중간보고서

2024년도 전기 졸업과제

학부	정보컴퓨터공학부
팀명	smartPlate
이름	이혁재(201924550)
	문성재(202155645)
	김상해(202155643)

August 21, 2024

Contents

1 과제 목표

이번 프로젝트를 통해 기존의 음식 종류 인식 서비스에서 더 나아가 음식의 양을 측정하여 그에 따른 영양 정보를 제공해주는 서비스를 개발하고자 한다. 이를 통해 식단을 관리하고자 하는 사용자들에게 섭취하는 식품의 영양 정보를 제공하고자 한다.

2 요구 조건 및 제약 사항 분석에 대한 수정사항

2.1 기존 요구조건 및 수정사항

2.1.1 기존 요구조건

- 식판 사진 업로드: 사용자가 모바일 또는 PC 환경에서 사진 촬영을 하거나 사진 파일을 업로드하여 음식을 인식
- 식판 위 음식 인식: AI-Hub의 음식 이미지 및 영양 정보 텍스트 파일을 이용하여 음식 분류 모델 학습
- 식판 위 음식의 양 추정: AI-Hub의 음식 이미지 및 영양 정보 텍스트 파일 을 이용하여 음식 양 추정 모델 학습
- 영양성분 정보 제공: 학습된 모델을 통해 인식된 음식의 칼로리와 주요 영양성분(탄수화물, 단백질, 지방 등) 정보를 제공
- 기록: 사용자가 업로드하여 입력한 식단 기록을 확인 가능

2.1.2 요구조건 수정사항

- 로그인 및 회원가입, 기록 기능은 후순위로 변경
- 음식의 양을 권장 섭취량 기준 25%, 50%, 75%, 100%, 125%의 5단계를 적음, 보통, 많음의 3단계로 단순화 하였다.

2.2 기존 제약사항과 수정사항

2.2.1 기존 제약사항 및 대책

- 네트워크 의존성: 인터넷 연결이 필요하므로 오프라인 환경에서는 제한적 기능 제공
  - 네트워크의 연결을 필수로 실행하도록 한다.

- DB 용량: 사용자 이미지와 데이터를 저장하기 위한 충분한 DB 용량 필요
  - 사진 1장을 전송하는 데이터는 크지 않으며, 압축된 JPEG 형태로 데이터를 전송할 경우 인터넷 서핑하는 정도의 트래픽이 발생한다. 또한, 사용자가 업로드한 이미지를 압축된 형태와 작은 크기로 리사이즈한다면 DB 용량에 대한 문제는 크지 않다.
- 서버 부하: 이미지 분석 작업이 서버 부하를 증가시킬 수 있으므로 성능 최적화 필요
  - YOLOv8 버전의 경우 RTX 4090 GPU를 사용하면 초당 100장이 넘는 이미지를 처리할 수 있다. 예측 이후의 작업은 CPU 기반 단순 사칙연산이므로 수만 명의 이용자가 생기지 않는 이상 서버 부하는 발생하지 않을 것으로 보인다. –? 필요한 부분인가?
- 학습 성능: 이미지 분류 학습 시 많은 시간이 필요하므로 고성능의 GPU가 필요
- 로컬에서는 모델만 이용, 학습은 고성능의 GPU가 있는 교내 리눅스 서버 사용

2.2.2 추가 제약사항 및 대책

Problem 1
음식은 만드는 사람에 따라 다양한 재료와 방법이 사용된다. 수많은 변수가 존재하는 문제를 일률적 데이터로 표준화하는 것이 적당한가?

Solution. 음식의 분류 모델 학습을 위해 2000장의 데이터가 사용되었고 이를 확인해본 결과 한 사람 또는 한 기관에서 만든 것이 아니라 여러 사람들에게서 데이터를 받아 처리한 것이므로 이는 기존의 AI-Hub의 음식 이미지 및 영양 정보 텍스트 파일을 이용하여 학습하면 된다.

Problem 2
양 추정의 경우 어떠한 것을 수치화 하여 linear model을 생성할 것인가?

Solution. AI-Hub의 음식 이미지 및 영양 정보 텍스트 파일의 영양 정보 DB의 섭취량을 보통으로 잡고 해당 음식을 인식할 때 바운딩 박스의 사이즈를 계산하여 이를 수치화 데이터로 이용한다.

3 설계 상세화 및 변경 내역

3.1 전체 설계 구조

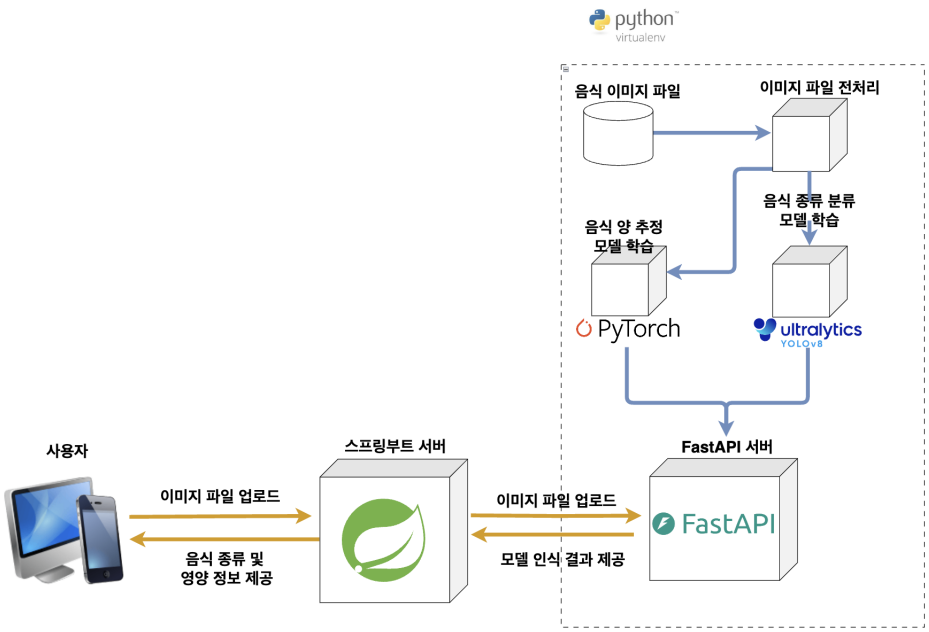

Figure 3.1: 전체 설계 구조 다이어그램

a) **모델 학습 구조**

먼저 AI-Hub에 올라와 있는 모든 음식 이미지들 중 한국 갤럽 조사에서 한국인이 선호하는 음식과 다빈도 섭취 음식류를 기반으로 하여 12가지의 음식을 선정하였다. 선정된 음식들읜 raw data의 이미지와 라벨 텍스트 파일을 이용하여 음식 종류 인식 모델 학습을 시킨다. 생성된 모델을 FastAPI 프레임워크 서버에 업로드한다.

3.2 인공지능 모델 학습

a) **모델 학습**

AI-Hub 데이터를 이용하여 음식 종류 인식 모델을 준수한 성능을 낼 수 있는것으로 확인하였다.

3.3 서비스 구조 및 API

3.3.1 서비스 구조

먼저 사용자가 음식 이미지를 촬영 또는 이미 저장된 음식 이미지를 업로드한다. Springboot 서버에 해당 이미지가 전송되면 웹에서는 세션을 통해 LocalStroage에 이미지가 저장되고 Springboot 서버에는 UUID를 이용하여 이미지 파일명을 변경시켜 자체적으로 저장한다. Springboot 서버에 저장된 이미지 파일은 다시 FastAPI 서버로 전송된다. FastAPI 서버에서 탑재된 모델을 이용하여 해당 이미지 파일의 음식 종류를 인식하고 해당 음식의 양을 추정하여 다시 Springboot 서버로 JSON 형태로 전달한다. 전달받은 Springboot 서버는 영양 정보가 들어 있는 CSV 파일을 읽어들여 인식된 음식의 종류의 영양 정보만을 다시 양에 맞게 계산하여 사용자에게 전달한다.

a) **프론트 엔드 구현 변경 사항**

- React를 사용하고자 하였으나, React의 숙련도가 높지 않고, 구현하고자 하는 웹 사이트가 상대적으로 단순하고 동적인 기능이 적기 때문에 HTML과 CSS만으로도 충분히 구현 가능할 것이라고 생각되어 HTML과 CSS를 사용하게 되었다.

3.3.2 API 구조

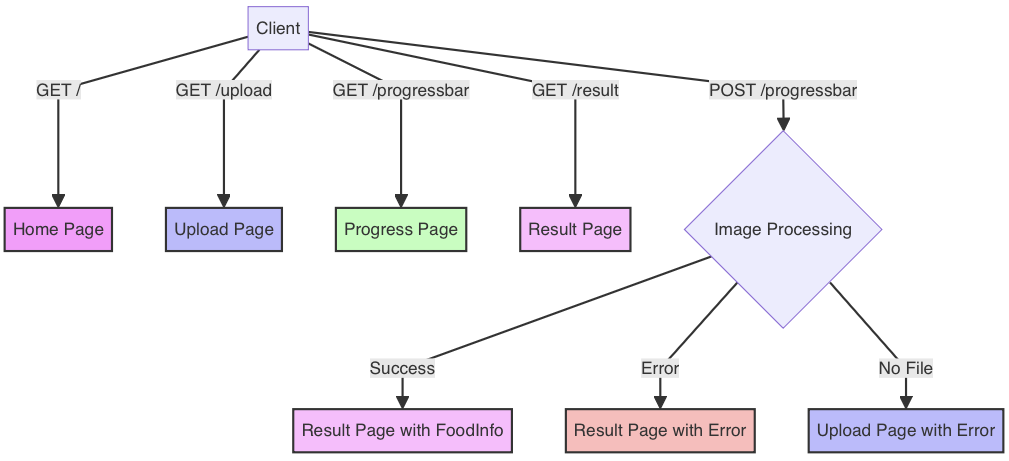

Figure 3.2: API 구조

서비스 구조에서 설명한 순서대로 사용자가 home page에 진입하여 이미지 선택 또는 음식 사진 업로드 버튼을 이용하여 식판 이미지 업로드 및 결정을 위한 upload에 진입할 수 있다. upload 페이지에서는 이미지가 업로드 되고 분석 시작 버튼을 이용하여 식판 이미지를 FastAPI

서버로 이미지를 전송하여 응답을 기다린다. 이 때 사용자는 progressbar로 이동하며 FastAPI 프레임워크 서버에서 응답을 다시 Springboot 서버로 전송하면 사용자는 result 페이지로 이동된다.

4 수정된 과제 추진 계획

4.1 서비스 구조 수정사항

음식 인식 및 음식의 양 추정 모델을 탑재하기 위한 서버를 Django 프레임워크를 사용하려 했으나 조원들과 조교님과의 토의 결과 Springboot 서버를 이용하여 사용자와 통신하고 사용자의 request는 다시 학습 모델을 탑재한 FastAPI를 보내는 구조를 이용하기로 변경하였다. 또한 모든 백엔드 부분 구현은 Python만을 이용하려고 하였으나 Springboot는 모든 조원들이 사용 경험이 있는 바 빠른 리눅스 서버 배포를 위해 Springboot와 FastAPI를 혼합하여 사용하였다.

4.2 양 추정 데이터셋

AI-Hub의 음식 이미지 및 영양 정보 텍스트 파일을 이용하여 양 추정 학습도 진행하려 하였으나 해당 이미지들의 경우 reference가 500원 동전이고 단계가 5단계로 나뉘어져 이미 라벨링이 되었기 때문에 식판을 reference로 하고 3단계로 양을 구분하려고 하는 기능과는 맞지 않는다고 판단하여 직접 식판과 저울을 구매하여 [1]3단계로 양을 나누어 식판에 담아 직접 데이터셋을 구성하는 것으로 변경하였다.

4.3 양 추정 linear model

양 추정 linear model은 각 단계마다 바운딩 박스의 크기를 수치화 하여 해당 크기별로 양을 추정할 수 있는 linear model을 생성한다.

[1]3단계는 '적음/보통/많음' 으로 구성되며 보통의 경우 AI-Hub의 음식 이미지 및 영양 정보 텍스트 파일의 DB에서 정한 섭취량을 기준으로 한다.

5 구성원별 진척도

- 이혁재
- 완료 사항
  a) 웹사이트 구현
- 문성재
  - 완료 사항
    a) FastAPI 서버 구축
    b) FastAPI와 Springboot 간 연결
  - 진행 사항
    a) 식판의 음식 양 추정 바운딩 박스를 통한 양추정 모델 학습
- 김상해
  - 완료 사항
    a) Springboot 서버 구축
    b) 식판 음식 인식 종류 선정
    c) Springboot와 FastAPI 간 통신 연결
    d) Yolov8을 이용한 음식 인식 모델 학습
  - 진행 사항
    a) Springboot에서 사용자에게 영양 정보 결과 데이터 표현 구현
    b) 음식 인식 리스트 최적화를 위한 섭취 선호도 조사
- 공통
  - 보고서 작성
  - 데이터 정리

6 과제 수행 내용 및 중간 결과

6.1 데이터 분석

AI-Hub에서 음식 이미지 및 영양 정보 텍스트 데이터셋의 분석을 하였다.

6.1.1 데이터 분석 결과

음식 종류 분류 train용 image의 경우 400종의 음식 당 2000장의 jpeg, jpg, png 파일로 구성되어 있었다. 각 이미지 파일은 최소 500만 화소 이상의 이미지들로 구성되어 있다. 또한 해당 음식 종류 분류 용 이미지에 각각 대응되는 txt파일과 xml파일도 라벨 데이터로 구성되어 있다. 또한 양 추정을 위한 정밀 촬영 음식 이미지의 경우 83종의 음식 당 500여장의 jpeg, jpg, png 파일로 구성되어 있다. 또한 각각의 음식 이미지 파일에 대응되는 txt, xml 형식의 라벨 데이터로 구성되어 있다.

6.2 Yolov8을 이용한 detection

음식 종류 탐지를 위한 모델은 Yolov8을 이용하였다. 해당 학습은 아래와 같이 구성 되었다.

- 주요 설정
  - 모델: yolov8s.pt
  - epoch: 20회
  - batch size: 16
  - 이미지 크기: 640×640
- train 관련 설정
  - 옵티마이저: auto
  - lr0: 0.01
  - 모멘텀: 0.937
  - 가중치 감소: 0.0005
  - warm up epoch: 3.0
- 데이터 증강
  - HSV 색 공간 변형
  - 무작위 좌우 반전 (50% 확률)

- 모자이크 증강
- 랜덤 지우기 (40% 확률)

6.2.1 Confusion Matrix Normalized

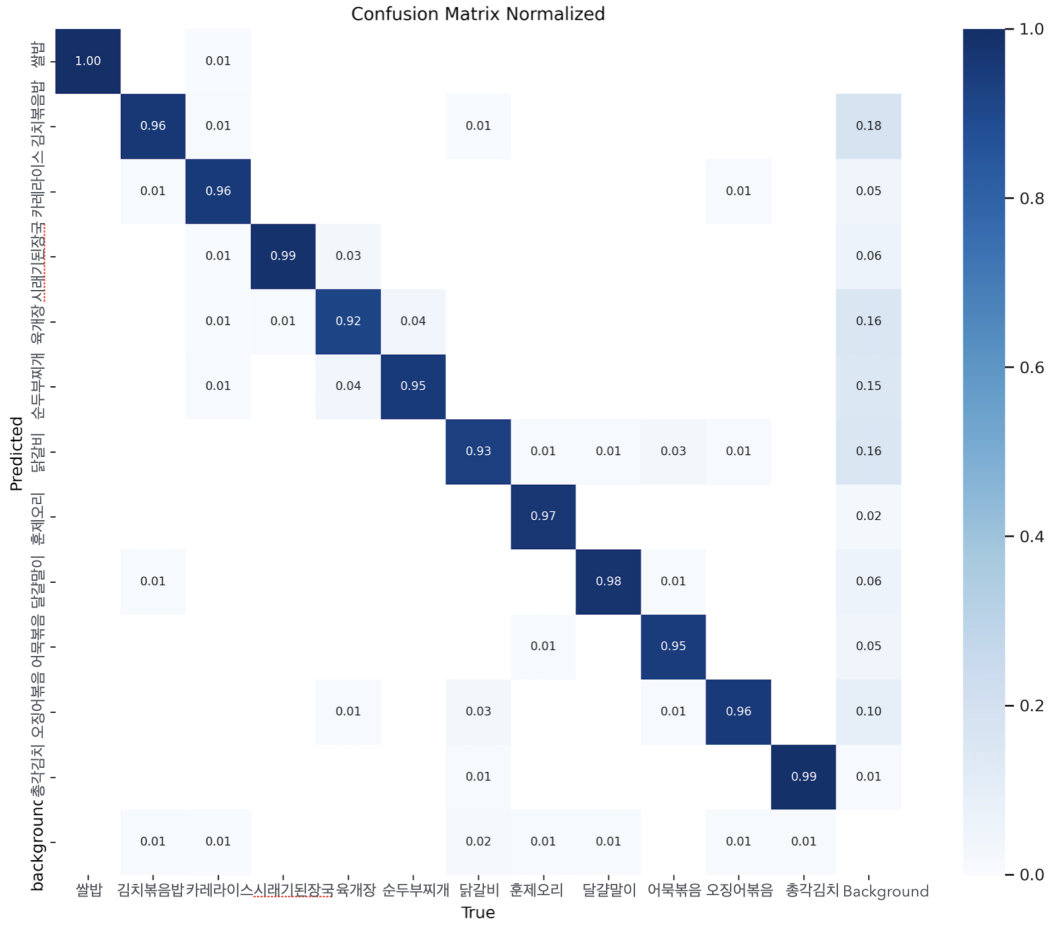

Figure 6.1: confusion matrix normalized

대부분의 값이 0.92에서 1.00 사이로 높은 정확도를 나타내고 있으며 가장 잘 분류되는 식품은 쌀밥이며 떡갈비가 0.92로 상대적으로 낮은 성능을 보이고 있다.