

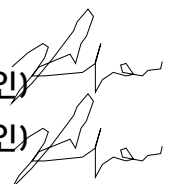
종합설계 프로젝트 수행 보고서

프로젝트명	“딥러닝과 AR을 이용한 식재료 인식 및 응용 애플리케이션 개발”
팀번호	S1-10
문서제목	수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 4차발표 중간보고서() 최종결과보고서(○)

2020.11.27.

S1-10 팀원 : 박병남 (팀장)
유현승 (팀원)
이동준 (팀원)

지도교수 : 최종필 교수 (인)
지도교수 : 최종필 교수 (인)



문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2019.12.15.	이동준	1.0	계획서 및 1차 발표자료 작성	최초 작성
2020.12.16.	박병남	1.1	계획서 내용 추가	개발 환경 추가
2020.12.16.	박병남	1.2	계획서 내용 변경	개발 환경 변경
2020.01.21.	유현승	2.0	수행 보고서	양식에 맞게 기존 보고서를 토대로 작성
2020.01.22.	박병남	2.1	보고서 내용 추가	작품선정 추가 작성 및 문서 틀 개선
2020.01.22.	이동준	2.2	보고서 내용 변경	최근 동향 보완
2020.01.23.	박병남	2.3	보고서 내용 변경	개발 목표 보완
2020.01.28.	이동준	2.4	최종 검토 후 제출	최종 검토 후 제출
2020.02.26.	박병남	3.0	2차 수행 보고서 작성	추가 작성
2020.02.28.	이동준	3.1	2차 수행 보고서 작성	추가 작성
2020.03.02.	유현승, 이동준, 박병남	3.2 , 3.3	2차 수행 보고서 작성	추가 작성 및 최종 검토
2020.04.30.	박병남, 유현승, 이동준	4.0	3차 수행 보고서 작성	프로토타입 작성
2020.06.27.	박병남, 유현승, 이동준	5.0	4차 수행 보고서 작성	데모, 코드 작성
2020.06.27.	박병남, 유현승	5.1	4차 수행	수정 및 검토
2020.11.27.	박병남, 유현승, 이동준	6.0	최종 수행 보고서 작성	최종 보고서 작성

이 문서는 한국산업기술대학교 컴퓨터공학부의 “종합설계”교과목에서 프로젝트 “딥러닝과 AR을 이용한 식재료 인식 및 응용 애플리케이션 개발”을 수행하는 (S1-10, 박병남,유현승,이동준)들이 작성한 것으로 사용하기 위해서는 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경	4
2. 기존 시장/기술동향 분석	5
3. 개발 목표	6
4. 팀 역할 분담	6
5. 개발 일정	7
6. 개발 환경	7

II. 본론

1. 개발 내용	8
2. 문제 및 해결방안	8
3. 시험시나리오 & 시스템 구성도	9
4. 상세 설계	10
5. Prototype 구현	15
6. 시험/ 테스트 결과	18
7. Coding & DEMO	20

III. 결론

1. 연구 결과	29
2. 작품제작 소요재료 목록	30

참고문헌	31
------------	----

I. 서론

1. 작품선정 배경

- 모바일 세대의 요리 콘텐츠에 대한 관심 증가

- 1인 가구 비율이 2000년과 비교해서 남자는 195.4% 증가, 여자는 120.9% 증가
- 남성 1인 가구 비율이 가장 높은 연령은 30세(22.5%)
- 여성 1인 가구 비율이 두 번째로 높은 연령은 27세(16.0%)

(천가구)

연도	1인가구 (천가구)	일반가구 대비 1인 가구 비율 (%)
2000년	2,224	15.5
2005년	3,171	20.0
2010년	4,142	23.9
2015년	5,203	27.2
2016년	5,398	27.9
2017년	5,619	28.6

— 1인가구 — 일반가구 대비 1인 가구 비율

- 요리에 대한 버즈량 증가
 - * 버즈량 : 온라인에서의 언급된 횟수
- 매일 같은 배달음식에 대한 갈증해소
- 1인 가구의 관심사는 일상의 한끼 해결
- '요섹남', '요섹녀'라는 신조어 등장

<요리에 대한 버즈량 비교>

943,028 → 1,443,591 (↑ 53.1%) → 1,972,685 (↑ 36.7%)

기간: '14.01.01~'14.06.03, '14.07.01~'14.12.31, '15.01.01~'15.06.03

(출처: SK플래닛 소셜분석 시스템(BNS) 3.0, beta, 분석 기간: '14.01.01~'15.06.30)

- 쿡방 예능의 급상승(동시간대 시청률 1위 섭렵)

- Food-Tech의 투자 가속화 추세

<p>북미</p> <ul style="list-style-type: none"> ■ 데이터 분석 기반, 레시피 검색 (9.2M\$ 투자) ■ '맛' 데이터를 분석, 1,500만 UV 확보. ■ Native Ad 및 API과금으로 100억 연매출 ■ eBay 출신 창업자, 제휴쇼핑BM 확장 	
<p>일본</p> <ul style="list-style-type: none"> ■ 인구의 절반, 여성의 80%가 사용하는 음식계의 구글 ■ 레시피 전문 사이트, 4,400만 UV ■ 연매출 650억, 영업이익 50% ■ 유료회원비, Native Ad, 신선식품 제휴 쇼핑 ■ 2009년 기업 공개(시가총액 1조원+) 	

- 변동하는 시장물가를 매번 파악하기 힘들
- 식재료의 칼로리나 영양성분의 파악이 번거로움

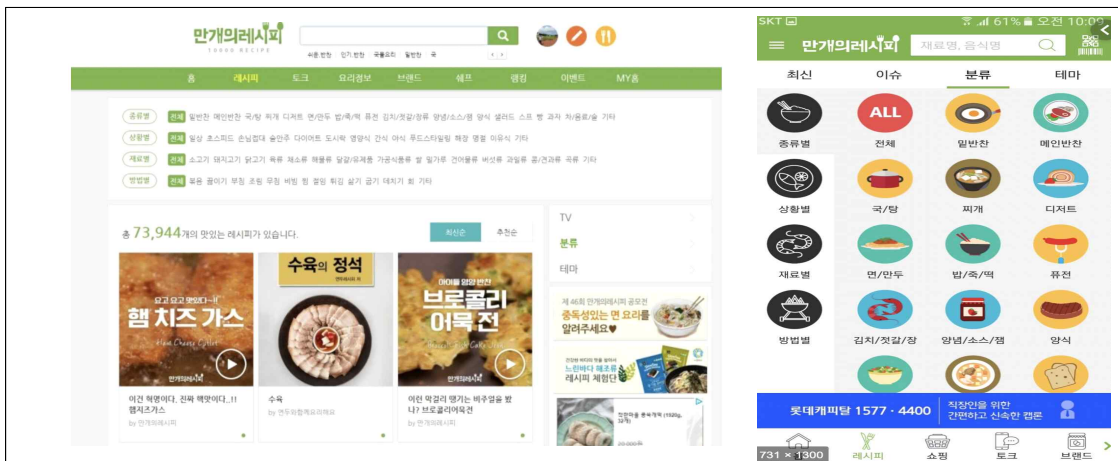
2. 기존 시장/기술동향 분석

가. 주방친구



- 나에게 필요한 정보와 레시피를 저장해서 보기 쉽게 정리하는 기능
- 카카오톡, 밴드, 페이스북 등 SNS 공유 기능
- 찾고 싶은 정보와 레시피를 검색기능으로 원하는 것을 찾을 수 있음
- 주방 상식 등의 정보들을 쉽게 접근할 수 있음

나. 요리백과 만개의 레시피



- 종류별 레시피로 각 종류별로 원하는 레시피를 확인할 수 있음
- 상황 별 레시피로 일상, 손님접대 등 상황에 맞는 레시피를 확인할 수 있음
- 방법 별 레시피로 끓이기, 볶음 등 방법에 따라 레시피를 확인할 수 있음
- 재료 별 레시피로 가진 재료에 따라 레시피를 확인할 수 있음
- 자기가 좋아하는 레시피를 스크랩하여 편집 또는 공유를 할 수 있음

다. 해먹남녀



- 많은 요리 레시피, 꿀 팁, 레시피 들에 대해 리뷰와 함께 확인 할 수 있음
- 자취생, 상황, 국가별 요리등의 요리를 확인 할 수 있음
- 레시피의 영양성분 및 주의 성분을 한 눈에 확인할 수 있음
- 내가 좋아하는 음식의 레시피를 스크랩 할 수 있음

3. 개발 목표

- 식재료를 인식하여 급변하는 시장물가와 영양성분을 AR로 시각화하여 편리함과 유용함을 제공
- 선호도 검사와 검색 내용을 기반으로 유사도를 측정하여 '맞춤 음식 레시피' 추천
- 커뮤니티 생성과 레시피를 공유를 통해 조리문화의 질을 높임
- 인식된 식재료의 레시피를 보고 다양한 요리를 혼자서도 할 수 있게 함

4. 팀 역할 분담

번호	성명	주요 역할
1	박병남	사용자 커뮤니티 및 애플리케이션 UI 구현, 답러닝 구현
2	유현승	식재료 이미지 추출 및 답러닝 구현, MySQL DB설계 및 서버 연동
3	이동준	AR 이미지 매칭 및 카메라 화면 구현 및 답러닝 구현

5. 개발 일정

추진사항	12월	1월	2월	3월	4월	5월	6월	7월
사전조사 및 제안서발표								
자료수집 및 스토리								
시스템 구현								
프로토타입 완성								
통합 테스트 및 보완								
최종 보고서 발표								

6. 개발 환경

- 운영체제 : Windows 10
- 개발언어 : Python 3.7, C#
- 사용 프레임워크 : Unity, Android 6.0 (Marshmallow) API 23
- 주요 라이브러리 : Keras (2.3.0), Tensorflow (1.15.0), Unity ARcore, Vuforia, OpenCV
- 서버 프로세스 : php-7.4.0.
- DB : MySQL 8.0, Vuforia Target Manager

Ⅱ. 본 론

1. 개발 내용

- 식재료를 AR카메라로 비춰 딥러닝된 모델과 비교하여 일치할 경우 이름, 시장물가, 칼로리를 화면 위에 나타냄
- 인식된 식재료를 터치하면 해당 식재료로 만들 수 있는 요리와 레시피를 보여줌
- 식재료들의 영양성분을 나타내주고 등록된 레시피의 개인 만족도(평점)를 제공함
- 선호도 조사와 검색한 음식 간의 유사도를 측정하여 유사도가 짝은 음식 레시피 추천

2. 문제 및 해결방안

가. 집에서 요리하는 사람들을 위한 레시피 제공

→ DB에 저장된 레시피 조회 및 추가

나. 식재료 모델의 데이터 처리

→ DB 활용

다. 식재료를 카메라로 인식하여 얻을 수 있는 정보를 사용자에게 제공

→ 시장물가 api를 활용하여 효율적인 대처를 할 수 있게 함

→ 추가기능 구현 시 AR카메라 기능을 통해 칼로리나 영양성분을 제공

라. 절단면에 따라 인식이 불가능한 식재료

→ 인식이 불가능하게 된 특수한 경우 사용자 입력 가능하게 함

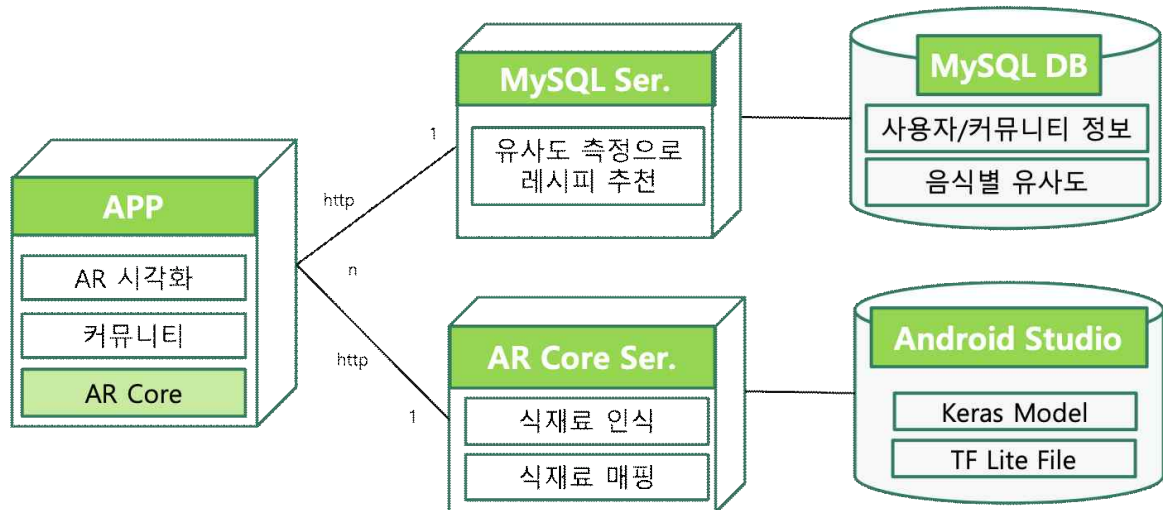
마. 이미지 검색에 대한 어려움

→ 웹 크롤링으로 해결

3. 시험 시나리오 & 시스템 구성도



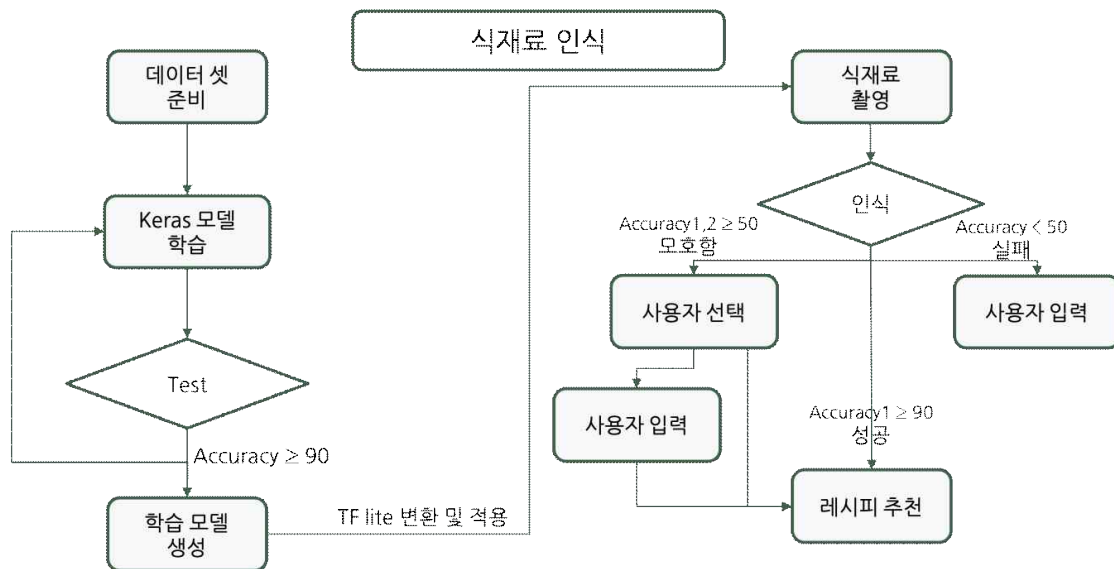
- 가. 모바일 어플리케이션 실행
- 나. 사용자 정보 입력 (로그인) : 첫 등록 후 자동 로그인
- 다. AR 카메라 구동
- 라. 식재료 인식
- 마. 인식된 식재료의 시장물가, 영양정보 확인
- 바. 식재료 터치 후 가능 레시피 조회 : 해당 식재료로 만들 수 있는 레시피를 조회



app	Android Studio	MySQL
<ul style="list-style-type: none"> - 음식 정보를 AR 시각화 구현 - 커뮤니티 UI와 창구 마련 	<ul style="list-style-type: none"> - AR core Server를 활용하여 식재료 인식과 매핑 - 식재료 학습 모델 저장 	<ul style="list-style-type: none"> - 사용자 정보 저장 - 유사도 측정 레시피 적용

4. 상세 설계

1. 딥러닝 모델



- Keras Model (식재료 100가지 : 훈련셋 각 1000장, 평가셋 각 30장 / 총 103000장)

• [food_dataset.py](#) (데이터 로딩 + npy파일 저장)

```

In [ ]: # -*- coding : utf-8 -*-

from sklearn import model_selection
from sklearn.model_selection import train_test_split
from PIL import Image
import os,glob
import numpy as np

# 분류 데이터 로딩
root_dir="C:/Users/LG/Desktop/deep/ingredient/"

#카테고리 유형 지정
categories=["onion","garlic","shrimp","carrot","potato"]
nb_classes=len(categories)

#이미지 크기 지정
image_width=64
image_height=64

#데이터 변수
X=[]
Y=[]
for idx, category in enumerate(categories):
    image_dir=root_dir + category
    files=glob.glob(image_dir+"/*.*.jpg")
    print(image_dir+"/*.*.jpg")

    for i,f in enumerate(files):
        #이미지 로딩
        img=Image.open(f)
        img=img.convert("RGB")
        img=img.resize((image_width, image_height))
        data=np.asarray(img)
        X.append(data)
        Y.append(idx)

X=np.array(X)
Y=np.array(Y)

#훈련 데이터와 테스트 데이터 나누기
X_train, X_test, Y_train, Y_test=train_test_split(X,Y)

xy=(X_train, X_test, Y_train, Y_test)

#데이터 파일 저장
np.save(root_dir+"kingredient.npy",xy)
  
```

• food_classify.py (모델 구성 + 딥러닝 실시)

```
In [ ]: # coding : utf-8

# 사용할 모델 라이브러리 import
import sys, os
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Activation
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers import Dense
from keras.utils import np_utils
import numpy as np

# 초기 설정
root_dir="C:/Users/LG/Desktop/deep/ingredient/"
categories=["onion", "garlic", "shrimp", "carrot", "potato"]
nb_classes=len(categories)
Image_size=64

# 데이터 로딩
def load_dataset():
    x_train, x_test, y_train, y_test=np.load("C:/Users/LG/Desktop/deep/ingredient/ingredient.npy")
    x_train=x_train.astype("float") / 256
    x_test=x_test.astype("float") / 256
    y_train=np_utils.to_categorical(y_train, nb_classes)
    y_test=np_utils.to_categorical(y_test, nb_classes)
    return x_train, x_test, y_train, y_test

# 모델 구성
def build_model(in_shape):
    model=Sequential()
    model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Convolution2D(64, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(64, 3, 3))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Flatten())
    model.add(Dense(512))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='binary_crossentropy', optimizer='rmsprop',
                  metrics=['accuracy'])

    return model

# 모델 학습을 수행하고 저장된 모델을 파일로 저장
def model_train(x, y):
    model=build_model(x.shape[1:])
    model.fit(x, y, batch_size=32, epochs=30)

    return model

# 모델 평가하기
def model_eval(model, x, y):
    score=model.evaluate(x, y)
    print('loss=', score[0])
    print('accuracy=', score[1])

# 모델 학습 및 평가
x_train, x_test, y_train, y_test=load_dataset()
model=model_train(x_train, y_train)
model_eval(model, x_test, y_test)

# 모델 저장
model.save("C:/Users/LG/Desktop/deep/ingredient/ingredient_model.h5")
```

· food_classify_run.py (모델 파일 로딩 + 입력 이미지 예측결과)

```
In [ ]: # coding : utf-8
import sys,os
from keras.models import Sequential
from keras.layers import Convolution2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.utils import np_utils
from keras.models import load_model
from PIL import Image
import numpy as np

# 테스트 이미지 목록
image_files=[ "C:/Users/LG/Desktop/deep/ingredient/testdata/onion1.jpg",
              "C:/Users/LG/Desktop/deep/ingredient/testdata/garlic1.jpg",
              "C:/Users/LG/Desktop/deep/ingredient/testdata/carrot1.jpg",
              "C:/Users/LG/Desktop/deep/ingredient/testdata/carrot2.jpg",
              "C:/Users/LG/Desktop/deep/ingredient/testdata/potato1.jpg",
              "C:/Users/LG/Desktop/deep/ingredient/testdata/potato2.jpg",
              "C:/Users/LG/Desktop/deep/ingredient/testdata/shrimp1.jpg",
              "C:/Users/LG/Desktop/deep/ingredient/testdata/shrimp2.jpg"
            ]
image_size=64
nb_classes=len(image_files)
categories=["onion", "garlic", "shrimp", "carrot", "potato"]

X=[]
files=[]
#이미지 불러오기
for fname in image_files:
    img=Image.open(fname)
    img=img.convert("RGB")
    img=img.resize((image_size, image_size))
    in_data=np.asarray(img)
    in_data=in_data.astype("float")/256
    X.append(in_data)
    files.append(fname)

X=np.array(X)

#모델 파일 읽어오기
model=load_model('C:/Users/LG/Desktop/deep/ingredient/kingredient_model.h5')

#예측 실행
pre=model.predict(X)

#예측 결과 출력
for i,p in enumerate(pre):
    y=p.argmax()
    print("입력:", files[i])
    print("예측:", "["+y+"]", categories[y], "/ Socre", p[y])

#이미지 불러오기
for fname in image_files:
    img=Image.open(fname)
    img=img.convert("RGB")
    img=img.resize((image_size, image_size))
    in_data=np.asarray(img)
    in_data=in_data.astype("float")/256
    X.append(in_data)
    files.append(fname)

X=np.array(X)

#모델 파일 읽어오기
model=load_model('C:/Users/LG/Desktop/deep/ingredient/kingredient_model.h5')

#예측 실행
pre=model.predict(X)

#예측 결과 출력
for i,p in enumerate(pre):
    y=p.argmax()
    print("입력:", files[i])
    print("예측:", "["+y+"]", categories[y], "/ Socre", p[y])
```

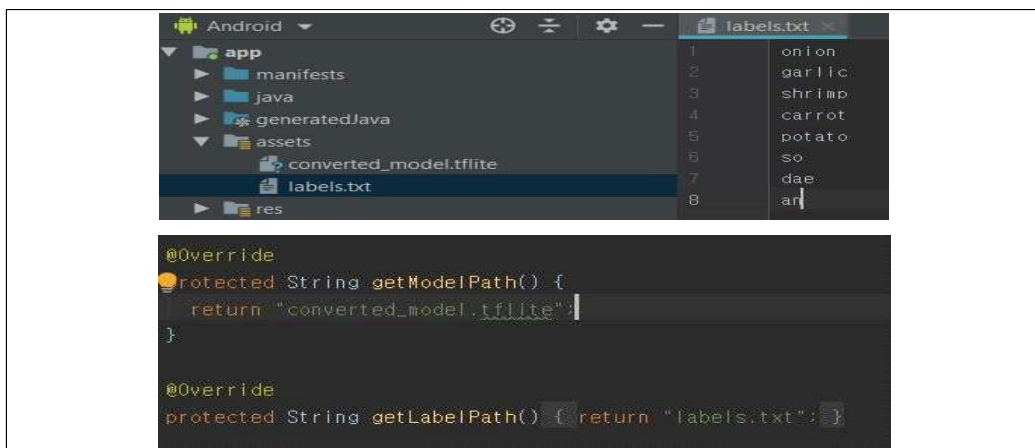
- TF Lite File 변환 (안드로이드에 Keras 모델을 적용시키기 위함)

```
import tensorflow as tf

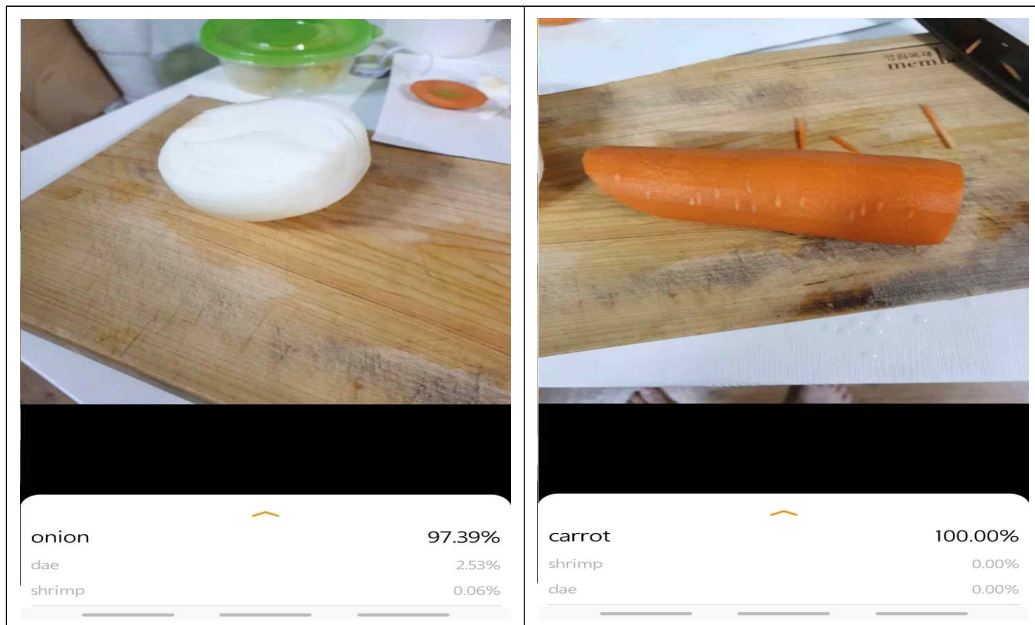
converter = tf.lite.TFLiteConverter.from_keras_model_file("keras_model.h5")
tflite_model = converter.convert()
open("converted_model.tflite", "wb").write(tflite_model)
```

2. 앱 (안드로이드)

- Android Studio 프로젝트의 asset에 tflite, labels 및 코드 적용



- 적용한 프로젝트를 실제 휴대폰으로 APK를 통해 테스트(양파와 당근)



- AR Core

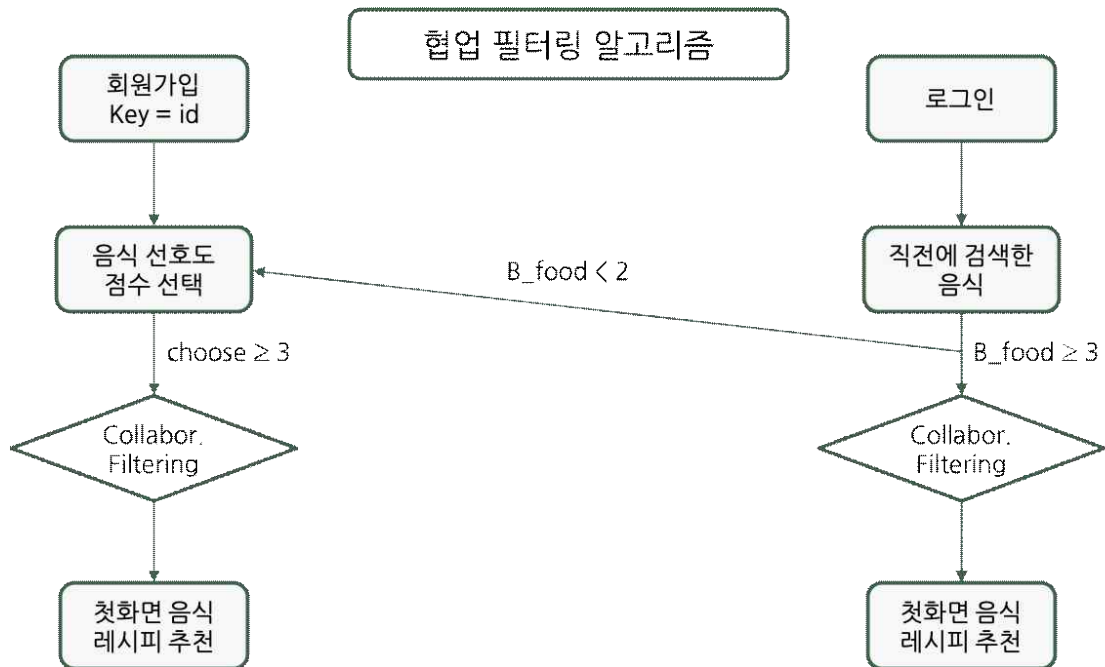
3. DB

- My SQL

• 협업 필터링 알고리즘

- 가. User-base Collaborative Filtering으로 n차원일 때 편집거리 측정
- 나. 유클리디안 사용
- 다. 유저별 편집거리를 정규화
- 라. 정규화 값이 가장 높은 유저 음식을 추천

• 협업 필터링 구성도



• 협업 필터링 예시 코드1

- 12번 라인 코드는 정규화를 보기위해 2차원 공간에서만 예시 출력

```
1 critics = {
2     'JPC':{'bulgogi':5,'kimchi':3,'pasta':1, 'pizza':2},
3     'JJL':{'bulgogi':2,'kimchi':0,'pasta':5, 'pizza':4},
4     'JGC':{'bulgogi':4,'kimchi':5,'pasta':3, 'pizza':1},
5     'KPU':{'kimchi':1,'pasta':4}
6 }
7
8 from math import sqrt
9 def sim(i, j):
10     return sqrt(pow(i,2)+pow(j,2))
11
12 for i in critics:
13     if i!='KPU':
14         num1 = critics.get('KPU').get('kimchi')- critics.get(i).get('kimchi')
15         num2 = critics.get('KPU').get('paste')- critics.get(i).get('paste')
16         print(i," : ", 1/(1+sim(num1,num2))) #정규화
```

· 협업 필터링 예시 코드2

- 유클리디안을 활용하여 다차원에서의 두 사람간 거리 구하기

```

19 def sim_distance(data, name1, name2):
20     sum=0
21     for i in data[name1]:
22         if i in data[name2]: #같은 음식을 평가했다면
23             sum+=pow(data[name1][i]- data[name2][i],2)
24
25     return 1/(1+sqrt(sum))
26
27
28 def top_match(data, name, index=3, sim_function=sim_distance):
29     li=[]
30     for i in data:
31         if name!=i: #자기 자신은 제외
32             li.append((sim_function(data,name,i),i)) # 유사도, 이름을 튜플에 묶어 리스트에 추가한다
33     li.sort() #오름차순 정렬
34     li.reverse() #내림차순 정렬
35
36     return li[:index]
37
38 top_match(critics, 'KPU')

```

- 위 코드 결과 예시 중 'JL'가 가장 유사도가 높음.

```

[(0.4142135623730951, 'JL'),
 (0.21712927295533244, 'JPC'),
 (0.1951941016011038, 'JGC')]

```

5. prototype 구현

1. 시작 화면	2. 로그인 화면
 <p>© 2019. HonCook All Rights Reserved</p>	 <p>Username : _____</p> <p>Password : _____</p> <p>Create an account</p> <p>© 2019. HonCook All Rights Reserved</p>

3. 메인 화면



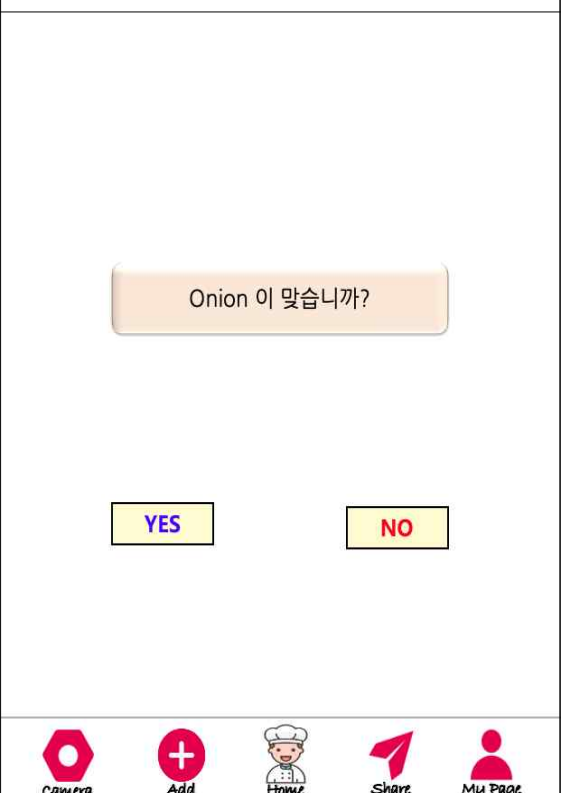
4. Camera 기능














4-a. Accuracy $\geq 90\%$



4-b. $50\% \leq$ Accuracy $< 90\%$



4-c. Accuracy < 50%	5. 메인 화면 한식 클릭 시
<p>인식에 실패했습니다. 각도를 다르게 조정해서 재촬영 혹은 직접 입력 해주세요.</p> <p>재촬영 하기 직접 입력</p>	<div> <div>  <div> <div>한 식</div> <div>중 식</div> <div>일 식</div> <div>양 식</div> </div> </div> <div> <div>레 시 피 1</div> <div>사 진</div> <div>조 리 과 정</div> </div> </div>
    	    

1. 시작 화면.

애플리케이션 시작 시 실행

2. 로그인 화면

- 회원가입 및 로그인 페이지

3. 메인 화면

- 한식 일식 중식 양식별 레시피 가이드
- 알고 싶은 레시피 검색창
- 사전 선호도 검사를 통한 사용자별 음식 추천하여 레시피 배너
- 하단 메뉴에서 식재료 인식, 레시피 쓰기, 메인 화면 이동, 친구 공유, 마이페이지 기능

4. Camera (하단 카메라 클릭 시)

- Camera 기능 활성화. 카메라에 비춰진 식재료 인식하여 하단에 정확도가 표기됨.
- 식재료 정확도가 90%가 넘을 시 성공하여 **4-a** 로 이동
- 식재료 정확도가 50%이상 90% 미만 일시 **4-b** 로 이동
- 식재료 정확도가 50%미만 일시 **4-c** 로 이동

5. 메인화면 한식(일식 중식 양식) 레시피 가이드 선택 시

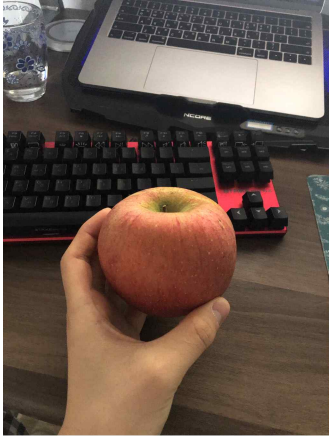
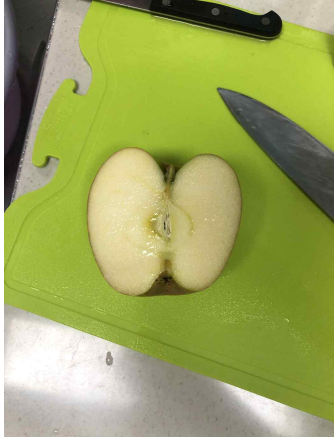
- 왼쪽 상단에서 각 분야 선택 가능
- 현재 검색, 찜하기(저장) 등 구현 예정.

6. 시험/ 테스트 결과

1. 식자재 비교알고리즘 시험/테스트

시험 테스트는 사과, 바나나, 굴비, 청경채 4가지로 진행

가. 사과

	
apple1.jpeg	apple5.jpeg

입력: ./drive/My Drive/ByeongNam_test/banana_test/apple1.jpeg

예측: [3] apple / Score 0.9112341

입력: ./drive/My Drive/ByeongNam_test/banana_test/apple2.jpeg

예측: [3] apple / Score 0.8992768

입력: ./drive/My Drive/ByeongNam_test/banana_test/apple3.jpeg

예측: [3] apple / Score 0.8699498


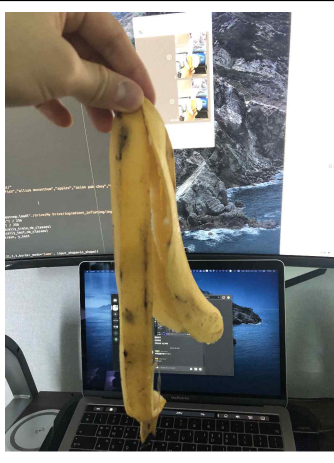
입력: ./drive/My Drive/ByeongNam_test/banana_test/apple4.jpeg

예측: [3] apple / Score 0.8681923

입력: ./drive/My Drive/ByeongNam_test/banana_test/apple5.jpeg



예측: [3] apple / Score 0.4982119

나. 바나나

	
banana1.jpeg	banana5.jpeg

입력: ./drive/My Drive/ByeongNam_test/banana_test/banana1.jpeg
 예측: [4] banana / Score 0.8885753
 입력: ./drive/My Drive/ByeongNam_test/banana_test/banana2.jpeg
 예측: [4] banana / Score 0.8746124
 입력: ./drive/My Drive/ByeongNam_test/banana_test/banana3.jpeg
 예측: [4] banana / Score 0.8720334
 입력: ./drive/My Drive/ByeongNam_test/banana_test/banana4.jpeg
 예측: [4] banana / Score 0.9681797
 입력: ./drive/My Drive/ByeongNam_test/banana_test/banana5.jpeg
 예측: [4] banana / Score 0.9188954

다. 굴비

	
a_dried_yellow_corvina_fish1.jpeg	a_dried_yellow_corvina_fish5.jpeg

입력: ./drive/My Drive/ByeongNam_test/banana_test/a_dried yellow corvina fish1.jpeg
 예측: [0] a dried yellow corvina fish / Score 0.7194926
 입력: ./drive/My Drive/ByeongNam_test/banana_test/a_dried yellow convina fish2.jpeg
 예측: [0] a dried yellow corvina fish / Score 0.8728123
 입력: ./drive/My Drive/ByeongNam_test/banana_test/a_dried yellow convina fish3.jpeg
 예측: [0] a dried yellow corvina fish / Score 0.7992354
 입력: ./drive/My Drive/ByeongNam_test/banana_test/a_dried yellow convina fish4.jpeg
 예측: [0] a dried yellow corvina fish / Score 0.7845813
 입력: ./drive/My Drive/ByeongNam_test/banana_test/a_dried yellow convina fish5.jpeg
 예측: [0] a dried yellow corvina fish / Score 0.7508544

라. 청경채

	
park choy1.jpeg	pak choy4.jpeg

입력: ./drive/My Drive/ByeongNam_test/banana_test/asian pak choy1.jpeg

예측: [2] asian pak choy / Score 0.8708062

입력: ./drive/My Drive/ByeongNam_test/banana_test/asian pak choy2.jpeg

예측: [2] asian pak choy / Score 0.9194926

입력: ./drive/My Drive/ByeongNam_test/banana_test/asian pak choy3.jpeg

예측: [2] asian pak choy / Score 0.9298652

입력: ./drive/My Drive/ByeongNam_test/banana_test/asian pak choy4.png

예측: [2] asian pak choy / Score 0.8898345

식자재 비교 알고리즘 결과

- 식재료를 정확하게 촬영 시 준수한 정확도를 보여줌.
- 모델 학습이 완벽하지 않음. 사과와 같은 단면은 정확도가 낮음.
- 모델 정확도 개선을 위해 inflating 실시 후 피드백 예정.

7. Coding & DEMO

1. tflite-Android Studio 연동 Code

- "~.tflite", "~labels.txt" 설정
- **CameraActivity.java (tflite파일, labels파일 연동)**

```
package com.amitshekhar.tflite;

import android.graphics.Bitmap;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.wonderkiln.camerakit.CameraKitError;
import com.wonderkiln.camerakit.CameraKitEvent;
import com.wonderkiln.camerakit.CameraKitEventListener;
import com.wonderkiln.camerakit.CameraKitImage;
import com.wonderkiln.camerakit.CameraKitVideo;
import com.wonderkiln.camerakit.CameraView;

import java.util.List;
import java.util.concurrent.Executor;
import java.util.concurrent.Executors;

public class CameraActivity extends AppCompatActivity {

    private static final String MODEL_PATH = "ingredient_inflating_test1.tflite";
    private static final String LABEL_PATH = "ingredient_test_labels.txt";
    private static final int INPUT_SIZE = 50;

    private Classifier classifier;
```

```

private Executor executor = Executors.newSingleThreadExecutor();
private TextView textViewResult;
private Button btnDetectObject, btnToggleCamera;
private ImageView imageViewResult;
private CameraView cameraView;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_camera2);
    cameraView = findViewById(R.id.cameraView);
    imageViewResult = findViewById(R.id.imageViewResult);
    textViewResult = findViewById(R.id.textViewResult);
    textViewResult.setMovementMethod(new ScrollingMovementMethod());

    btnToggleCamera = findViewById(R.id.btnToggleCamera);
    btnDetectObject = findViewById(R.id.btnDetectObject);

    cameraView.addCameraKitListener(new CameraKitEventListener() {
        @Override
        public void onEvent(CameraKitEvent cameraKitEvent) {
        }

        @Override
        public void onError(CameraKitError cameraKitError) {
        }

        @Override
        public void onImage(CameraKitImage cameraKitImage) {
            Bitmap bitmap = cameraKitImage.getBitmap();

            bitmap = Bitmap.createScaledBitmap(bitmap, INPUT_SIZE, INPUT_SIZE, false);

            imageViewResult.setImageBitmap(bitmap);

            /*final Bitmap finalBitmap = bitmap;
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    imageViewResult.setImageBitmap(finalBitmap);
                }
            });*/

            final List<Classifier.Recognition> results = classifier.recognizeImage(bitmap);

            textViewResult.setText(results.toString());
        }

        @Override
        public void onVideo(CameraKitVideo cameraKitVideo) {
        }
    });

    btnToggleCamera.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            cameraView.toggleFacing();
        }
    });

    btnDetectObject.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            cameraView.captureImage();
        }
    });

    initTensorFlowAndLoadModel();
}

@Override
protected void onResume() {
    super.onResume();
    cameraView.start();
}

@Override
protected void onPause() {
    cameraView.stop();
    super.onPause();
}

@Override
protected void onDestroy() {
    super.onDestroy();
}

```

```

executor.execute(new Runnable() {
    @Override
    public void run() {
        classifier.close();
    }
});

private void initTensorFlowAndLoadModel() {
    executor.execute(new Runnable() {
        @Override
        public void run() {
            try {
                classifier = TensorFlowImageClassifier.create(
                    getAssets(),
                    MODEL_PATH,
                    LABEL_PATH,
                    INPUT_SIZE);
                makeButtonVisible();
            } catch (final Exception e) {
                throw new RuntimeException("Error initializing TensorFlow!", e);
            }
        }
    });
}

private void makeButtonVisible() {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            btnDetectObject.setVisibility(View.VISIBLE);
        }
    });
}
}

```

- 분류된 이미지 정보 주고 받기

▪ Classifier.java

```

package com.amitshekhar.tflite;

import android.graphics.Bitmap;
import java.util.List;

public interface Classifier {
    class Recognition {
        private final String id;

        private final String title;

        private final Float confidence;

        public Recognition(
            final String id, final String title, final Float confidence) {
            this.id = id;
            this.title = title;
            this.confidence = confidence;
        }

        public String getId() {
            return id;
        }

        public String getTitle() {
            return title;
        }

        public Float getConfidence() {
            return confidence;
        }

        @Override
        public String toString() {
            String resultString = "";
            if (id != null) {
                resultString += "[" + id + " ] ";
            }
        }
    }
}

```

```

if (title != null) {
    resultString += title + " ";
}

if (confidence != null) {
    resultString += String.format("%.1f%% ", confidence * 100.0f);
}

return resultString.trim();
}
}

List<Recognition> recognizeImage(Bitmap bitmap);

void close();
}

```

- 이미지 분류하기

• TensorFlowImageClassifier.java(BATCH_SIZE, PIXEL_SIZE 설정)

```

package com.amitshekhar.tflite;

import android.annotation.SuppressLint;
import android.content.res.AssetFileDescriptor;
import android.content.res.AssetManager;
import android.graphics.Bitmap;

import org.tensorflow.lite.Interpreter;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.MappedByteBuffer;
import java.nio.channels.FileChannel;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import java.util.PriorityQueue;

public class TensorFlowImageClassifier implements Classifier {

    private static final int MAX_RESULTS = 3;
    private static final int BATCH_SIZE = 1;
    private static final int PIXEL_SIZE = 3;
    private static final float THRESHOLD = 0.1f;

    private Interpreter interpreter;
    private int inputSize;
    private List<String> labelList;

    private TensorFlowImageClassifier() {
    }

    static Classifier create(AssetManager assetManager,
        String modelPath,
        String labelPath,
        int inputSize) throws IOException {

        TensorFlowImageClassifier classifier = new TensorFlowImageClassifier();
        classifier.interpreter = new Interpreter(classifier.loadModelFile(assetManager, modelPath));
        classifier.labelList = classifier.loadLabelList(assetManager, labelPath);
        classifier.inputSize = inputSize;

        return classifier;
    }

    @Override
    public List<Recognition> recognizeImage(Bitmap bitmap) {
        ByteBuffer byteBuffer = convertBitmapToByteBuffer(bitmap);
        float[][] result = new float[1][labelList.size()];
        interpreter.run(byteBuffer, result);
        return getSortedResult(result);
    }

    @Override
    public void close() {
        interpreter.close();
    }
}

```

```

interpreter = null;
}

private MappedByteBuffer loadModelFile(AssetManager assetManager, String modelPath) throws IOException {
    AssetFileDescriptor fileDescriptor = assetManager.openFd(modelPath);
    FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());
    FileChannel fileChannel = inputStream.getChannel();
    long startOffset = fileDescriptor.getStartOffset();
    long declaredLength = fileDescriptor.getDeclaredLength();
    return fileChannel.map(FileChannel.MapMode.READ_ONLY, startOffset, declaredLength);
}

private List<String> loadLabelList(AssetManager assetManager, String labelPath) throws IOException {
    List<String> labelList = new ArrayList<>();
    BufferedReader reader = new BufferedReader(new InputStreamReader(assetManager.open(labelPath)));
    String line;
    while ((line = reader.readLine()) != null) {
        labelList.add(line);
    }
    reader.close();
    return labelList;
}

private ByteBuffer convertBitmapToByteBuffer(Bitmap bitmap) {
    // ByteBuffer byteBuffer = ByteBuffer.allocateDirect(BATCH_SIZE * inputSize * inputSize * PIXEL_SIZE);
    ByteBuffer byteBuffer = ByteBuffer.allocateDirect(3 * inputSize * inputSize * 4);
    byteBuffer.order(ByteOrder.nativeOrder());
    int[] intValues = new int[inputSize * inputSize];
    bitmap.getPixels(intValues, 0, bitmap.getWidth(), 0, 0, bitmap.getWidth(), bitmap.getHeight());
    int pixel = 0;
    for (int i = 0; i < inputSize; ++i) {
        for (int j = 0; j < inputSize; ++j) {
            final int val = intValues[pixel++];
            byteBuffer.put((byte) ((val >> 16) & 0xFF));
            byteBuffer.put((byte) ((val >> 8) & 0xFF));
            byteBuffer.put((byte) (val & 0xFF));
        }
    }
    return byteBuffer;
}

@SuppressWarnings("DefaultLocale")
private List<Recognition> getSortedResult(float[][] labelProbArray) {

    PriorityQueue<Recognition> pq =
        new PriorityQueue<>(
            MAX_RESULTS,
            new Comparator<Recognition>() {
                @Override
                public int compare(Recognition lhs, Recognition rhs) {
                    return Float.compare(rhs.getConfidence(), lhs.getConfidence());
                }
            });

    for (int i = 0; i < labelList.size(); ++i) {
        float confidence = (labelProbArray[0][i] * 100) / 127.0f;
        if (confidence > THRESHOLD) {
            pq.add(new Recognition("" + i,
                labelList.size() > i ? labelList.get(i) : "unknown",
                confidence));
        }
    }

    final ArrayList<Recognition> recognitions = new ArrayList<>();
    int recognitionsSize = Math.min(pq.size(), MAX_RESULTS);
    for (int i = 0; i < recognitionsSize; ++i) {
        recognitions.add(pq.poll());
    }

    return recognitions;
}

```

- 데이터 부풀리기

- [data_inflating.py](#)

```

# 데이터 부풀리기 (ImageData Generator)
import numpy as np
import os
from os import listdir
from os.path import isfile, join
from PIL import Image
np.random.seed(3)

```





















```

from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
data_datagen = ImageDataGenerator(rescale=1./255)
data_datagen = ImageDataGenerator(rescale=1./255,
rotation_range=30,
shear_range=5.5,
width_shift_range=0.1,
height_shift_range=0.1,
zoom_range=0.,
horizontal_flip=True,
vertical_flip=True,
fill_mode='nearest')
filename_in_dir = []
for root, dirs, files in os.walk('./drive/My Drive/ingredient/banana'):
    for fname in files:
        full_fname = os.path.join(root, fname)
        filename_in_dir.append(full_fname)
    for file_image in filename_in_dir:
        #print file_image
        img = load_img(file_image)
        x = img_to_array(img)
        x = x.reshape((1,) + x.shape)
        i = 0
    for batch in data_datagen.flow(x,save_to_dir='./drive/My Drive/ingredient_inflating/banana',
save_prefix='banana', save_format='jpg'):
        i += 1
    if i > 30:
        break

```

2. DEMO


1. 시작 화면	2. 회원가입 화면
   <p>© 2019. HonCook All Rights Reserved</p>	 <h3>회원가입</h3> <div> <input type="text" value="ID를 입력해주세요."/> </div> <div> <input type="text" value="Password를 입력해주세요."/> </div> <div> <input type="text" value="위와 동일한 Password를 입력해주세요."/> </div> <div> <input type="text" value="이름을 입력해 주세요."/> </div> <div> <input type="text" value="나이를 입력해주세요."/> </div> <div> <input type="button" value="회원가입"/> </div>
3. 로그인 화면	4. Home - 레시피 종류별 보기 및 검색

<div data-bbox="371 280 470 398">  <p>Hon Cook</p> </div> <div data-bbox="617 315 746 367">로그인</div> <div data-bbox="347 566 863 602"> <input type="text" value="ID"/> </div> <div data-bbox="347 629 863 663"> <input type="text" value="Password"/> </div> <div data-bbox="539 687 671 734"> <input type="button" value="로그인"/> </div> <div data-bbox="545 759 662 779">계정이 없으신가요?</div>	<div data-bbox="930 280 1029 398">  <p>test님</p> </div> <div data-bbox="1088 309 1391 360">Food Recipe</div> <div data-bbox="959 405 1029 495">  <p>한식</p> </div> <div data-bbox="1066 405 1136 495">  <p>일식</p> </div> <div data-bbox="1173 405 1243 495">  <p>중식</p> </div> <div data-bbox="1279 405 1350 495">  <p>양식</p> </div> <div data-bbox="911 515 1415 542"> <input type="text" value="검색할 레시피 입력"/> </div> <div data-bbox="943 584 986 640">  </div> <div data-bbox="1027 555 1144 582">계란말이주먹밥</div> <div data-bbox="1027 580 1069 602">한식</div> <div data-bbox="1027 600 1369 674"> <p>밥 2공기, 계란 4개, 말린 표고버섯 2개, 당근 30g, 미나리 10줄기, 소금 약간, 식용유 약간, 오이 1/2개, 다시마 1조각, 식초(배합초) 1큰술, 설탕 1큰술, 소금 1작은술</p> </div> <div data-bbox="943 719 986 775">  </div> <div data-bbox="1027 689 1192 716">치킨개밥과 카레볶음밥</div> <div data-bbox="1027 714 1069 736">양식</div> <div data-bbox="1027 734 1399 799"> <p>닭 200g, 카레 30g, 밥 1공기, 식용유 약간, 양송이버섯 4개, 피망 1/2개, 파프리카 1/2개, 방울토마토 약간, 고지 적당량</p> </div> <div data-bbox="943 853 986 909">  </div> <div data-bbox="1027 826 1112 853">무지개초밥</div> <div data-bbox="1027 851 1069 873">일식</div> <div data-bbox="1027 871 1382 947"> <p>참치 30g, 초새우 4마리, 고등어 50g, 연어 50g, 김 2장, 초밥 200g, 계란 1개, 아보카도 1/2개, 단무지 약간, 오이 약간, 와사비 약간, 소금 약간, 설탕 2작은술, 청주 2작은술, 미나리 2줄기, 식용유 2큰술</p> </div> <div data-bbox="930 987 986 1055">  <p>Home</p> </div> <div data-bbox="1027 1010 1069 1055">  <p>Camera</p> </div> <div data-bbox="1141 1010 1173 1055">  <p>Add</p> </div> <div data-bbox="1244 1010 1276 1055">  <p>Share</p> </div> <div data-bbox="1347 1010 1402 1055">  <p>My Page</p> </div>
<div data-bbox="330 1081 670 1117">4-a. Home - 한식 선택 시</div>	<div data-bbox="888 1081 1232 1117">4-b. Home - 일식 선택 시</div>
<div data-bbox="371 1171 470 1249">  <p>Hon Cook</p> </div> <div data-bbox="534 1171 831 1225">Food Recipe</div> <div data-bbox="403 1267 474 1357">  <p>한식</p> </div> <div data-bbox="512 1267 582 1357">  <p>일식</p> </div> <div data-bbox="624 1267 694 1357">  <p>중식</p> </div> <div data-bbox="735 1267 805 1357">  <p>양식</p> </div> <div data-bbox="359 1375 858 1402"> <input type="text" value="검색할 레시피 입력"/> </div> <div data-bbox="384 1447 427 1503">  </div> <div data-bbox="474 1417 587 1442">계란말이주먹밥</div> <div data-bbox="474 1442 513 1464">한식</div> <div data-bbox="474 1462 809 1536"> <p>밥 2공기, 계란 4개, 말린 표고버섯 2개, 당근 30g, 미나리 10줄기, 소금 약간, 식용유 약간, 오이 1/2개, 다시마 1조각, 식초(배합초) 1큰술, 설탕 1큰술, 소금 1작은술</p> </div> <div data-bbox="384 1581 427 1637">  </div> <div data-bbox="474 1552 553 1576">굴뱅이덮밥</div> <div data-bbox="474 1576 513 1599">한식</div> <div data-bbox="474 1597 850 1671"> <p>밥 4공기, 굴뱅이 200g, 오이 1개, 당근 1/2개, 양파 1/2개, 깻잎 10장, 고추장 2큰술, 고춧가루 1큰술, 설탕 2큰술, 간장 1/2큰술, 식초 1큰술, 마늘 1/2큰술, 깨소금 1/2큰술, 참기름 1큰술, 초창기 1작은술</p> </div> <div data-bbox="368 1861 411 1917">  <p>Home</p> </div> <div data-bbox="474 1861 517 1917">  <p>Camera</p> </div> <div data-bbox="592 1861 620 1917">  <p>Add</p> </div> <div data-bbox="694 1861 722 1917">  <p>Share</p> </div> <div data-bbox="793 1861 836 1917">  <p>My Page</p> </div>	<div data-bbox="930 1171 1029 1249">  <p>Hon Cook</p> </div> <div data-bbox="1088 1171 1399 1225">Food Recipe</div> <div data-bbox="959 1267 1029 1357">  <p>한식</p> </div> <div data-bbox="1066 1267 1136 1357">  <p>일식</p> </div> <div data-bbox="1173 1267 1243 1357">  <p>중식</p> </div> <div data-bbox="1279 1267 1350 1357">  <p>양식</p> </div> <div data-bbox="911 1375 1415 1402"> <input type="text" value="검색할 레시피 입력"/> </div> <div data-bbox="943 1447 986 1503">  </div> <div data-bbox="1027 1417 1117 1442">무지개초밥</div> <div data-bbox="1027 1442 1069 1464">일식</div> <div data-bbox="1027 1462 1386 1536"> <p>참치 30g, 초새우 4마리, 고등어 50g, 연어 50g, 김 2장, 초밥 200g, 계란 1개, 아보카도 1/2개, 단무지 약간, 오이 약간, 와사비 약간, 소금 약간, 설탕 2작은술, 청주 2작은술, 미나리 2줄기, 식용유 2큰술</p> </div> <div data-bbox="943 1581 986 1637">  </div> <div data-bbox="1027 1552 1123 1576">일본식닭조림</div> <div data-bbox="1027 1576 1069 1599">일식</div> <div data-bbox="1027 1597 1374 1671"> <p>닭 1마리, 마늘 3톨, 생강 1개, 소금 약간, 간장 4큰술, 올리고당 3큰술, 청주 3큰술, 1, 마늘과 생강은 편으로 썰어둔다.</p> </div> <div data-bbox="930 1861 973 1917">  <p>Home</p> </div> <div data-bbox="1027 1861 1069 1917">  <p>Camera</p> </div> <div data-bbox="1141 1861 1173 1917">  <p>Add</p> </div> <div data-bbox="1244 1861 1276 1917">  <p>Share</p> </div> <div data-bbox="1347 1861 1402 1917">  <p>My Page</p> </div>
<div data-bbox="330 1939 657 1975">4-c. Home - 중식 선택시</div>	<div data-bbox="888 1939 1232 1975">4-d. Home - 양식 선택 시</div>


<div data-bbox="368 315 472 398"> </div> <div data-bbox="534 315 831 369"> <h1>Food Recipe</h1> </div> <div data-bbox="403 412 472 472"> </div> <div data-bbox="512 412 580 472"> </div> <div data-bbox="624 412 692 472"> </div> <div data-bbox="735 412 804 472"> </div> <div data-bbox="359 521 491 546"> <input type="text" value="검색할 레시피 입력"/> </div> <div data-bbox="384 591 432 647"> </div> <div data-bbox="474 562 569 586"> <p>짜장해물탕밥</p> </div> <div data-bbox="474 586 507 609"> <p>중식</p> </div> <div data-bbox="474 607 847 678"> <p>밥 2그릇, 돼지고기 150g, 오징어 1/2마리, 조개살 100g, 홍합 200g, 양파 1개, 당근 20g, 호박 100g, 대파 1뿌리, 목이버섯 6장, 고춧가루 3큰술, 간장 1큰술, 다진마늘 1큰술, 참기름 1큰술, 소금 약간, 후춧가루 약간</p> </div> <div data-bbox="384 723 432 779"> </div> <div data-bbox="474 694 529 719"> <p>잡채밥</p> </div> <div data-bbox="474 719 507 741"> <p>중식</p> </div> <div data-bbox="474 739 836 810"> <p>당면 50g, 돼지고기 100g, 표고버섯 1장, 부추 30g, 호박 1/4개, 당근 1/2개, 밥 1공기, 다진마늘 1큰술, 진간장 4큰술, 참기름 1큰술, 소금 약간, 후춧가루 약간</p> </div> <div data-bbox="368 1010 408 1059"> </div> <div data-bbox="474 1032 521 1059"> </div> <div data-bbox="592 1032 616 1059"> </div> <div data-bbox="695 1010 727 1059"> </div> <div data-bbox="791 1010 839 1059"> </div>	<div data-bbox="927 315 1031 398"> </div> <div data-bbox="1093 315 1390 369"> <h1>Food Recipe</h1> </div> <div data-bbox="962 412 1031 472"> </div> <div data-bbox="1070 412 1139 472"> </div> <div data-bbox="1182 412 1251 472"> </div> <div data-bbox="1294 412 1362 472"> </div> <div data-bbox="917 521 1050 546"> <input type="text" value="검색할 레시피 입력"/> </div> <div data-bbox="943 591 991 647"> </div> <div data-bbox="1032 562 1198 586"> <p>치킨케밥과 카레볶음밥</p> </div> <div data-bbox="1032 586 1066 609"> <p>양식</p> </div> <div data-bbox="1032 607 1406 678"> <p>닭 200g, 카레 30g, 밥 1공기, 식용유 약간, 양송이버섯 4개, 피망 1/2개, 파프리카 1/2개, 방울토마토 약간, 고치 적당량</p> </div> <div data-bbox="943 723 991 779"> </div> <div data-bbox="1032 694 1112 719"> <p>치즈크로켓</p> </div> <div data-bbox="1032 719 1066 741"> <p>양식</p> </div> <div data-bbox="1032 739 1410 810"> <p>계란흰자 2개, 피자치즈 50g, 밀가루(중력분) 50g, 식용유(튀김용) 적당량, 소금 약간, 후춧가루 약간, 계란흰자가 하얗게 될 때까지 거품기로 젓는다.</p> </div> <div data-bbox="927 1010 967 1059"> </div> <div data-bbox="1032 1032 1080 1059"> </div> <div data-bbox="1150 1032 1174 1059"> </div> <div data-bbox="1254 1010 1286 1059"> </div> <div data-bbox="1350 1010 1398 1059"> </div>
<p>4-e. Home - 검색어 입력 시</p>	<p>5. 레시피 상세화면</p>
<div data-bbox="368 1178 472 1261"> </div> <div data-bbox="534 1178 831 1232"> <h1>Food Recipe</h1> </div> <div data-bbox="403 1279 472 1339"> </div> <div data-bbox="512 1279 580 1339"> </div> <div data-bbox="624 1279 692 1339"> </div> <div data-bbox="735 1279 804 1339"> </div> <div data-bbox="359 1377 399 1402"> <input type="text" value="잡채"/> </div> <div data-bbox="384 1447 432 1503"> </div> <div data-bbox="474 1417 529 1442"> <p>잡채밥</p> </div> <div data-bbox="474 1442 507 1464"> <p>중식</p> </div> <div data-bbox="474 1462 836 1534"> <p>당면 50g, 돼지고기 100g, 표고버섯 1장, 부추 30g, 호박 1/4개, 당근 1/2개, 밥 1공기, 다진마늘 1큰술, 진간장 4큰술, 참기름 1큰술, 소금 약간, 후춧가루 약간</p> </div> <div data-bbox="368 1865 408 1915"> </div> <div data-bbox="474 1888 521 1915"> </div> <div data-bbox="592 1888 616 1915"> </div> <div data-bbox="695 1865 727 1915"> </div> <div data-bbox="791 1865 839 1915"> </div>	<div data-bbox="927 1137 1031 1261"> </div> <div data-bbox="1093 1178 1390 1232"> <h1>Food Recipe</h1> </div> <div data-bbox="1121 1290 1206 1402"> </div> <div data-bbox="1080 1406 1249 1438"> <p>계란말이주먹밥</p> </div> <div data-bbox="904 1438 989 1462"> <p>분류 : 한식</p> </div> <div data-bbox="904 1462 1388 1534"> <p>재료 : 밥 2공기, 계란 4개, 말린 표고버섯 2개, 당근 30g, 미나리 10줄기, 소금 약간, 식용유 약간, 오이 1/2개, 다시마 1조각, 식초(배합초) 1큰술, 설탕 1큰술, 소금 1/2작은술</p> </div> <div data-bbox="904 1574 1420 1859"> <p>방법 : 1. 다시마는 마른 행주로 닦은 후 냄비 바닥에 갈아 쌀을 넣어 고슬하게 밥을 짓고, 다시마를 빼낸 후 분량의 배합초를 넣어 버무려 식힌다. 2. 볼에 계란을 풀고 소금간을 한 뒤 프라이팬에 식용유를 두르고 계란 지단을 부친다. 3. 지단이 식으면 가로 x 세로가 20cm x 20cm 정도가 되도록 자른다. 4. 표고버섯을 불린 다음 얇게 채썰고, 오이는 돌려깎기한 후 껍질 부분만 채를 썰고, 당근은 곱게 채썰어 각각 볶은 후 소금으로 간을 한다. 5. 미나리는 잎을 떼고 끓는 소금물에 데친다. 6. 계란 지단 위에 밥을 펴고 볶아 둔 표고버섯, 오이, 당근을 올리고 감밥 말듯이 말아준 후 데친 미나리로 한입 크기 마다</p> </div> <div data-bbox="1080 1877 1249 1901"> <p>나의 레시피 목록에 추가</p> </div>
<p>6. Add - 레시피 추가화면</p>	<p>7. My Page - 내 레시피 종류 화면</p>

The image shows a mobile application interface for a recipe book. At the top, there is a circular profile picture of a chef with the text 'Hon Cook' below it. To the right, the text 'test님' is displayed above the large title 'My Page'. Below the profile picture is a larger chef icon. Underneath this is the text '아이다디 : test'. A blue button labeled 'MY RECIPE' is positioned below the text. The main content area displays a list of recipes, each preceded by a small chef icon. The first recipe is '계란말이주먹밥' (Egg Roll Maki) and the second is '굴뱅이덮밥' (Gulbaengi Ddeppab). The bottom of the screen features a navigation bar with five icons: a house for 'Home', a camera for 'Camera', a plus sign for 'Add', a share icon for 'Share', and a person icon for 'My Page'.

8-a. Camera - 관련 레시피 추천 클릭 시



banana 관련 레시피 추천




바나나머핀

양식

박력분 500g, 바나나 500g, 버터 350g, 설탕 350g, 계란 8개, 베이킹파우더 25g, 베이킹소다 10g


1. 버터와 설탕 1/2을 크림화한다.



바나나스프레드토스트

양식

바나나 2개, 식빵 3개, 딸기 약간, 물 2큰술, 꿀(또는 청정원 올리고당) 2큰술, 계핏가루 약간, 코코넛가루 약간, 버터 1큰술




바나나초코카스텔라를

양식

바나나 1개, 초콜릿 50g, 카스텔라 120g, 생크림 3큰술, 럼주 1/2작은술

1. 초콜릿을 잘게 다쳐 생크림을 넣고 천자레인지에



바나나컵케이크

양식

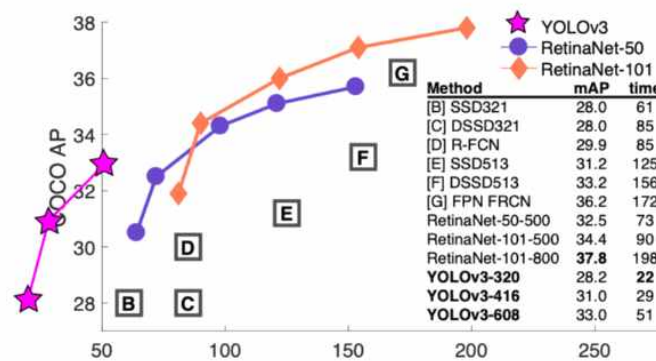
버터 90g, 계란 1개, 바나나 4개, 박력분 80g, 설탕 80g, 베이킹파우더 1작은술

1. 큰 볼에 상온 상태의 버터와 설탕을 넣고 고무주걱으로

Ⅲ. 결 론

1. 연구 결과

- 본 연구는 정제된 식재료 DataSet을 이용하는 것과 yolo V3 pb 모델로 기존 모델보다 복잡도를 줄여 모바일 환경에서도 realtime으로 구동할 수 있다는 점이 핵심이다. 이를 활용하여 초보 요리사에게 낯선 식재료를 판별하고 추천해준다.
- Object Detection 모델 중 Yolo 모델이 모바일 환경에서 가장 적합한 성능을 내주었다.



backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19[15]	74.1	91.8	7.29	1246	171
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152[5]	77.6	93.8	29.4	1090	37
Darknet-53	77.2	93.8	18.7	1457	78

- 모델 생성 시 epoch 는 150~200으로 만드는 것이 가장 준수한 정확도를 보여줬다.
- 개발과정에서 가장 힘들었던 점은 단연코 Dataset 가공 문제이다. 사진을 크롤링해서 모으다보면 피로도가 많이 쌓인다. 이를 해결하는 방법으로는 소비자가 직접 식재료 DataSet을 찍어 등록하는 방법이지만 이는 기술적으로 실현이 불가능한 부분이 크다. 따라서 딥러닝을 이용한 애플리케이션을 제작한 기업에게 자문을 구했지만 DataSet은 구매해서 프로젝트를 진행하는 편이 효율적이며 가장 좋은 결과를 가져다준다는 말을 들었다.
- 최근 코로나19로 사회적 거리두기 제한으로 식당들이 문을 일찍 닫으며 식당보다 집밥을 찾는 이들이 많아졌다. 예를 보면 배달음식은 지난해보다 115% 증가로 급성장했으며 주방 제품에 대해서도 지난해보다 30% 증가한 판매량을 기록했다. 따라서, 기대효과로 해당 연구결과물을 활용해 사람들이 편리함을 얻을 것으로 예측된다.

2. 제품 제작 소요 제품 목록

- 개발에 필요한 SW는 아래와 같으며 따로 소요비용은 없다.

Software Development Environment		
Server	Apache Tomcat	ver. 8.5.34
DBMS	MySQL	ver. 8.0.12
Android IDE	Android Studio	ver. 3.1.4
Deep Learning	Tensorflow	ver 1.15.0
"	Yolo	ver 3

- 개발에 필요한 HW는 안드로이드 핸드폰 중 갤럭시s6를 사용했다. 팀원 개인 핸드폰을 사용했기에 소요 비용은 없다.



참고 문헌

URL 링크 ▼

http://kostat.go.kr/portal/korea/kor_nw/1/2/2/index.board?bmode=read&aSeq=370806&pageNo=1&rowNum=10&amSeq=&sTarget=title&sTxt 통계청 - 1인가구 증가 보고서

<http://mediask.co.kr/2228> SK - 요리 버즈량 증가

<https://kutar37.tistory.com/entry/파이썬-협업필터링Collaborative-Filtering-추천-알고리즘-1>
kutar37 - 협업필터링 알고리즘

https://www.youtube.com/watch?v=BS600zOGX4E&list=PLIMkM4tgfnLS0jrEJN31gZATbcj_MpUm ,

<http://hunkim.github.io/ml/> ,

<https://github.com/hunkim/DeepLearningZeroToAll/>

Sung kim , 모두를 위한 딥러닝

https://github.com/Yoonkyung/Movie_recommendation
recommondation project

yoonkyung,

Movie

서재 ▼

실무가 훤히 보이는 머신러닝 & 딥러닝 - 마창수, 최재철