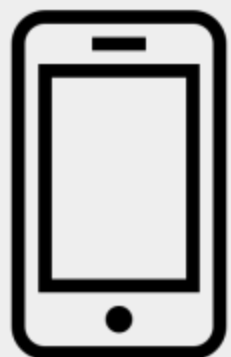
김예린 김혜빈 우미경 이하늘

AI 융합학부 20192897

김혜빈

## Contents

---



### 서비스 소개

핵심 서비스 소개



### UI 세부 설계

화면 구성 및 기능 소개



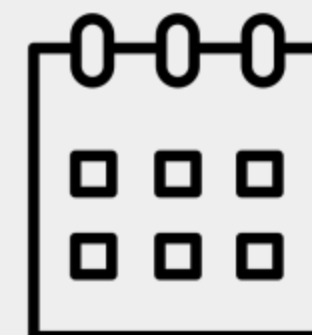
### 시스템 구조 설계

전체 시스템 구조



### 프로젝트 진행 현황

구현할 기능에 대한 소개



### 프로젝트 계획

프로젝트 진행 계획

## 핵심 서비스 소개

# 당일일지

- 식단 / 혈당 기록 동시 조회 가능
- 식단 기록 시 영양 성분 정보 자동 계산
- 음식 이미지 등록으로 음식 종류 및 영양 성분 정보 자동 설정

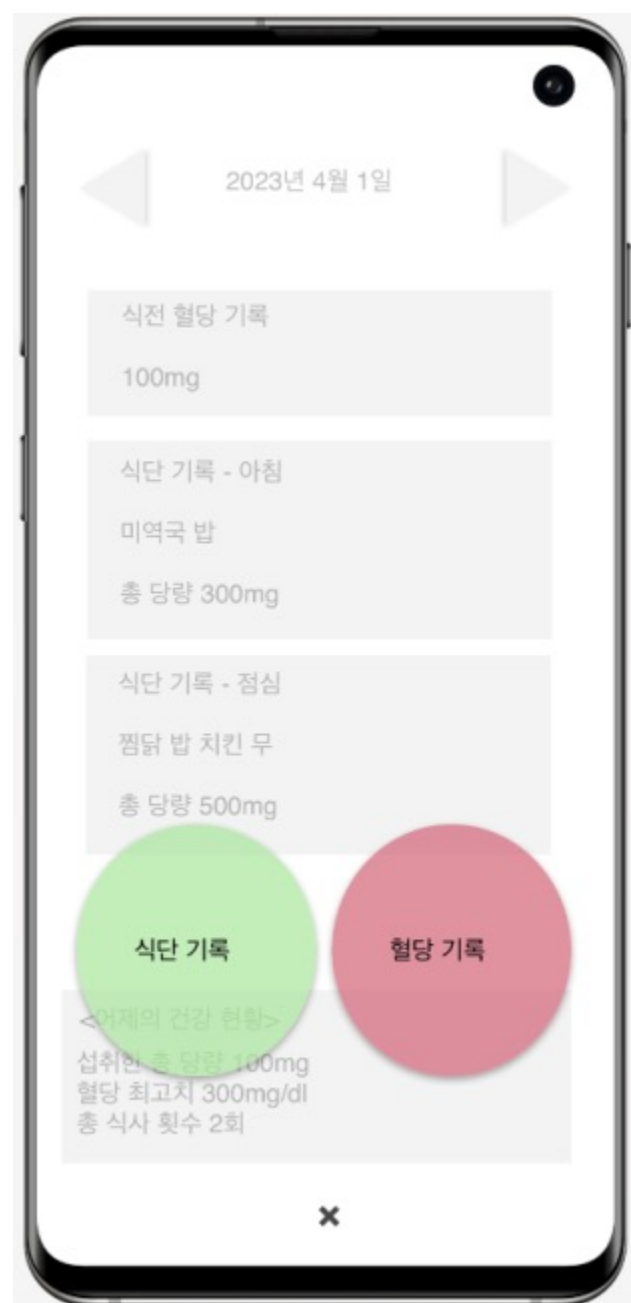
기록

조회

자동화



## 메인 화면



### 메인 뷰 구성

- 시간 순서대로 식단과 혈당 구분 없이 하나의 타임라인으로 제공
- 혈당 변화와 관련된 식단 바로 확인 가능
- 전날 기록 요약 제공

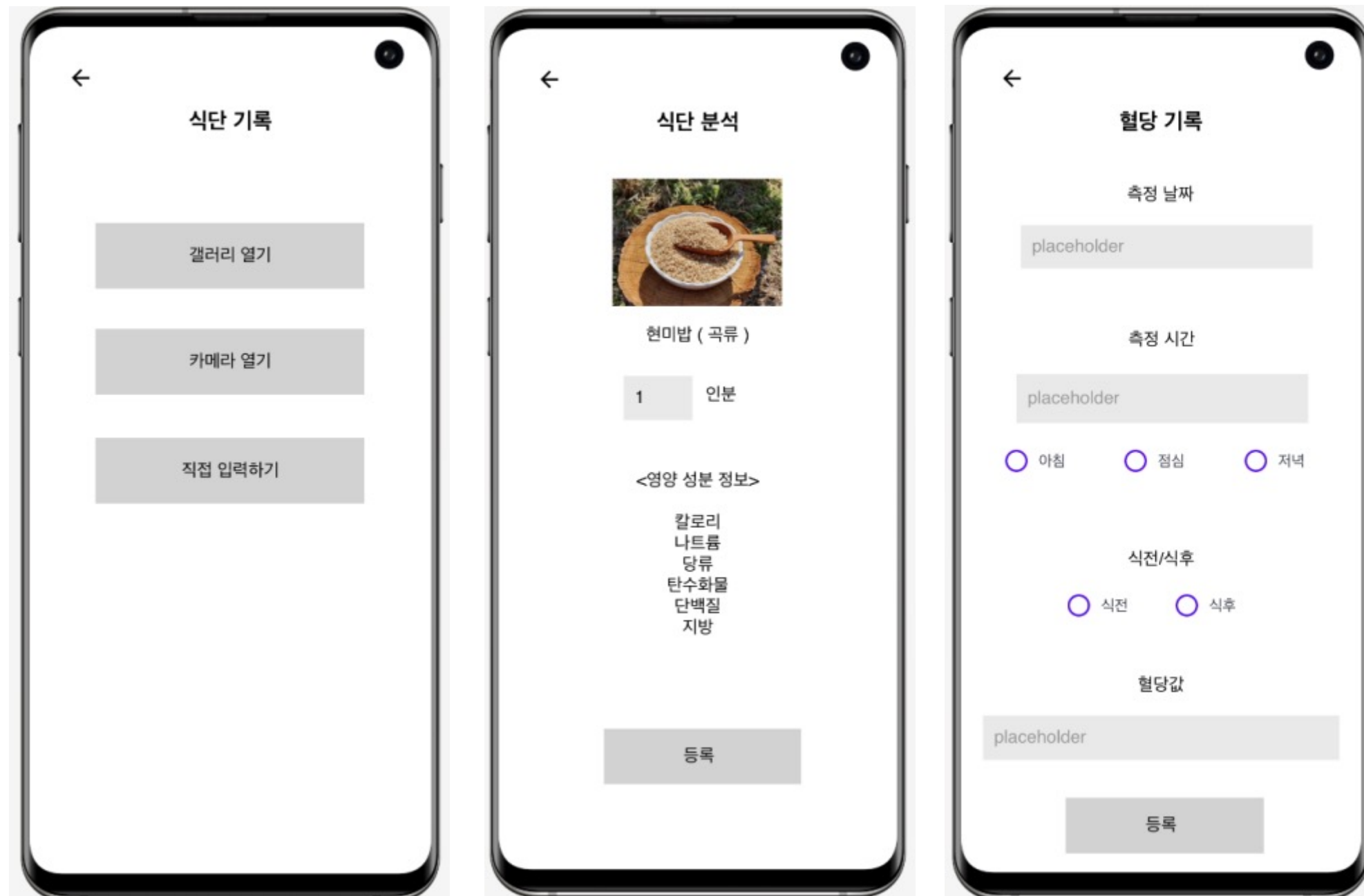
### 하단 네비게이션바 구성

- 식단 메뉴와 혈당 메뉴에서 각각의 상세 기록 확인 가능
- 기록 메뉴에서 식단과 혈당 기록 가능

### 식단 / 혈당 기록

- 식단 기록과 혈당 기록으로 분류 나누어 등록

## 식단 / 혈당 기록 화면



### 식단 기록

- 이미지를 활용한 식단 기록 : 음식 이미지 업로드
- 텍스트를 활용한 식단 기록 : 음식 이름 직접 입력

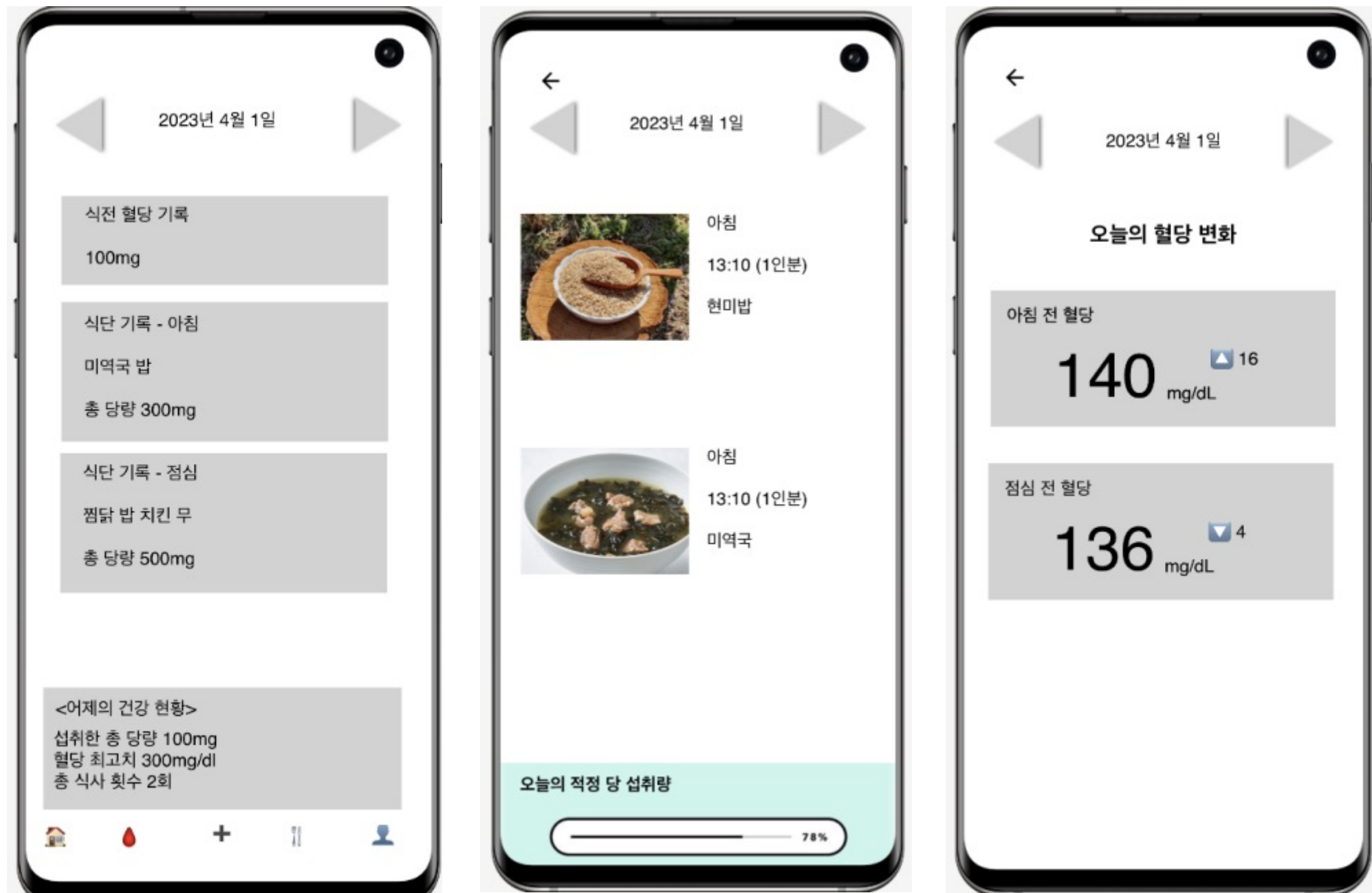
### 식단 분석

- 이미지 분류를 통해 음식 이름 자동 설정
- 음식량 직접 입력
- 음식 이름으로 영양 성분 정보 찾아 자동 계산

### 혈당 기록

- 측정 시간은 현재 시각을 기본값으로 사용
- 혈당값 직접 입력

## 식단 / 혈당 조회 화면



### 전체 기록 조회

- 날짜별 기록한 식단 정보와 혈당 정보 한 번에 조회 가능

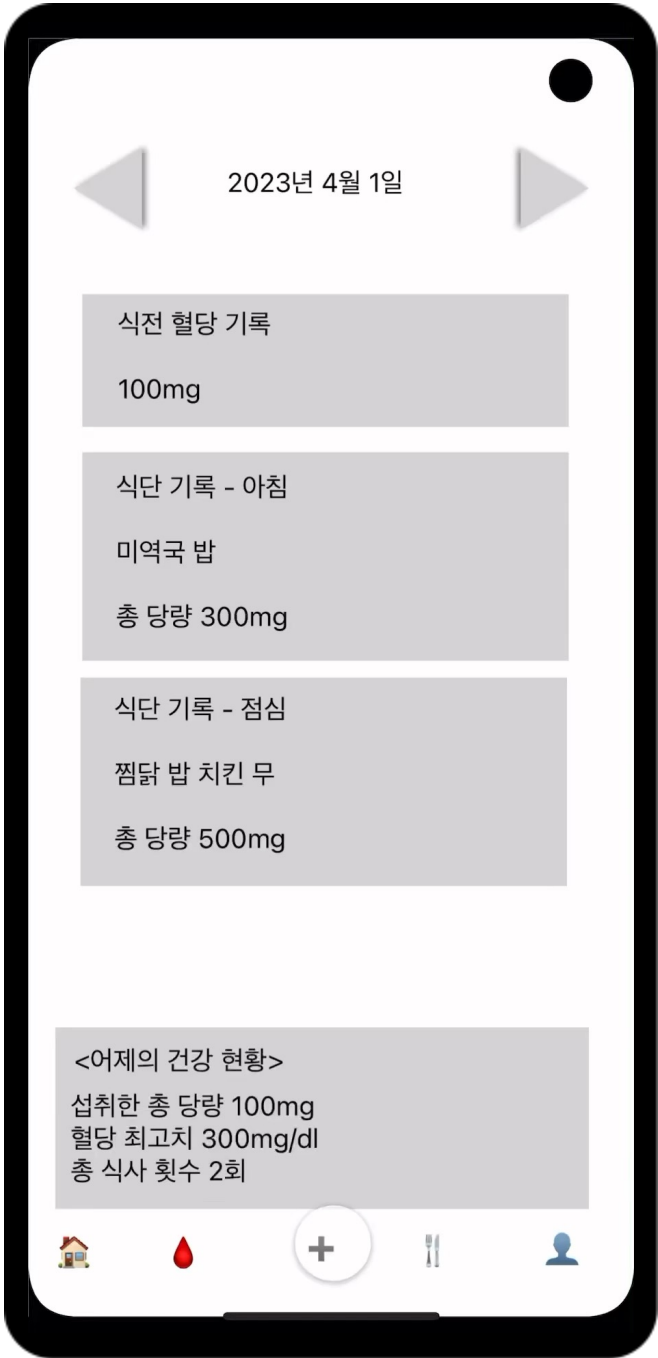
### 식단 조회

- 기록 시간 순으로 식단 조회 가능
- 혈당 관리를 위해 하루 당 섭취량을 하단에 보여줌
- 각각의 블록 선택 시 상세 영양 정보 확인 가능

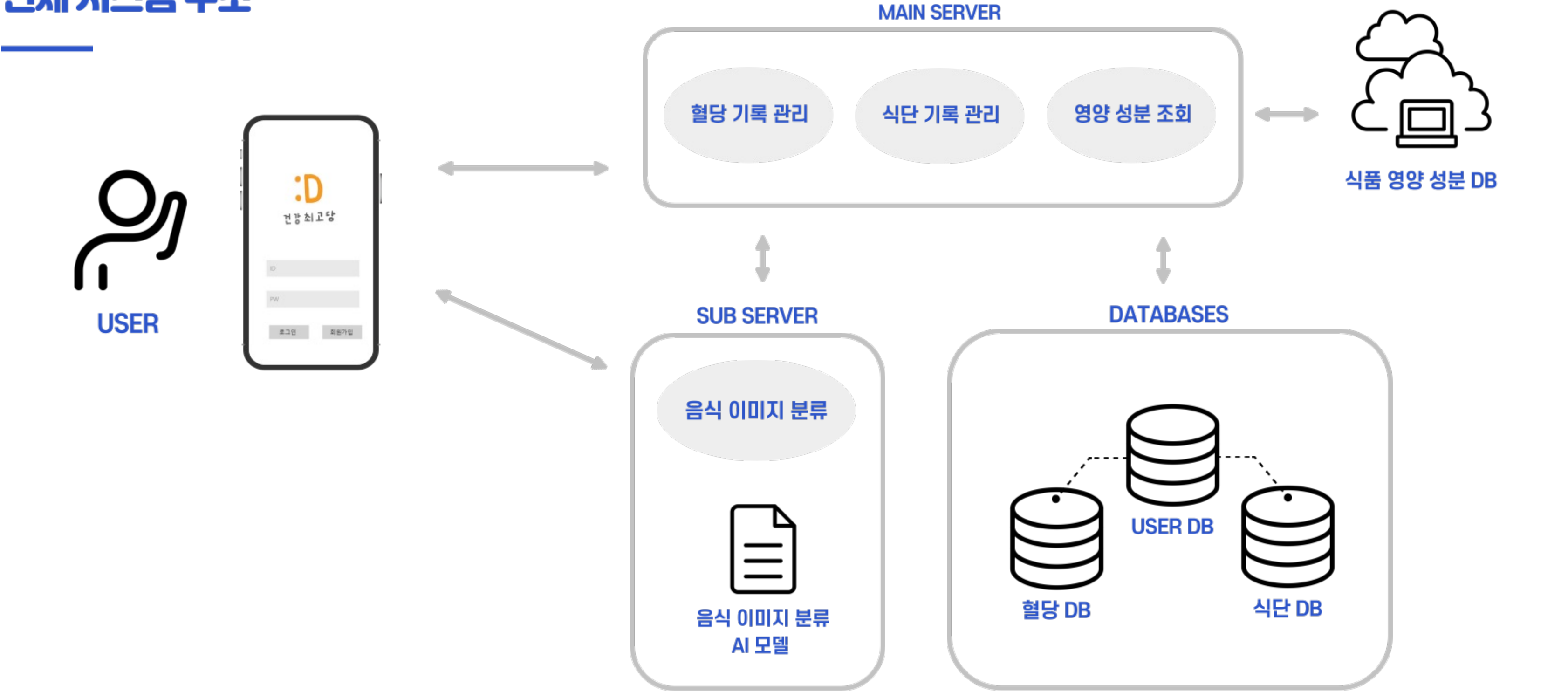
### 혈당 조회

- 기록 시간 순으로 혈당 조회 가능
- 이전 혈당 값과 비교하여 증감량 계산하여 보여줌

# 프로토타입 시연 영상

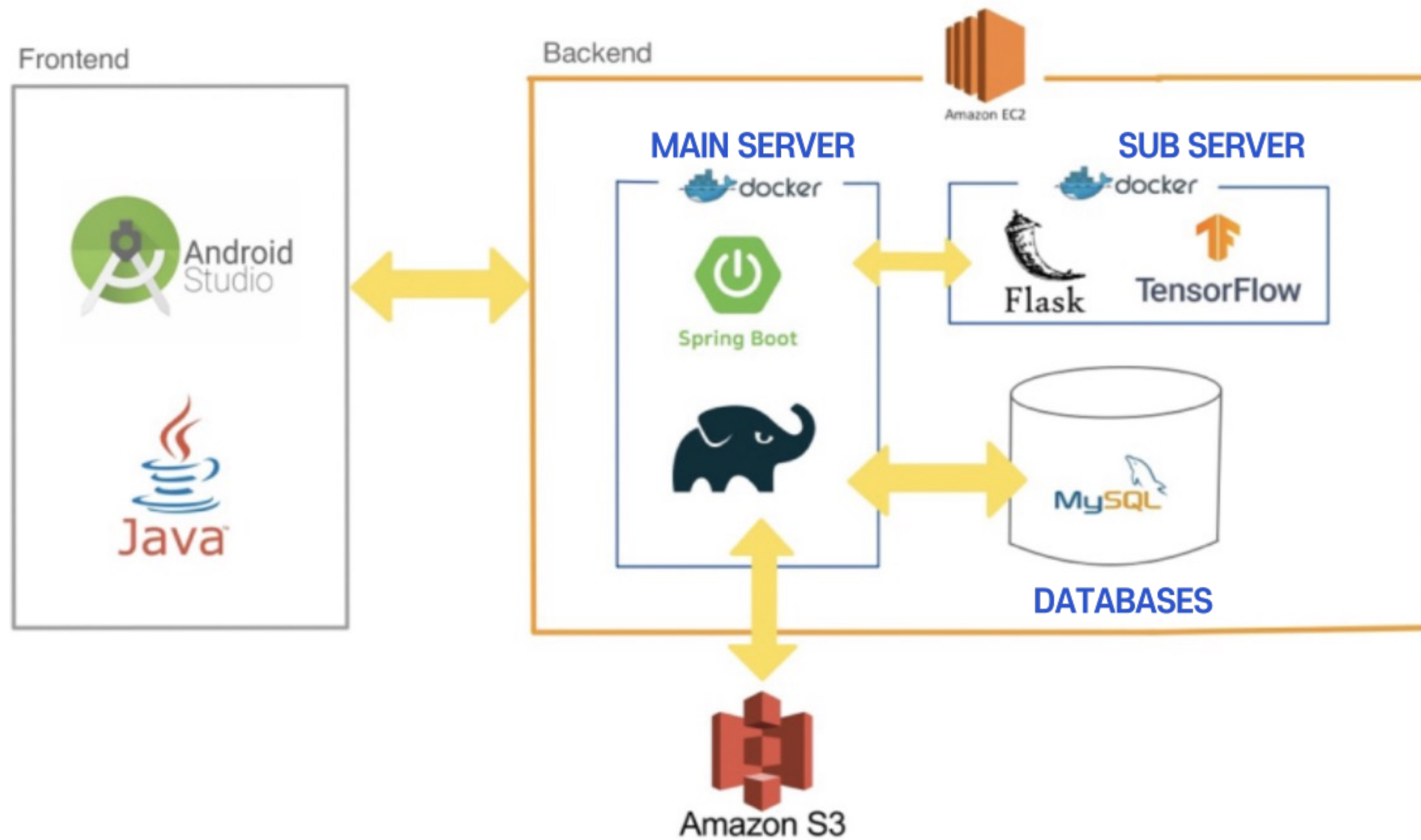


전체 시스템 구조

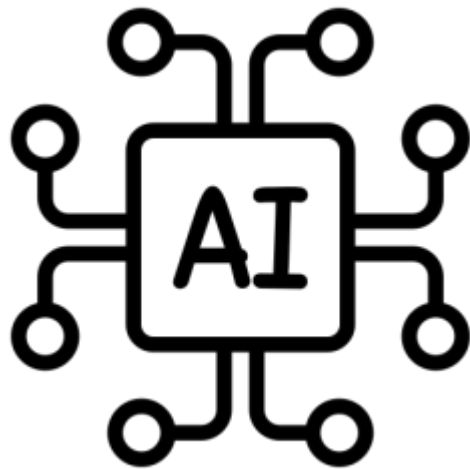




## 시스템 구현 기술



한식 분류 AI 모델



맞춤형 모델의 필요성

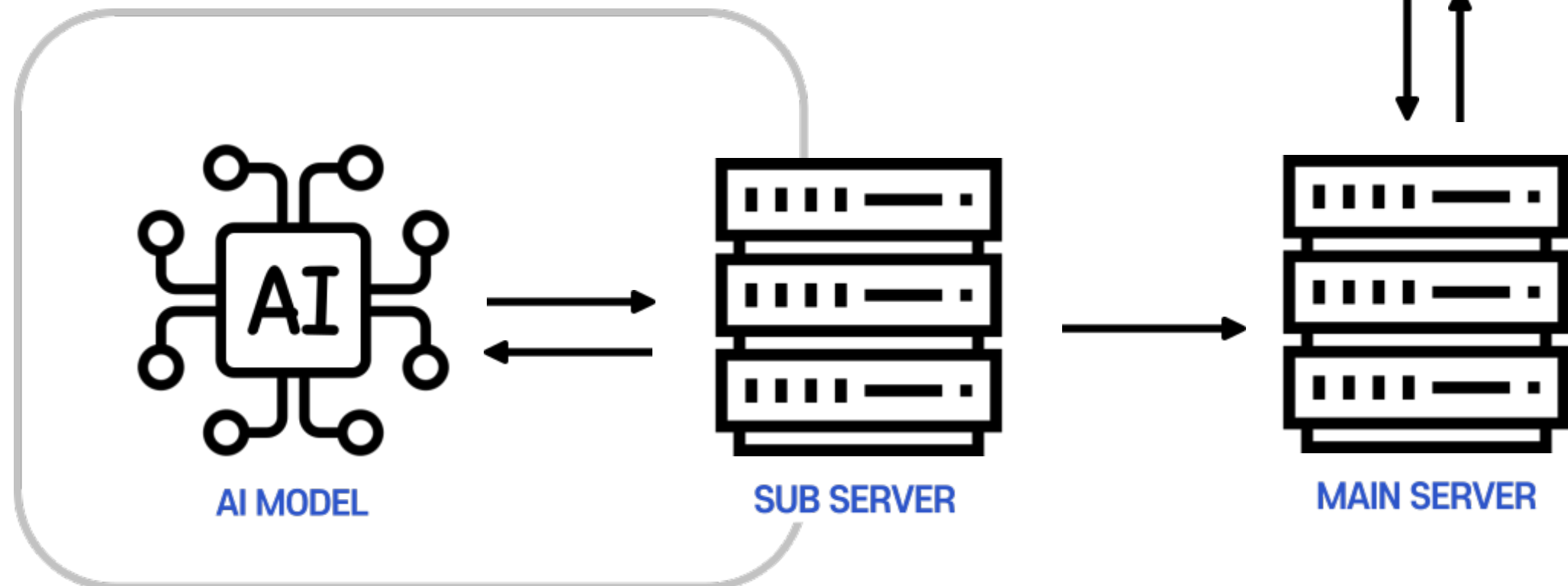
- 1 한식 이미지 분류 모델의 필요성**  
음식 이미지 분류 모델은 많았지만,  
한식 종류가 포함되어 있는 경우는 적어 활용이 어려움
- 2 활용 가능한 리소스의 한계**  
모델 크기가 너무 클 경우 서버에서 활용하는데 한계가 있음

한식 이미지 데이터 + 기존 이미지 분류 AI 모델

## 한식 분류 AI 모델 활용

### Input

- Sub server -> AI model
- 확장자 .jpg의 음식 이미지 파일
- 사용자가 업로드 한 이미지 전달 받음



### Output

- AI model -> Sub server
- 음식 이름을 결과값으로 반환

Sub server부터 다음의 과정을 따라  
최종적으로 식품 영양 정보를 얻는 것을  
목표로 함

- Sub server -> Main server  
: 결과값 전달
- Main server -> 식품 영양 성분 DB  
: OpenAPI를 통해 음식 이름으로 전달
- 식품 영양 성분 DB -> Main server  
: 최종적으로 식품 영양 정보 가져옴



이미지 데이터





#한식 이미지 #한국 음식

### 한국 이미지(음식)

분야 영상이미지 유형 이미지

갱신년월 : 2023-03 구축년도 : 2018 조회수 : 5,076

다운로드

샘플 데이터



kfood > 구이 > 갈비구이		
이름	크기	유형
crop_area.properties	7KB	PROPERTIES 파일
img_000_0000.jpg	129KB	이미지(jpg) 파일
img_000_0001.jpg	56KB	이미지(jpg) 파일
img_000_0002.jpg	226KB	이미지(jpg) 파일
img_000_0003.jpg	73KB	이미지(jpg) 파일
img_000_0004.jpg	129KB	이미지(jpg) 파일
img_000_0005.jpg	349KB	이미지(jpg) 파일
img_000_0006.jpg	155KB	이미지(jpg) 파일
img_000_0007.jpg	129KB	이미지(jpg) 파일
img_000_0008.jpg	71KB	이미지(jpg) 파일

이미지 데이터 정보

- 음식 종류 : 150 종
- 종류별 이미지 수 : 1000장 이상
- 데이터 포맷 : 이미지 포맷  
(jpg, png, jpeg, JPG, gif, ... 등)
- 이미지 파일명 규칙  
: img\_(소분류코드)\_(이미지번호).(이미지 확장자)
- 파일 구조  
: kfood / 음식 대분류 / 음식 소분류 / 이미지파일

대분류	소분류	대분류	소분류
구이	갈비구이,갈치구이,고등어구이,곰창구이, 닭갈비,더덕구이,떡갈비,불고기,삼겹살, 장어구이,조개구이,황태구이,훈제오리	국	계란국, 떡국/만두국, 무국, 미역국, 북엇 국, 소고기무국, 시래기국, 육개장, 콩나물 국

## AI 모델 학습 환경 구축

### 사용 언어



### 개발 환경



Google Colaboratory

빠른 이미지 처리를 위해 GPU 가속기 사용

### 이미지셋 및 모델 저장소



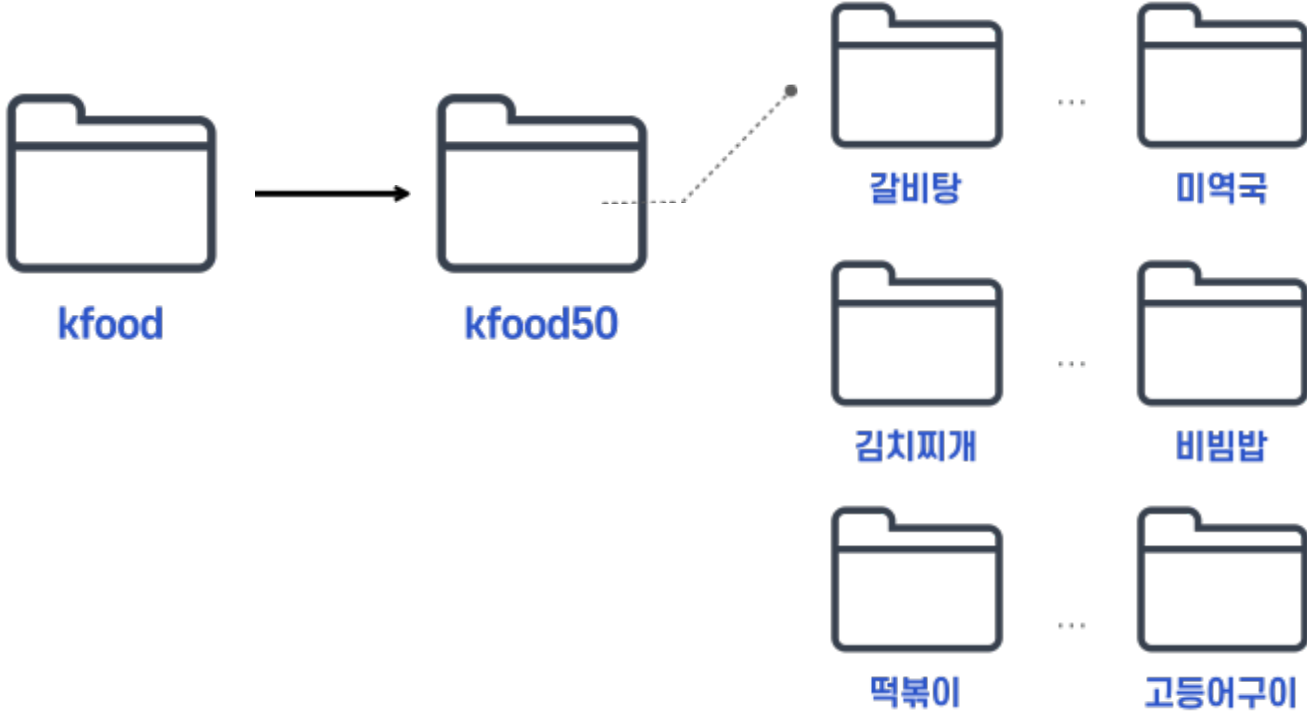
### AI 모델 설계 및 학습



### 이미지 처리



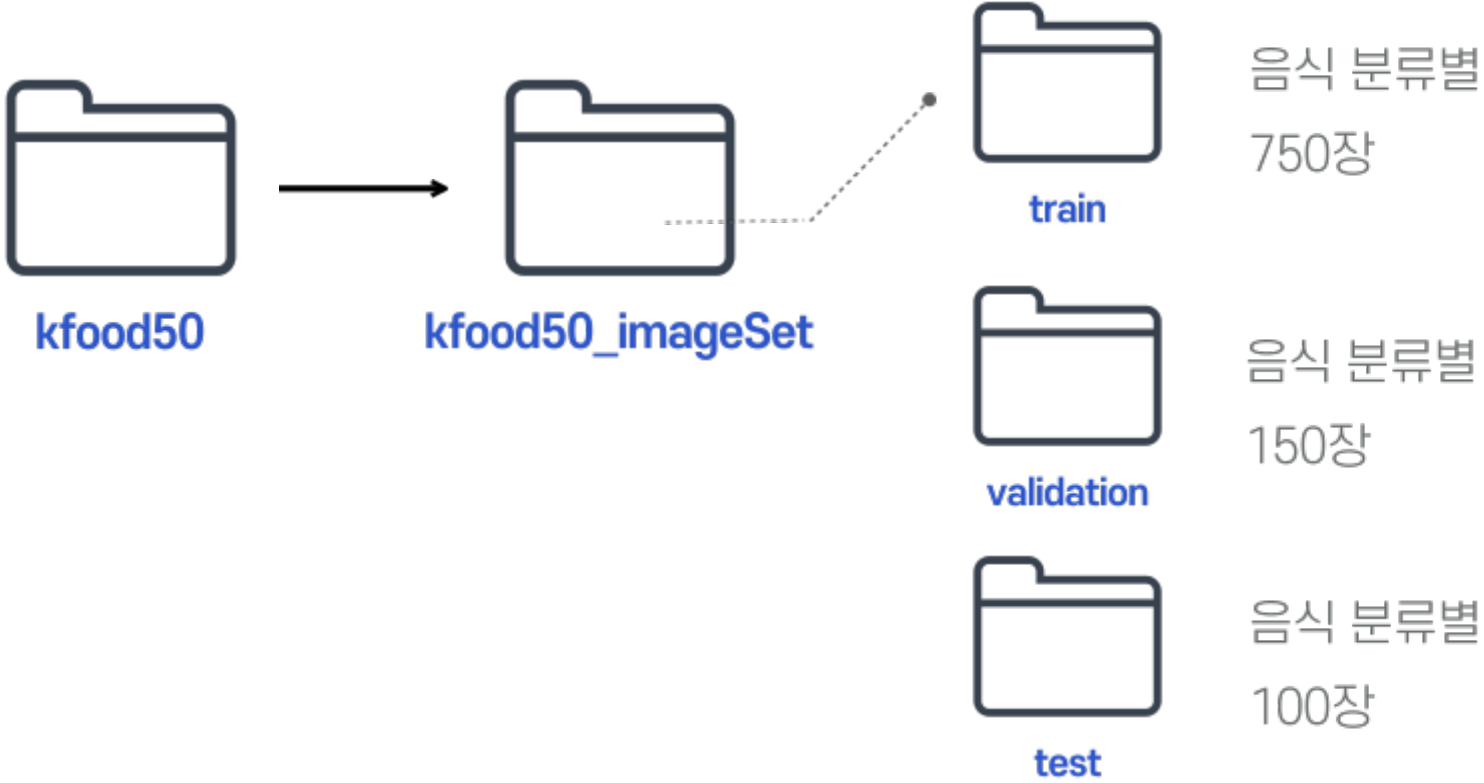
## 이미지 전처리 과정



### 이미지 선별

전체 150 종류의 음식 중 50 종류 음식 선별

- 파일 구조  
: kfood50 / 음식 소분류 / 이미지파일  
대분류 계층 없이 하나의 폴더 내에 음식 소분류 폴더를 담음



### 이미지셋 제작

각 분류별로 train, validation, test로 이미지 나눔

ImageDataGenerator.flow\_from\_directory() 함수 활용시 path 설정을 위함

- 파일 구조  
: kfood50\_imageSet / (train, validation, test) / 음식 소분류 / 이미지파일



## 이미지 전처리 과정

```
for img in train :
    filename, _ = img.split('.')
    imgRGB = Image.open(main_path+food+'/'+img).convert('RGB')
    imgRGB.save(imgSet_path+sub_dir[2]+food+'/'+filename+'.jpg', 'jpeg')
```

```
[ ] datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = datagen.flow_from_directory(
    train_path,
    # shuffle=False,
    target_size=(img_height, img_width),
    batch_size=batch_size_train,
    class_mode='categorical')

validation_generator = datagen.flow_from_directory(
    validation_path,
    # shuffle=False,
    target_size=(img_height, img_width),
    batch_size=batch_size_validation,
    class_mode='categorical')

test_generator = datagen.flow_from_directory(
    test_path,
    # shuffle=False,
    target_size=(img_height, img_width),
    batch_size=batch_size_test,
    class_mode='categorical')
```

```
Found 37500 images belonging to 50 classes.
Found 7500 images belonging to 50 classes.
Found 5000 images belonging to 50 classes.
```

```
train -----
['고등어구이', '떡볶이', '소세지볶음', '달걀국', '육개장', '어묵볶음', '소고기무국', '소고기무국']
<Figure size 640x480 with 0 Axes>
```



메모리 사용량 초과를 방지하기 위해

ImageDataGenerator.flow\_from\_directory() 함수 활용


(함수 인자 중 batch\_size를 지정 시 해당 크기만큼씩 이미지를 메모리에 올림)

ImageDataGenerator.flow\_from\_directory()

RGB 3채널로 받아오는데 4채널로 된 이미지 파일 존재 -> 해당 이미지 가져오지 못함

pillow 라이브러리를 활용하여 RGB 채널에 .jpg 확장자를 갖는 이미지 파일로 변환

# 기존 이미지 분류 AI 모델



1. ResNet (Residual Neural Network):

ResNet은 이미지 분류를 위한 딥 러닝 모델로, 매우 깊은 신경망을 효과적으로 학습할 수 있도록 설계되었습니다. ResNet은 이전의 모델들이 가지고 있던 기울기 소실 문제를 해결하기 위해 Residual Block이라는 새로운 아키텍처를 도입했으며, 이로 인해 매우 깊은 신경망도 학습할 수 있게 되었습니다.

2. Inception (GoogLeNet):

Inception 모델은 컴퓨터 비전 분야에서 사용되는 딥 러닝 모델로, 입력 이미지를 여러 개의 필터로 분리하고 병렬로 처리하는 아키텍처를 가지고 있습니다. 이를 통해 모델의 성능을 향상시키고, 모델의 크기와 복잡도를 줄일 수 있습니다.

3. MobileNet:

MobileNet은 모바일 기기에서 사용하기 위해 설계된 경량화된 딥 러닝 모델입니다. 이 모델은 컨볼루션 계산을 효율적으로 수행하는 Depthwise Separable Convolution 레이어를 사용하여 모델 크기를 크게 줄이면서도 성능을 유지할 수 있습니다.

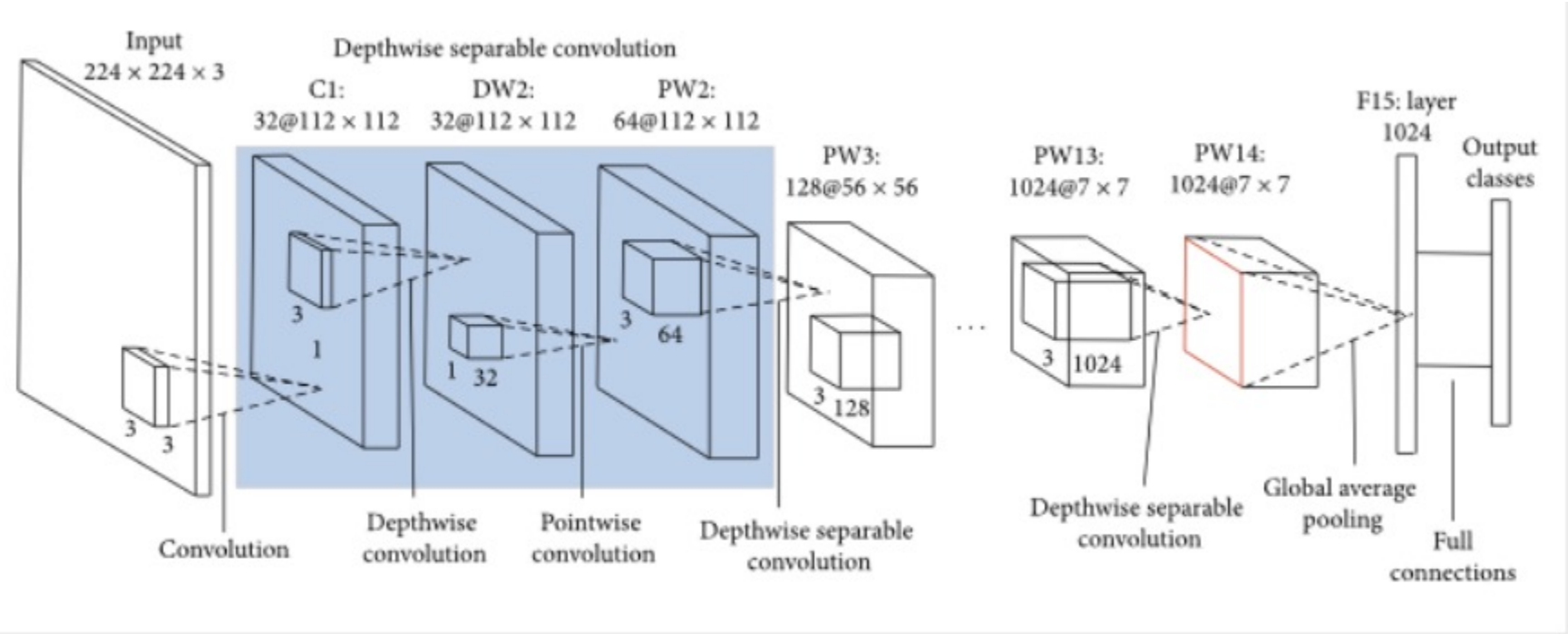
4. EfficientNet:

EfficientNet은 모델의 정확도와 효율성을 모두 고려한 모델로, 기존 모델들과는 달리 모델의 깊이, 너비, 해상도를 동시에 조절하면서 최적화된 모델을 생성합니다. 이를 통해 모델의 크기와 계산 복잡도를 최적화하면서도 높은 성능을 유지할 수 있습니다.

5. SqueezeNet:

SqueezeNet은 크기가 작지만 성능이 좋은 딥 러닝 모델로, 2016년 ILSVRC 대회에서 가장 작은 크기로 우승을 차지한 모델입니다. SqueezeNet은 1x1 필터와 3x3 필터를 적극적으로 사용하여 모델의 크기를 줄이면서도 성능을 유지합니다.

## MobileNet



목표 예측 정확도 : 0.7 이상

학습 진행 후 예측 정확도와 예측 시간 등을 고려하여 하나의 모델 선택  
서버에서 사용 가능한 리소스 양도 고려해야 함



## 프로젝트 진행 계획

### AI 모델 학습

준비한 이미지셋으로 학습

### AI 모델 예측 테스트

모델별 예측 정확도 비교

### AI 모델 최종 선택

최종 서버에 올릴 모델 선택

### 서버 내 테스트

서버 위에서의 예측 성능 테스트



## 출처 자료

---

아이콘 활용 : <https://www.iconfinder.com/>

p.10, p.12 한식 이미지 데이터 : <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=79>

p.13 사용 기술 아이콘

python : [https://www.google.com/url?](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.python.org%2Fcommunity%2Flogos%2F&psig=AOvVaw210t4EFNNDXoFk-zjdiiQq&ust=1681630380084000&source=images&cd=vfe&ved=0CBMQjhxqFwoTCIjRhsSvq_4CFQAAAAAdAAAAABAD)

[sa=i&url=https%3A%2F%2Fwww.python.org%2Fcommunity%2Flogos%2F&psig=AOvVaw210t4EFNNDXoFk-](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.python.org%2Fcommunity%2Flogos%2F&psig=AOvVaw210t4EFNNDXoFk-zjdiiQq&ust=1681630380084000&source=images&cd=vfe&ved=0CBMQjhxqFwoTCIjRhsSvq_4CFQAAAAAdAAAAABAD)

[zjdiiQq&ust=1681630380084000&source=images&cd=vfe&ved=0CBMQjhxqFwoTCIjRhsSvq\\_4CFQAAAAAdAAAAABAD](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.python.org%2Fcommunity%2Flogos%2F&psig=AOvVaw210t4EFNNDXoFk-zjdiiQq&ust=1681630380084000&source=images&cd=vfe&ved=0CBMQjhxqFwoTCIjRhsSvq_4CFQAAAAAdAAAAABAD)

colaboratory : [https://www.google.com/url?sa=i&url=https%3A%2F%2Fruslanmv.com%2Fblog%2FHow-to-connect-Google-Colab-to-your-](https://www.google.com/url?sa=i&url=https%3A%2F%2Fruslanmv.com%2Fblog%2FHow-to-connect-Google-Colab-to-your-computer&psig=AOvVaw0MMqmkxUmxNYAjfJx_gHuR&ust=1681630458077000&source=images&cd=vfe&ved=2ahUKEwiOgYbir6v-AhUBgFYBHcfKD3kQr4kDegUIARDGAQ)

[computer&psig=AOvVaw0MMqmkxUmxNYAjfJx\\_gHuR&ust=1681630458077000&source=images&cd=vfe&ved=2ahUKEwiOgYbir6v-AhUBgFYBHcfKD3kQr4kDegUIARDGAQ](https://www.google.com/url?sa=i&url=https%3A%2F%2Fruslanmv.com%2Fblog%2FHow-to-connect-Google-Colab-to-your-computer&psig=AOvVaw0MMqmkxUmxNYAjfJx_gHuR&ust=1681630458077000&source=images&cd=vfe&ved=2ahUKEwiOgYbir6v-AhUBgFYBHcfKD3kQr4kDegUIARDGAQ)

**감사합니다**