

객체 인식을 활용한 식재료 인식 및 레시피 추천 프로그램

환끼메이커

Meal Maker



Contents

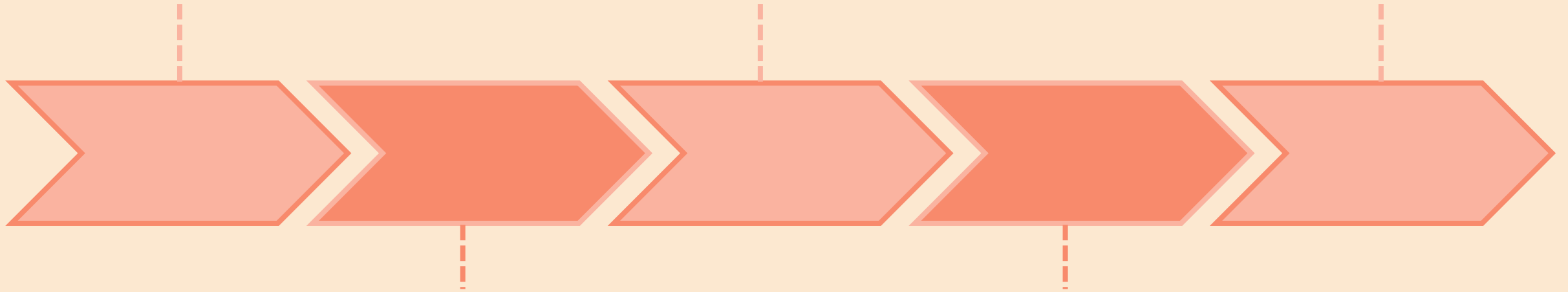
01 프로젝트 개요

03 개발 공정

05 결과 및 마무리

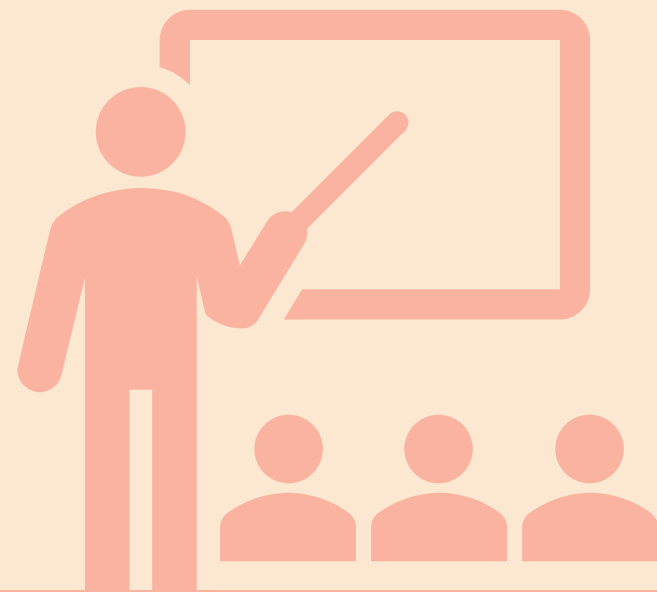
02 데이터 수집

04 모델 학습 및 평가



이1

프로젝트 개요



01 프로젝트 개요

한끼메이커란?

객체 인식 기술을 활용하여 식재료를 인식하고,
그에 맞는 요리와 레시피를 추천해주는 프로그램

식사 메뉴
고민 해결



최적의 요리와
레시피 추천



새로운 요리 도전
기회 제공



식재료 낭비
문제 해결



02

데이터 수집



02 데이터 수집

Image Data

구글 네이버 (2,451장)
직접 촬영 (74장)

Recipe Data

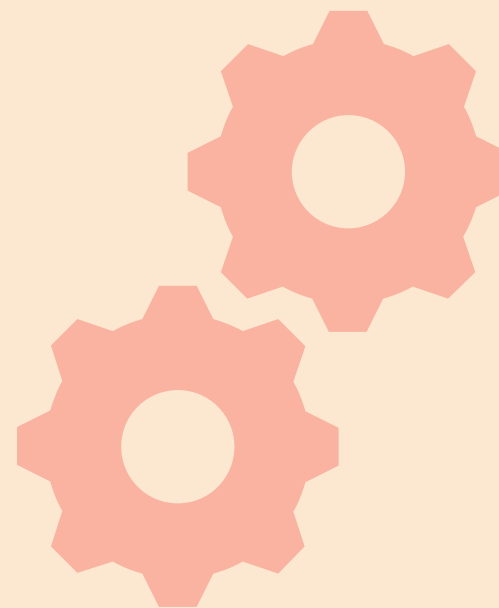
KADX 농식품 빅데이터 거래소
2007년부터 2023년까지 '만개의 레시피'
서비스를 통해 수집된 레시피 데이터

****44가지 식재료****

가지 감자 고구마 고추 김치 깻잎 달걀 닭고기 당근 대파 두부 돼지고기 떡 라면 마늘 메추리알
무 미나리 배 배추 베이컨 부추 사과 새송이버섯 새우 소고기 숙주 스팸 식빵애호박 양배추
양상추 양파 오리고기 오이 오징어 우유 참치캔 치즈 콩나물 파스타 파프리카 팽이버섯 표고버섯

03


개발 공정



03 개발 공정

① 레시피 데이터 전처리

초기 18개 컬럼에서 ['요리명', '조회수', '조리방법', '재료', '인분수', '난이도', '조리시간']을
제외하고 나머지 컬럼 삭제, 인분수 == 1인분만 남겨놓고 제외



'재료' 컬럼에서 인식 가능한 재료만 추출하여 '체크할 재료' 컬럼 생성



요리명에 ['는법', '손질', '보관', '깍기', '자르기']가 있으면 제외

03 개발 공정

② 이미지 데이터



① 이미지 업로드

총 2,525장

2,166장 → 2,525장
(정확도 향상을 위해 훈련 데이터 추가)



② 라벨링

44가지 식재료
(채소, 과일, 유제품 등등)



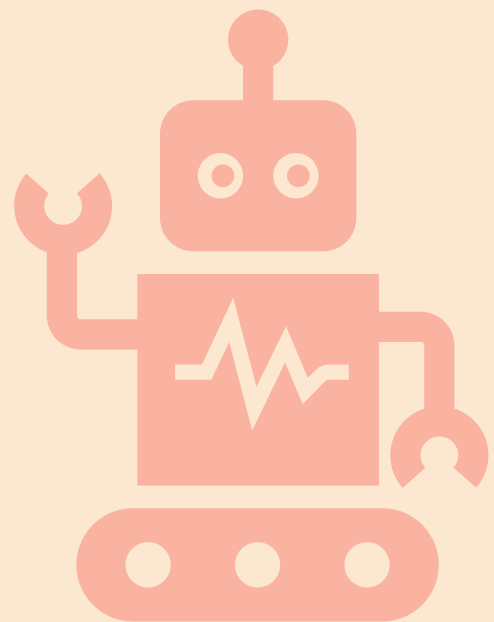
③ 데이터 증강

반전, 회전, 자르기, 비틀기

(X3) 총 6,613장
* 증강시키면서 무의미한 이미지 삭제

04

모델 학습 및 평가



04 모델 학습 및 평가

① 라이브러리 설치

```
!pip install ultralytics==8.0.196  
!pip install roboflow
```

② 라이브러리 및 모듈 임포트

```
from ultralytics import YOLO  
from roboflow import Roboflow  
import os  
from IPython.display import display, Image  
import glob
```

04 모델 학습 및 평가

③ Roboflow API로 데이터셋 다운로드

```
rf = Roboflow(api_key="zDz3bkG3qq1BvgKqELTB")  
project = rf.workspace("project4mealmakerbox-wiokd").project("foodproject-8rpvr")  
version = project.version(7)  
dataset = version.download("yolov8")
```

④ 모델 훈련

```
!yolo task=detect mode=train model=yolov8n.pt data={dataset.location}/data.yaml  
epochs=100, imgsz=640 save=True
```

- train = 학습모드, YOLOv8n의 사전 학습된 가중치 사용, 훈련 100번 이미지 사이즈 640, 훈련 저장

04 모델 학습 및 평가

⑤ 모델 검증

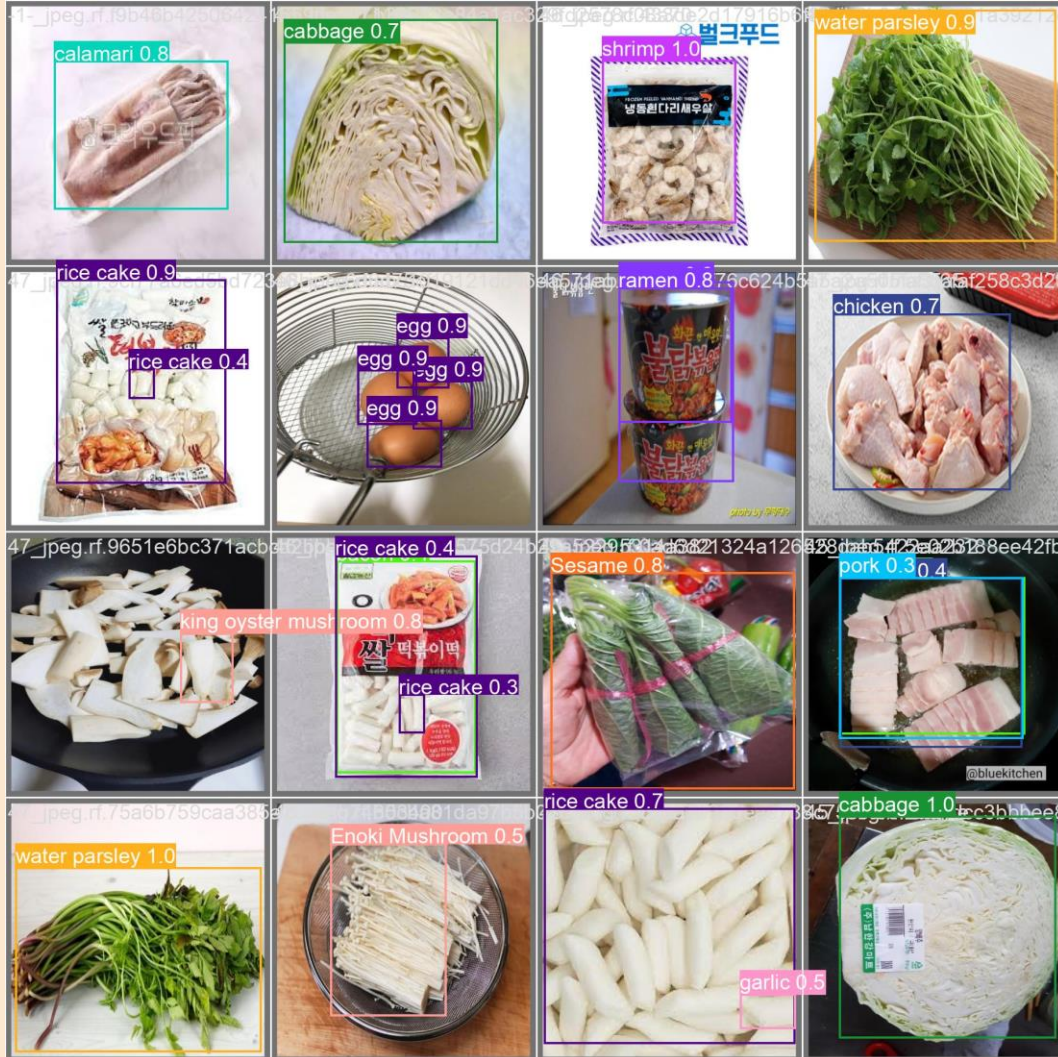
```
!yolo task=detect mode=val model=/content/runs/detect/train/weights/best.pt  
data={dataset.location}/data.yaml
```

- val = 검증모드, 학습된 best.pt를 이용하여 검증

```
display(Image(filename=f'/content/runs/detect/val/val_batch0_pred.jpg', width=600))
```

- 검증한 결과 이미지 보여주기

04 모델 학습 및 평가



개별 식재료 이미지는
높은 확률로 인식 가능

04 모델 학습 및 평가

⑥ 모델 테스트

```
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt  
conf=0.5 source=/content/test_img
```

- 학습된 best.pt로 테스트 실시

05

결과 및 마무리



05 결과 및 마무리

- 소고기 vs 돼지고기



05 결과 및 마무리

- 숙주 vs 콩나물

mung bean sprout 0.72



Bean sprouts 0.96



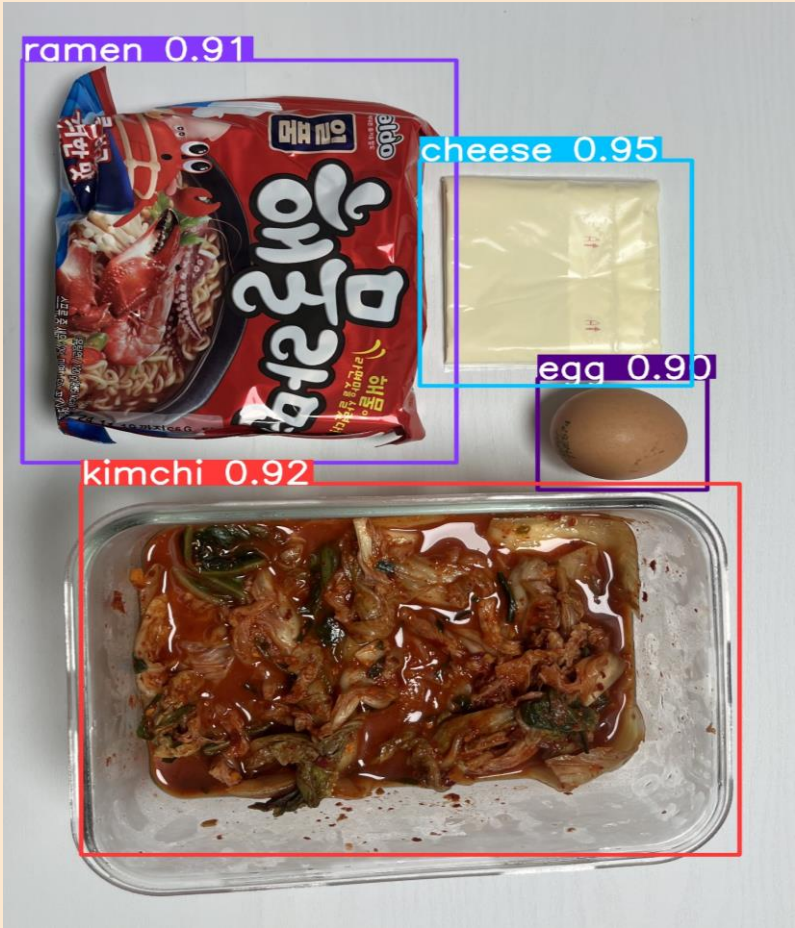
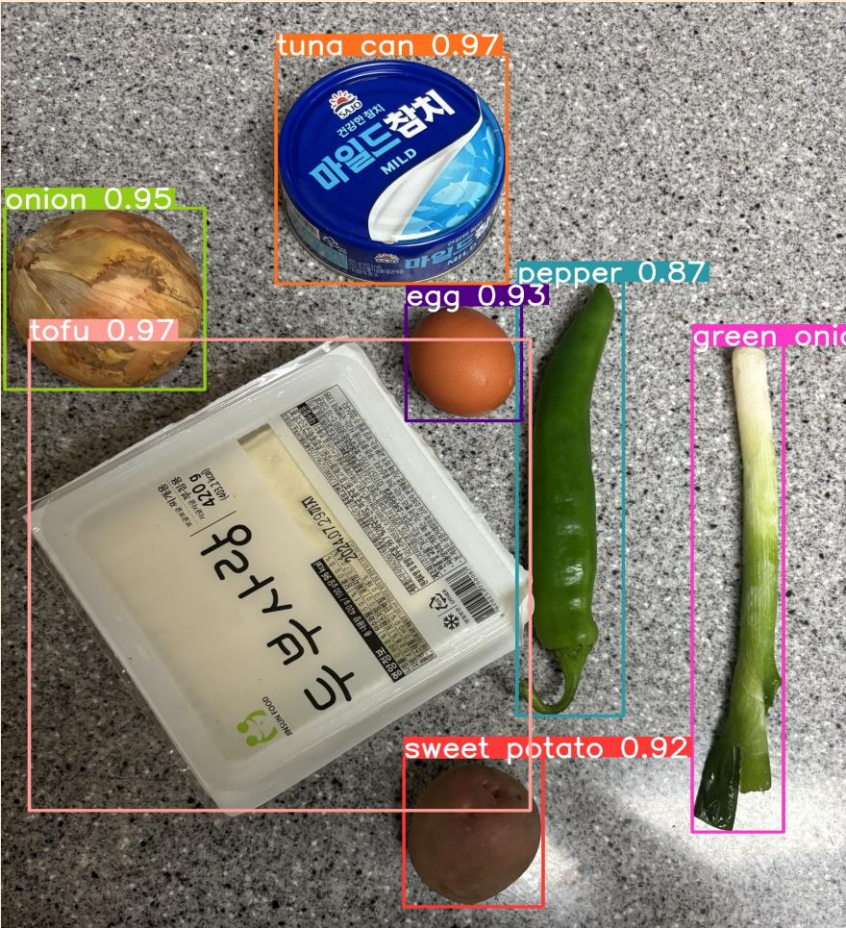
Bean sprouts 0.96



mung bean sprout 0.86



05 결과 및 마무리

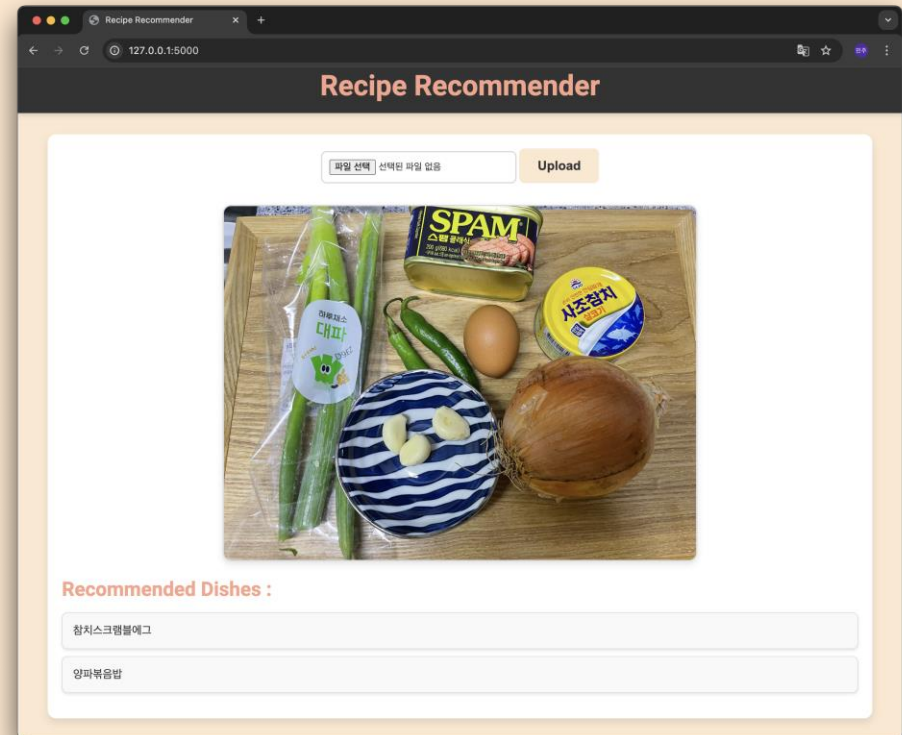
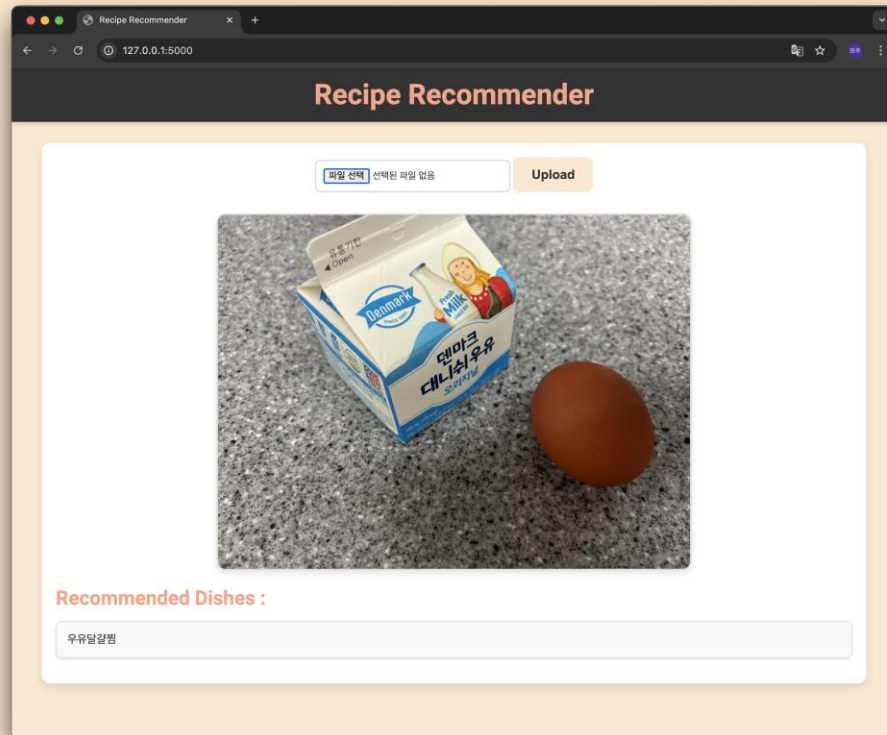


05 결과 및 마무리



05 결과 및 마무리

- Flask로 구현한 '요리 추천' 웹 페이지



Recipe Recommender

파일 선택

선택된 파일 없음

Upload

05 결과 및 마무리

1

Train image 6132장
Valid image 481장
Test image 7장

2

T4 gpu / 100 에포크 / 1시간 40분

3

Precision(정밀도) : 0.738
Recall(재현율) : 0.727

05 결과 및 마무리

- 아쉬운 점

웹 구현의
한계

레시피 데이터
미흡

식재료 다양성 부족

감사합니다

