



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공 학 석 사 학 위 논 문

스마트폰을 활용한 음식인식 및
칼로리 추정 기법

2023년 8월

부 경 대 학 교 대 학 원

IT융합응용공학과

허 지 원

공 학 석 사 학 위 논 문

스마트폰을 활용한 음식인식 및 칼로리 추정 기법

지도교수 송 하 주

이 논문을 공학석사 학위논문으로 제출함.

2023년 8월

부 경 대 학 교 대 학 원

IT융합응용공학과

허 지 원

허지원의 공학석사 학위논문을 인준함.

2023년 8월 18일



위 원 장 공학박사 김 영 봉 (인)

위 원 공학박사 권 오 흠 (인)

위 원 공학박사 송 하 주 (인)

목 차

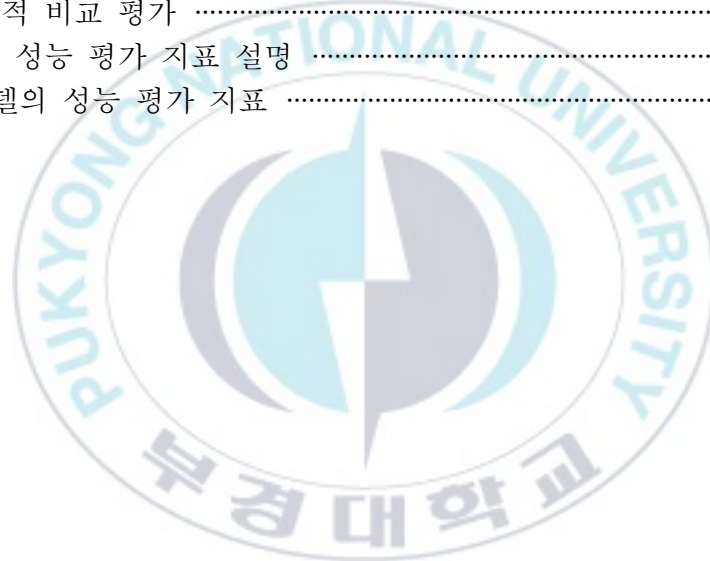
표 목차	iii
그림 목차	iv
Abstract	vi
1. 서 론	1
1.1. 연구의 필요성	1
1.2. 연구의 목적 및 방법	2
1.3. 논문의 구성	3
2. 관련 연구	4
2.1. 기존 연구	4
2.2. YOLO	5
2.3. 전이학습(transfer learning)	7
2.4. 음식영양정보	7
3. 스마트폰을 활용한 음식인식 및 칼로리 추정	8
3.1. YOLOv5를 이용한 음식인식 모델 개발	8
3.1.1. 데이터 수집	10
3.1.2. 데이터 라벨링	12
3.1.3. 데이터 증강	15
3.1.4. 딥러닝 모델 학습 방법	18
3.2. 제안 시스템의 동작 과정	19
3.2.1. 참조 객체 기반 칼로리 추정 방법	20
3.2.2. 참조 객체가 존재하지 않는 경우	24
3.3. 인식된 음식에 대한 영양정보의 정보 취득	24
4. 음식인식 및 칼로리 추정 앱 구현	27
4.1. 앱 개발환경	27
4.2. 앱 구현 화면	28
4.3. 성능평가	31
4.3.1. 정성적 비교 평가	32
4.3.2. 정량적 비교 평가	33
4.3.3. 한계점	38

5. 결 론	40
참고 문헌	41



표 목 차

표 1. 한국 이미지(음식) 데이터 종류 예시	11
표 2. 라벨링 데이터 예시	14
표 3. 데이터 종류 및 개수	17
표 4. 참조객체 및 그 크기	21
표 5. 그림 11의 딥러닝 모델을 통한 예측 결과값	22
표 6. 음식영양정보 데이터	24
표 7. 음식영양정보 데이터 출력 결과 예시	26
표 8. 정성적 비교 평가	32
표 9. 모델 성능 평가 지표 설명	35
표 10. 모델의 성능 평가 지표	36



그 립 목 차

그림 1. 다중 객체 인식 이미지	2
그림 2. YOLO 모델의 기본 동작 원리	5
그림 3. YOLO 모델의 객체탐지 과정	6
그림 4. 스마트폰을 활용한 음식 인식 및 칼로리 추정 알고리즘	9
그림 5. 한국 음식 이미지 예시	11
그림 6. labelimg 프로그램	12
그림 7. 라벨링 데이터 예시	13
그림 8. Roboflow 라벨링 된 데이터	14
그림 9. 데이터 증강 예시	16
그림 10. YOLOv5 모델 종류	18
그림 11. 스마트폰을 활용한 음식 인식 및 칼로리 추정 의사코드	19
그림 12. 참조 객체가 존재하는 경우 이미지 예시	22
그림 13. 음식영양정보 데이터 예시	25
그림 14. 앱 개발환경	28
그림 15. 앱 초기 화면	29
그림 16. 참조 객체가 존재하는 경우	30
그림 17. 참조 객체가 존재하지 않는 경우	31
그림 18. YOLOv5 모델의 학습과정 및 성능지표 1	33
그림 19. YOLOv5 모델의 학습과정 및 성능지표 2	34
그림 20. Precision-Recall Curve	36
그림 21. Confusion matrix	37
그림 22. 딥러닝 모델의 예측 결과	39

스마트폰을 활용한 음식인식 및 칼로리 추정기법

허지원

부경대학교 대학원 IT융합응용공학과

요 약

건강한 삶을 영위하기 위해서는 적절한 칼로리와 영양을 갖춘 식이조절이 중요하다. 이를 위해 본 논문에서는 스마트폰을 통해 촬영한 한 장의 사진을 사용하여 식사에 포함된 영양소와 칼로리를 추정할 수 있는 기법을 제안한다. 제안 기법은 딥러닝을 활용하여 사진에 포함된 개별 음식들을 인식하고 그 칼로리를 추정한다. 이 과정에서 사진 속에 포함된 참조 객체와의 상대적인 크기 비교를 통해 인식된 음식의 양을 대략적으로 계산하였다. 제안기법을 구현한 안드로이드 앱에서는 음식 데이터베이스를 활용하여 식사에 포함된 영양정보를 함께 제공하도록 하여 활용성을 높였다. 음식 인식을 위한 이미지 인식을 위해서는 YOLOv5를 사용하였고 일반 가정식 식단들에 대한 테스트를 통해 제안 기법의 실용성을 확인하였다.

Food Recognition and Calorie Estimation Technique Using A Smartphones

JI WON HEO

Department of IT Convergence and Application Engineering
The Graduate School
Pukyong National University

Abstract

In order to lead a healthy life, it is important to control a diet with adequate calories and nutrients. To this end, this paper proposes a technique that can estimate the nutrients and calories contained in a meal using a single picture taken through a smartphone. The proposed technique utilizes deep learning to recognize individual foods included in a picture and estimate their calories. In this process, the amount of recognized food was roughly calculated through relative size comparison with the reference object included in the picture. In the Android app that implemented the proposed technique, the food database was used to provide nutritional information included in the meal, increasing its usability. For image recognition for food recognition, YOLOv5 was used, and the practicality of the proposed method was confirmed through tests on Korean home-cooked meals.

1. 서론

1.1. 연구의 필요성

현대 사회에서 건강한 삶에 대한 관심은 증가하고 있다. 로열 필립스의 ‘아시아 국가 개인건강관리 실태조사’ 결과에 따르면 한국, 싱가포르, 태국, 인도네시아 4개국 4,000명을 대상으로 실시한 설문 결과 코로나19 이후 아시아 국가 내 건강한 식습관 및 생활습관 등 전반적인 건강에 대한 사람들의 관심이 증가하고 있다[1].

영양 섭취는 건강과 밀접한 관련이 있다. 영양의 불균형한 섭취는 건강에 부정적인 영향을 미칠 수 있으므로, 균형 잡힌 식사 관리가 중요하다. 따라서 건강한 삶을 위해서는 영양과 칼로리의 균형 잡힌 식사 관리가 필요하다.

국민건강보험공단 건강검진 자료에 따르면 지난 10년간 비만 유병률이 꾸준히 증가하고 있다[2]. 비만과 만성질환, 중증질환은 밀접한 관계를 가지고 있으며, 예방 및 관리를 위한 연구가 진행되고 있다. 연구에 따르면 비만을 가지고 있는 사람은 그렇지 않은 사람보다 심혈관 질환과 같은 만성질환에 노출될 확률이 높다[3]. 또한 비만인 사람들은 비만이 아닌 사람들에 비해 대장암과 같은 중증질환에 걸릴 확률이 더 높다[4]. 건강한 삶을 위해서는 만성질환 및 중증질환과 밀접한 관련이 있는 비만을 예방해야 한다. 비만을 예방하기 위해 개인의 활동량, 나이, 성별, 신체 조건에 따라 적당한 칼로리를 섭취 하는지에 대한 모니터링이 필요하다.

본 연구에서는 사용자가 섭취한 음식을 모니터링하기 위한 앱을 제안하고자 한다. 스마트폰의 앱으로 이미지 인식을 통해 사용자가 간편하

게 섭취한 음식을 모니터링할 수 있도록 섭취 음식의 종류와 칼로리 및 영양정보를 제공한다.

1.2. 연구의 목적 및 방법

본 논문은 스마트폰을 통해 간단하게 섭취한 음식에 대한 모니터링을 하는 앱을 제안한다. 앱은 다중 객체탐지(object detection) 딥러닝 모델을 통해 다중 음식 인식 및 음식 열량을 추정한다. 다중 객체 인식 모델은 이미지 안에 있는 여러 개의 객체를 인식하는 모델이다. 그림 1과 같이 하나의 이미지 안에 포함된 여러 음식을 동시에 인식한다. 본 연구에서는 전형적인 한식 식단에 포함되는 총 111가지의 음식을 인식한다. 이를 위해 55,919장의 음식 이미지를 수집하였으며 잘 알려진 객체탐지 모델인 YOLOv5를 전이학습 기법으로 트레이닝 하였다. YOLO는 일단계(one stage) 객체탐지 모델의 일종이며 스마트폰과 같은 엣지 디바이스에 사용하기 적합한 특성이 있다고 알려져 있다.

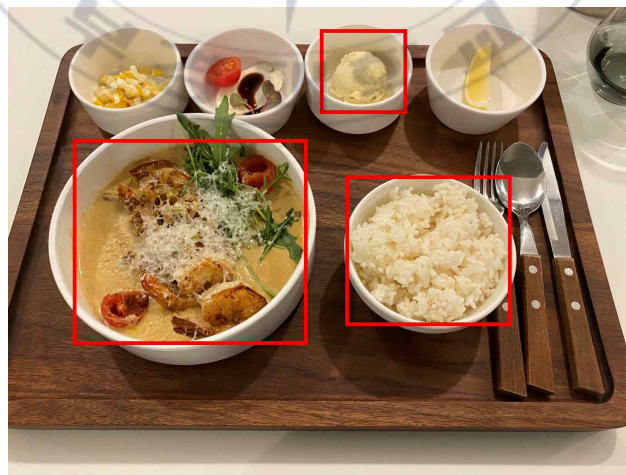


그림 1. 다중 객체 인식 이미지

다중 객체 인식 모델을 사용하면 여러 객체의 위치와 종류를 식별할 수 있다. 다중 객체 인식 딥러닝 모델을 통해 얻은 정보로 식단에 포함된 각각의 음식의 칼로리를 추정한다. 음식 영양 및 칼로리 정보는 공공데이터포털의 식품 영양성분 정보를 OPEN API를 통해 받아 온다. 참조 객체가 존재할 경우 인식된 참조 객체의 크기를 이용하여 음식의 양을 추정하고, 추정한 양을 반영하여 칼로리와 영양정보를 계산한다. 참조 객체가 존재하지 않는 경우, 공공데이터포털 식품의약품안전처_식품 영양성분 정보의 1회 제공량을 기준으로 영양정보를 계산한다.

1.3. 논문의 구성

본 논문의 구성은 다음과 같다. 2장은 관련 연구와 YOLO 객체탐지 모델에 대하여 소개한다. 3장은 스마트폰을 활용한 음식인식 및 인식된 음식의 칼로리를 추정하는 방법을 제시한다. 4장은 음식인식 및 칼로리 추정 앱의 개발환경과 구현 방법에 대해서 다루며, 제시한 앱의 성능을 평가한다. 5장은 본 연구의 결론 및 추후 연구 방향을 논의한다.

2. 관련 연구

2.1. 기존 연구

딥러닝은 이미지 인식 및 분류에 뛰어난 성능을 보여주며, 음식 이미지 분류에도 적용되고 있다. Liu는 UEC Food 100 음식 이미지 데이터를 사용하여 음식 분류를 위한 CNN 모델 기반 딥러닝 모델을 설계하였다[5]. M.Chun등은 한국 음식 이미지를 범주 별 특징으로 정의하고 한국 음식 이미지 데이터의 특징에 맞게 딥러닝 모델을 설계하였다[6]. 위의 두 연구 모두 한 가지 음식에 대한 분류를 위한 딥러닝 모델 개발에 초점을 맞췄다. 사용자의 음식 섭취를 모니터링하기 위한 앱에서는 한 가지 음식만 분류하는 딥러닝 모델을 사용할 경우 음식을 여러 번 찍어야 하는 단점이 있다. V.B.Kasyap등은 음식의 평균적인 양을 기반으로 딥러닝 모델을 학습시켜 칼로리를 추정하였다[7]. V.B.Kasyap등의 연구에서는 음식의 종류가 14가지로 제한적이었으며 사과, 바나나 등 주로 과일의 평균적인 크기와 무게를 기반으로 칼로리를 추정하고 음식의 종류도 다양하지 않다. Sutedja등은 당뇨병 환자를 타겟으로 음식 섭취 모니터링과 혈당 수치에 대한 모니터링을 위한 앱을 제안하였다[8]. 하지만 섭취한 음식의 종류와 음식의 양을 직접 입력해야 하는 번거로움이 있다. B. Alshujaa과 S. Naritomi는 iOS의 ARKit로 음식의 길이를 구하여 칼로리를 추정하고 음식을 분류하는 방법을 제안하였다[9-10]. 하지만 iOS 운영체제를 기반으로 한 전자기기를 타겟으로 하기 때문에 안드로이드 기반 전자기기에서는 사용할 수 없다는 단점이 있다. 본 연구에서는 iOS와 안드로이드 운영체제에서 동시에 사용할 수 있는 음식인식 및 칼로리 추정 앱을 제안한다.

2.2. YOLO

YOLO(You Only Look Once)는 일단계(one-stage) 객체탐지 딥러닝 모델로 단일 신경망을 통해 이미지의 객체 분류와 객체탐지를 한 번에 수행하는 알고리즘이다. YOLO 모델의 특징은 다른 객체탐지 모델에 비해 빠른 속도와 높은 정확도를 가지고 있는 것으로 알려져 있다[11].

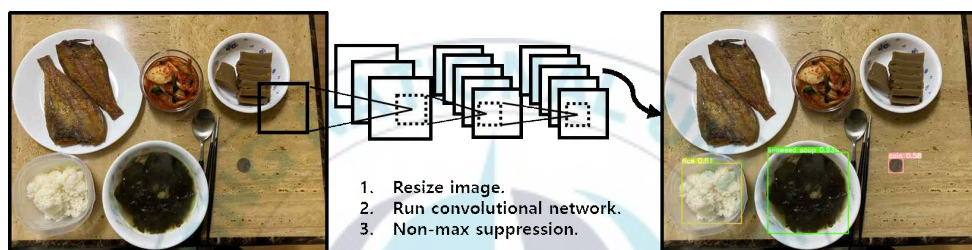


그림 2. YOLO 모델의 기본 동작 원리[11]

YOLO 모델의 기본 동작 원리는 그림 2와 같다. 입력으로 받은 이미지의 크기를 재조정하고, 단일 신경망을 통해 이미지에 포함된 객체의 위치를 인식하고, 인식된 객체들을 분류한다. 그런 다음 NMS(Non-Maximum Suppression)를 통해 중복 인식된 객체들을 제거한다[11].

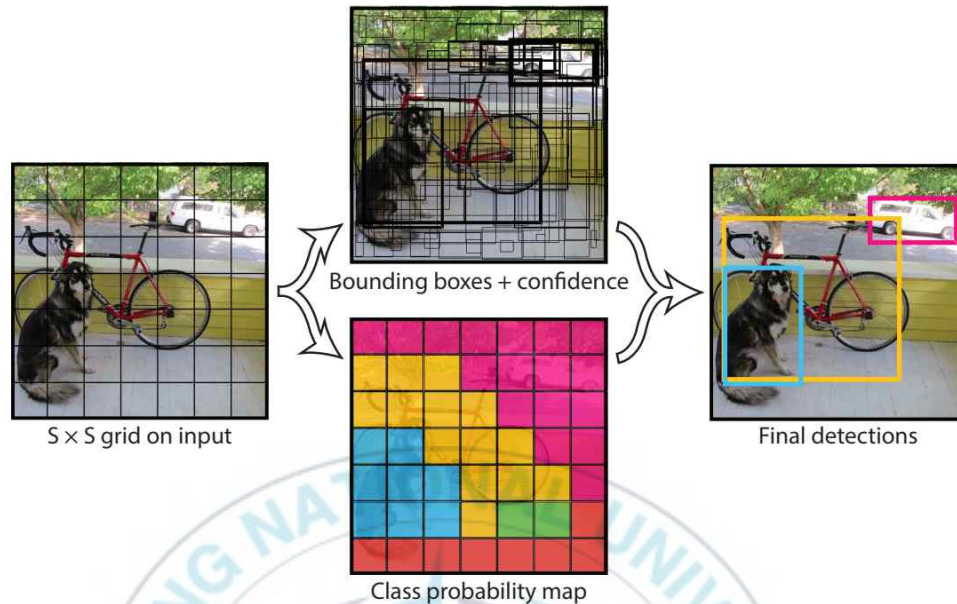


그림 3. YOLO 모델의 객체탐지 과정[11]

YOLO 모델의 객체탐지 과정은 그림 3과 같다. 먼저 입력받은 이미지를 $S \times S$ 그리드로 분할한다. 각 그리드 셀마다 물체가 있을 것으로 생각되는 경계상자(bounding box)를 미리 정해둔 개수만큼 예측한다. 각각의 경계 상자는 중점의 좌표와, 너비, 높이에 의해 정의된다. 또한 각각의 경계 상자에 대해서 물체가 있을 확률(confidence)을 계산한다. 마지막으로 각각의 그리드 셀마다 경계 상자의 물체가 특정 물체일 확률을 계산한다[11]. YOLO는 이렇게 객체의 경계상자의 탐지와 객체의 분류를 동시에 처리하는(unified detection) 특징을 가진다. 경계상자 탐지와 분류를 따로 처리하는 2단계 모델에 비해서 빠른 처리 속도를 가지며, 이런 점에서 스마트폰과 같은 모바일 장치에 적합한 면이 있다.

2.3. 전이 학습 (transfer learning)

전이 학습은 데이터가 부족한 상태에서 딥러닝 모델을 구축하기 위해 데이터가 풍부한 상태에서 훈련된 모델을 재사용하는 것을 의미한다. YOLOv5 모델은 COCO 2017 데이터 셋에서 훈련시킨 사전학습(pre-trained)된 모델을 제공하고 있다. COCO 2017 데이터 셋 80개의 클래스로 분류된 총 164,000장의 이미지로 구성되며, 학습(training), 검증(validation), 테스트(test) 데이터로 나누어져 있다[12]. 객체 인식 딥러닝 모델을 학습시키기 위해 필요한 이미지 데이터와 경계 상자 데이터를 제공한다. 본 연구에서는 사전 학습된 모델인 YOLOv5m를 음식 이미지 데이터를 추가 학습하여 사용하였다.

2.4. 음식영양정보

음식의 영양정보는 공공데이터포털이 제공하는 식품의약품안전처의 식품 영양성분 정보를 사용하였다. 음식 영양정보 데이터베이스로는 원재료성 식품 DB, 가공식품 DB, 음식 DB 등이 있다. 본 연구에서는 통합식품 DB인 식품의약품안전처_식품 영양성분 정보를 사용하였다[13].

3. 스마트폰을 활용한 음식인식 및 칼로리 추정

건강한 삶을 위해서는 영양과 칼로리의 균형 잡힌 식사 관리가 필요하다. 비만으로부터 오는 각종 질환을 예방하기 위해서는 음식의 칼로리를 모니터링할 필요가 있다. 본 논문은 사용자가 섭취한 음식을 간편하게 모니터링을 하기 위해 음식인식 및 칼로리를 추정하는 방법을 제안하고, 스마트폰의 앱을 통해 음식 영양 정보를 제공하고자 한다.

제안하는 시스템은 1장의 식단 사진을 분석하여 식단에 포함된 각각의 음식의 종류를 파악하고 음식에 대한 칼로리를 추정한다. 참조 객체가 존재하지 않을 경우 식품의약품안전처_식품 영양성분 정보에서 음식의 1회 제공량과 음식의 1회 제공량에 대한 칼로리와 영양정보 정보를 기반으로 칼로리를 추정하며, 조금 더 정확한 음식의 칼로리 추정을 위해서는 크기가 알려져 있는 참조 객체를 이용해서 참조 객체의 상대 크기를 사용하여 음식의 양을 추정하고, 이를 통해 좀 더 정확한 칼로리를 추정하는 기능을 제공한다.

3.1절에서는 음식인식을 위한 모델 개발 과정인 데이터 수집 및 학습 과정에 대하여 서술한다. 3.2절에서는 인식된 음식의 칼로리 추정 방법에 대하여 서술한다. 3.3절에서는 인식된 음식에 대한 영양정보의 정보 취득방법에 대하여 서술한다.

3.1. YOLOv5를 이용한 음식인식 모델 개발

본 논문은 사용자가 음식 사진을 앱에 업로드하면, 딥러닝 모델을 통해 음식 인식 및 인식된 음식의 영양정보를 제공하는 시스템을 제안하였다. 그림 4는 본 논문에서 제안한 스마트폰을 활용한 음식 인식 및 칼로리 추정 시스템을 도식화하였다.

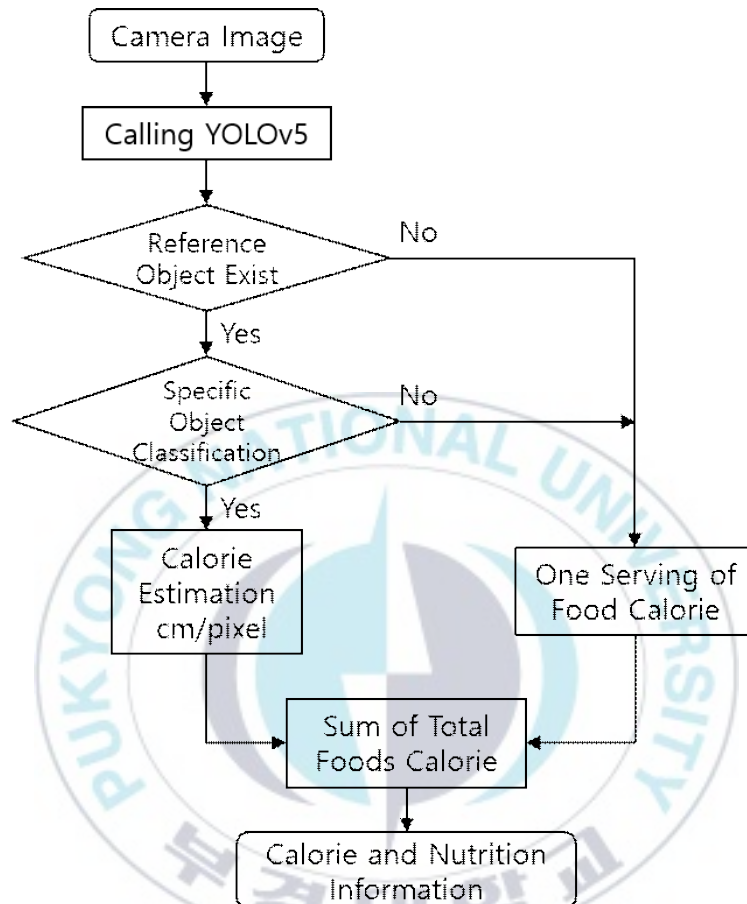


그림 4. 스마트폰을 활용한 음식 인식 및 칼로리 추정 알고리즘

이용자가 음식 영양 정보를 알고자 하는 음식 이미지를 스마트폰 카메라로 찍거나, 스마트폰의 갤러리에서 가져온다. 가져온 이미지를 미리 학습시킨 딥러닝 모델에 입력한다. YOLOv5 딥러닝 모델을 통해 예측한 객체 인식 데이터와 상대 위치 데이터 값을 받는다. 받은 데이터 값에서 참조 객체의 유무를 확인한다. 참조 객체가 존재하면 참조 객체를 기준으로 인식된 객체의 실제 길이를 계산한다. 실제 길이를 계산 후

크기 비율에 따라 칼로리를 추정한다. 참조 객체가 존재하지 않거나 특정 클래스로 분류되지 않은 경우 공공데이터포털의 식품의약품안전처_식품 영양성분 정보에서 음식의 1회 제공량에 해당하는 칼로리를 받아 음식의 총 칼로리를 계산한다. 음식의 총 칼로리를 계산한 후 식품에 대한 영양성분 정보 데이터와 칼로리 총합 데이터를 제공한다.

3.1.1. 데이터 수집

음식인식 및 칼로리 추정을 위한 딥러닝 모델을 학습시키기 위해서는 음식 이미지 데이터의 수집이 필요하다. 본 연구의 음식 인식 및 칼로리를 추정하는 앱에서 한국 사람들의 섭취 음식을 모니터링하고, 영양성분을 제공하기 위해서 한국 사람들이 즐겨 먹는 음식 이미지 데이터를 수집하였다.

음식 이미지 데이터는 AI-Hub에서 제공하는 한국 이미지(음식) 데이터를 사용하였다. 데이터는 한식재단의 음식 분류 및 한국인이 즐겨 먹는 음식 통계를 참조하여 선정된 150종의 음식으로 구성되어 있고[14], 각 클래스별 1,000장씩 총 150,000장의 음식 이미지로 구성된다. 표 1과 그림 5는 한국 이미지(음식) 데이터의 예시이다.

표 1. 한국 이미지(음식) 데이터 종류 예시

대분류	소분류
구이	갈비구이, 갈치구이, 고등어구이, 곱창구이
김치	갯김치, 깍두기, 배추김치, 총각김치
국	떡국, 미역국
나물	고사리나물, 시금치나물
밥	김밥, 김치볶음밥, 비빔밥



그림 5. 한국 음식 이미지 예시

3.1.2. 데이터 라벨링

YOLOv5 모델을 학습시키기 위해서는 이미지 데이터와 이미지에 대한 라벨링 데이터가 담긴 주석(annotation) 파일이 필요하다. 이미지에 대한 라벨링 데이터는 객체의 클래스 데이터와 경계 상자(Bounding Box) 데이터로 구성되어 있다. 본 연구에서는 그림 6의 labeling 프로그램 램을 통해 이미지 데이터를 라벨링 하였다.



그림 6. labeling 프로그램

데이터 라벨링을 위해서는 먼저 클래스를 정의해 주어야 한다. 그 후 그림 6처럼 경계 상자의 영역을 설정한다. 경계 상자는 이미지에서 하나의 객체 전체를 포함하는 가장 작은 직사각형을 의미한다. 본 연구에서는 음식 크기에 비례하여 음식의 칼로리를 추정하기 때문에 그릇이 아닌 음식이 존재하는 영역에 맞게 경계 상자를 설정하였다. 경계 상자

를 설정한 영역의 클래스를 지정하면 그림 7과 같이 라벨링 한 데이터가 담긴 주석 파일을 얻을 수 있다.



그림 7. 라벨링 데이터 예시

그림 7의 라벨링 데이터의 예시를 보면 객체 별 1개의 열로 이루어져 있으며, 5개의 인자로 구성되어 있다. 5개의 인자는 순서대로 객체의 클래스, 경계 상자의 중점의 x 좌표, 중점의 y 좌표, 너비, 높이를 표시한다. 만약 그림 6처럼 하나의 이미지에 클래스가 다른 2개의 객체가 존재한다면, 그림 7과 같이 2개의 열이 생성된다.

본 연구에서는 위의 라벨링 도구(labelimg)를 사용하여 생성한 데이터와 그림 8과 같이 Roboflow에서 제공하는 라벨링 된 데이터를 사용하였다.

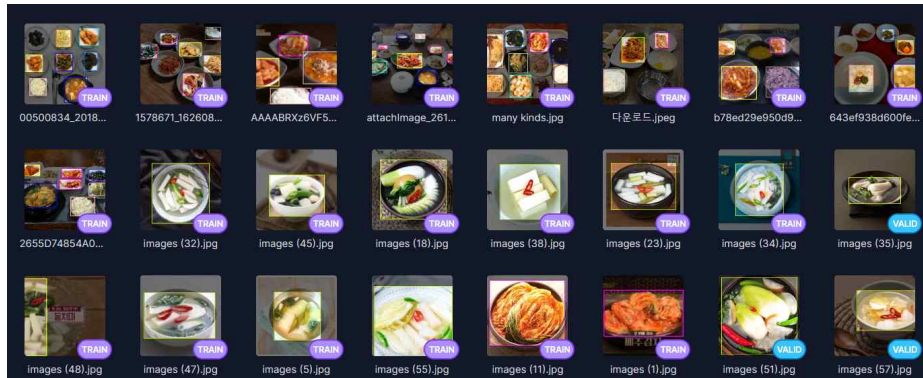


그림 8. Roboflow 라벨링 된 데이터

음식은 총 111 클래스로 나누었으며, 라벨링 된 데이터 예시는 표 2와 같다.

표 2. 라벨링 데이터 예시

Class number	Class name
0	쌀밥
1	잡곡밥
2	현미밥
3	카레라이스
...	
108	메추리알
109	메밀전병
110	모래집튀김

3.1.3. 데이터 증강

데이터의 양이 많을수록 정확도가 높은 딥러닝 모델을 구축할 수 있다. 대규모 데이터 셋은 모델이 다양한 패턴과 특징을 학습할 수 있도록 도와준다. 따라서 데이터의 양을 최대한 확보하여 모델을 훈련시키는 것이 중요하다. 본 연구에서는 Roboflow의 데이터 증강툴을 사용하여 데이터를 증강하였다. 데이터 증강은 모델 성능을 향상시키며, 데이터의 양이 적거나 클래스별 데이터의 양이 불균형한 경우 사용한다.

앱 사용자가 촬영한 음식 사진은 채도와 방향이 다양할 수 있다. 다양한 환경에서 찍힌 음식 사진을 인식하고, 적은 데이터를 보완하기 위해 데이터를 증강하였다. 데이터 증강으로 이미지의 채도 조절(saturation)과 회전(rotate)을 하였다. 채도 조절은 이미지의 색감을 변화시키는 방법이다. 이 기법을 사용하여 딥러닝 모델이 다양한 조건에서 음식을 인식할 수 있도록 한다. 본 연구에서는 이미지의 채도를 50% 증가시키고, 50% 감소시켜 이미지를 증강하였다. 이미지 회전은 이미지의 방향을 바꾸는 방법이다. 본 연구에서는 시계 방향과 반시계 방향으로 90도 회전시키고, 원본 이미지에서 상하로 뒤집어 이미지를 증강하였다. 그림 9는 이미지 데이터의 채도를 조절하고 방향을 회전시킨 예시이다.

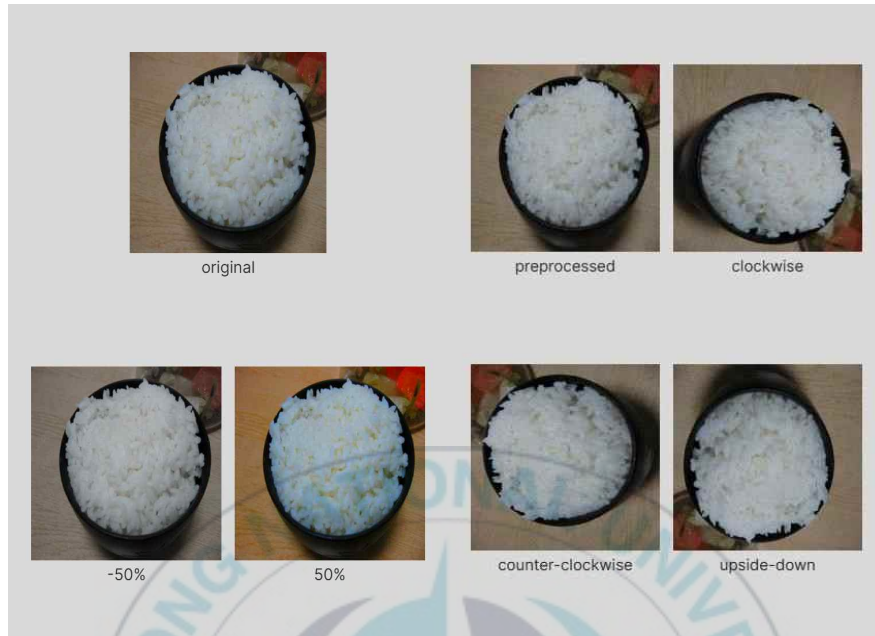


그림 9. 데이터 증강 예시

본 연구에서는 111 클래스, 각 클래스별 350장 55,919장의 음식 사진 데이터를 활용하여 딥러닝 모델을 학습시켰다. 표 3은 본 연구에서 딥러닝 모델을 학습시킬 때 사용한 모든 음식의 종류 및 개수이다.

표 3. 데이터 종류 및 개수

num	name	count	num	name	count	num	name	count
0	쌀밥	3620	37	나박김치	103	74	소시지볶음	1731
1	잡곡밥	509	38	냉면	73	75	시루떡	517
2	현미밥	109	39	갯잎장아찌	543	76	숙주나물	599
3	카레라이스	179	40	족발	1059	77	송편	514
4	김밥	1000	41	삼겹살	1662	78	계란후라이	926
5	김치볶음밥	800	42	깍두기	577	79	감자전	583
6	김치	941	43	멸치볶음	492	80	계란찜	965
7	연근조림	965	44	삼각김밥	122	81	순두부찌개	1013
8	동치미	36	45	팔죽	455	82	채소튀김	462
9	잡채	2023	46	도라지무침	66	83	골뱅이무침	485
10	김치찌개	237	47	고사리	619	84	취포튀김	482
11	갈비	32	48	호박무침	51	85	깨강정	508
12	육회	56	49	미역국	1556	86	야채죽	508
13	콩나물	673	50	김	22	87	꿀떡	537
14	우동	63	51	조개국	40	88	미꾸라지튀김	524
15	백김치	61	52	육개장	885	89	백설기	533
16	잔치국수	1061	53	시금치나물	84	90	원봉	533
17	콩나물국밥	58	54	고등어조림	50	91	닭꼬치	429
18	곰탕	32	55	멸치조림	68	92	닭튀김	560
19	도라지나물	57	56	열무김치	78	93	달걀말이	596
20	비빔밥	1510	57	짬뽕	120	94	단무지	488
21	어묵탕	57	58	라면	715	95	한과	514
22	부대찌개	87	59	양념치킨	146	96	더덕구이	520
23	불고기	198	60	된장찌개	754	97	돈까스	523
24	총각김치	748	61	감자탕	151	98	동그랑땡	453
25	오이김치	91	62	멸치볶음	492	99	어묵볶음	527
26	무말랭이	768	63	짜장면	818	100	감자튀김	416
27	북엇국	56	64	칼국수	76	101	간장게장	457
28	장어	56	65	만두	601	102	가래떡	477
29	볶음밥	53	66	과전	1493	103	김말이튀김	488
30	파김치	68	67	삼계탕	201	104	김무침	495
31	고등어구이	200	68	순대국밥	200	105	훈제오리	553
32	장조림	75	69	떡볶이	716	106	인절미	520
33	제육볶음	820	70	떡국	76	107	마늘쫀장아찌	559
34	메밀소바	35	71	생연어	514	108	메추리알	796
35	된장국	66	72	호박전	1253	109	메밀전병	511
36	무생채	427	73	미역줄기	837	110	모래집튀김	540

3.1.4. 딥러닝 모델 학습 방법

본 연구에서는 YOLOv5 딥러닝 모델을 통해 커스텀 데이터를 학습시켰다. 수많은 이미지 인식 딥러닝 모델 중 YOLOv5 모델을 선택한 이유는 YOLO 모델은 다중 객체 인식과 인식된 객체의 상대 위치에 대한 결과를 알 수 있으며 또한 다른 객체탐지 모델인 R-CNN 계열의 모델에 비해 가볍고, 인식 속도가 빠르다는 장점을 가지고 있기 때문이다.

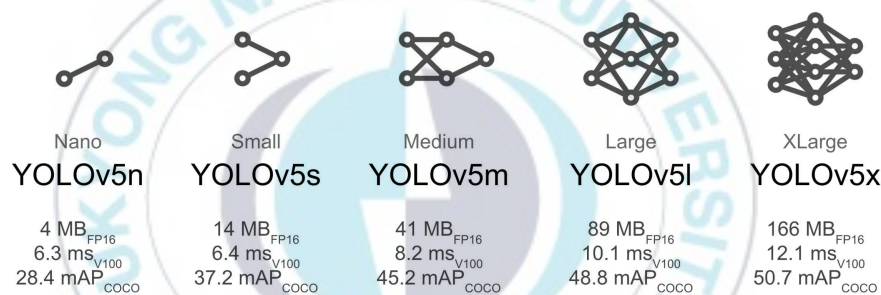


그림 10. YOLOv5 모델 종류

YOLOv5 딥러닝 모델에는 네트워크의 크기에 따라 그림 9와 같이 여러 가지 버전이 제공된다. 모바일에서의 이미지 인식을 타겟으로 딥러닝 모델을 구축하였기 때문에 모델의 크기가 작으면서도 이미지 인식 성능이 좋은 모델을 사용해야 한다. 모바일에서 이미지 인식을 하기 위한 적당한 모델은 모델의 크기가 작은 YOLOv5n, YOLOv5s, YOLOv5m 모델이다. 본 연구에서는 모바일에 적합한 모델의 크기가 작으면서도 정확도가 높은 YOLOv5m 모델을 사용하였다.

3.2. 제안 시스템의 동작 과정

앱 사용자는 스마트폰을 통해 음식 사진을 찍거나 스마트폰의 갤러리에서 음식 사진을 가져오는 것 중 하나를 선택하여 사진을 가져온다. 가져온 사진은 앱의 객체 인식 딥러닝 모델을 통해 다중 음식 인식을 한다. 그 후 참조 객체가 존재하는 경우와 존재하지 않는 경우로 나눈다. 참조 객체가 존재하는 경우 상대 크기를 사용하여 음식의 크기를 추정하고 인식된 모든 음식에 대하여 음식 영양정보 데이터를 제공한다. 참조 객체가 존재하지 않는 경우 인식된 음식 영양정보의 1회 제공량에 대한 데이터를 제공하는 앱을 구현했다.

```
capture image
object(food class, region) list ← call YOLO
if reference object exists then
    calculate mm/pixel
foreach object in object list
    if reference object exists then
        calculate food amount and calorie
    else
        use default amount (average amount for a serve)
        consult public nutrition information using OPEN API
        show region, food class, calorie, nutrition information
show total calories
```

그림 11. 스마트폰을 활용한 음식 인식 및 칼로리 추정 의사코드

그림 11은 스마트폰을 활용한 음식 인식 및 칼로리 추정 시스템의 의사코드(Pseudo-code)이다. 먼저 스마트폰에서 이미지를 가져온다. 커스텀 데이터를 학습시킨 YOLOv5 딥러닝 모델을 통해 음식의 클래스

값과 상대 위치를 리스트로 받는다. 만약 참조 객체가 존재한다면 참조 객체를 통해 음식의 실제 크기를 계산하고 음식의 칼로리를 추정한다. 그렇지 않다면 인식된 음식의 1회 제공량을 기준으로 칼로리를 계산한다. 음식의 영양정보는 공공데이터포털의 식품의약품안전처_식품 영양 성분 정보에서 OPEN API를 사용하여 음식의 영양정보를 받아 온다. 딥러닝 모델을 통해 예측한 음식의 종류, 위치정보와 API를 통해 받아 온 음식의 칼로리와 영양성분에 대한 정보를 앱의 인터페이스에 보여준다.

다음은 입력 사진에 식기들의 크기를 추정할 수 있는 참조 객체가 존재하는 경우와 그렇지 않은 경우를 구분하여, 용기의 크기와 그것에 담긴 음식의 양을 측정하는 과정에 대해 설명한다.

3.2.1. 참조 객체 기반 칼로리 추정 방법

본 연구는 한국 가정식에 대한 식품 영양성분을 모니터링하는 시스템을 제안하였다. 일반적인 한국 가정식은 밥류와 국물류가 식사 대부분의 칼로리를 차지한다고 가정한다.

YOLO 인식 모델은 인식된 객체들에 대해 이미지 클래스와 MBR(x , y , width, height) 정보를 제공한다. 여기서 이미지 클래스는 인식된 객체의 예측한 클래스를 의미한다. x 는 인식된 객체 중심의 x 좌표 값이고 y 는 인식된 객체 중심의 y 좌표 값이다. width는 인식된 객체의 너비이고 height는 인식된 객체의 높이이다. 이 값들을 통해 이미지 내 인식된 객체의 상대적인 위치를 알 수 있다.

식탁 위에 올려진 동전과 같이 그 크기가 알려진 객체가 인식되는 경우, 그것의 크기를 이용하여 식기 내에 담긴 음식의 양을 추정할 수 있

다. 이와 같이 그 크기가 사전에 알려진 객체를 참조 객체로 정의한다. 참조 객체는 일상생활에서 흔히 사용할 수 있는 동전, 명함, 신용카드 등을 활용할 수 있으며, 참조 객체를 인식하기 위한 별도의 학습과정이 필요하다.

표 4. 참조객체 및 그 크기

참조객체	크기(가로×세로)
100원 동전	24mm
명함	90×50mm
카드	85.60×53.98mm

음식의 칼로리를 추정하기 위해 참조 객체의 상대 크기를 활용하였다. 본 연구에서는 기준이 되는 참조 객체로 한국 100원 동전을 사용하였다. 100원 동전을 참조 객체로 사용한 이유는 대한민국 시중에서 가장 많이 유통되는 동전이며 크기가 고정되어 있기 때문이다. 100원 동전의 지름은 24.00mm이며 음식의 칼로리를 추정하기 위해 동전의 지름 값을 사용하였다. 그림 12는 커스텀 데이터를 학습시킨 YOLOv5 모델에 직접 찍은 음식 사진 이미지를 넣었을 때의 인식된 결과의 이미지 예시이다.

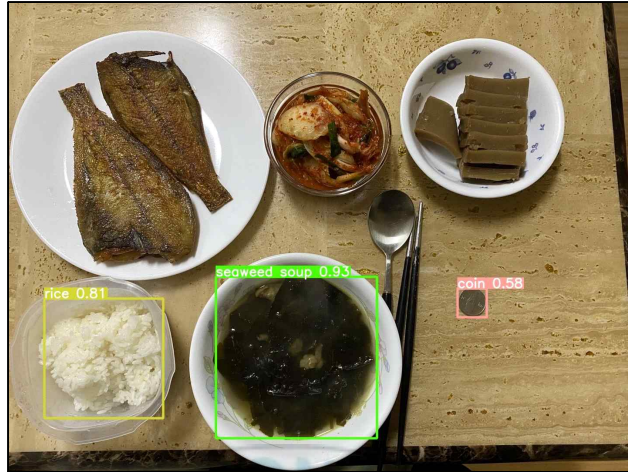


그림 12. 참조 객체가 존재하는 경우 이미지 예시

표 5. 그림 12.의 딥러닝 모델을 통한 예측 결과값

class	center_x	center_y	width	height
1	0.739931	0.639815	0.04652788	0.06111111
4	0.152431	0.75787	0.189583	0.25463
5	0.459722	0.756019	0.256944	0.343519

표 5는 딥러닝 모델의 결과 값이며 클래스를 분류해 주는 값과 상대적 위치 좌표를 제공한다. 다음은 딥러닝 모델을 통해 받은 결과값으로 참조 객체를 통해 음식의 칼로리를 추정하는 방법에 대하여 서술한다. 먼저 참조객체의 길이 L_R 과 이미지 상에서 참조객체의 픽셀수 P_R 및 음식 부분의 한 면의 픽셀수 P_F 를 이용하여 음식 객체의 너비 L_F 를 계산한다.

$$L_F = \frac{P_F \times L_R}{P_R}$$

추정된 음식 객체의 너비 L_F 를 이용하여 음식의 무게 W_F 를 다음과 같이 추정한다.

$$W_F = \left(\frac{L}{10} \right)^2 \times 200g$$

본 연구에서는 한국의 식당 밥그릇을 기준으로 음식의 무게를 추정하였다. 한국의 식당 밥그릇의 경우 직경 100mm, 높이 55mm을 기준으로 쌀밥이 약 200g 정도 담긴다. 식당 밥그릇의 너비 비로 음식의 무게를 추정하였기 때문에 음식의 실제 너비 값을 얻은 후 10을 나눈 후 제곱을 하여 음식의 무게를 추정하였다. 이제 다음과 같이 음식의 칼로리를 계산한다.

$$\frac{W_F}{S_WT} \times kcal = Food\ kcal$$

여기서 S_WT는 SERVING_WT의 약자로 식품의약품안전처_식품 영양성분 정보에서 OPEN API로 받아오는 식품별 1회 제공량을 의미한다. 식품의약품안전처_식품 영양성분 정보를 통해 1회 제공량 데이터와 칼로리 데이터를 받아오면 추정한 음식의 무게를 1회 제공량 데이터로 나누고 1회 제공량 칼로리로 곱하여 음식의 칼로리를 추정한다.

인식된 음식의 모든 칼로리를 합하고, 식품의약품안전처_식품 영양성분 정보에서 탄수화물, 단백질, 지방에 대한 정보를 받아온다. 그 후 인식된 모든 음식의 영역, 클래스 분류, 칼로리, 영양정보를 최종적으로 사용자에게 제공한다.

3.2.2. 참조 객체가 존재하지 않는 경우

참조 객체가 존재하지 않거나, 참조 객체가 있지만 특정 클래스로 분류되지 않은 경우, 본 연구에서는 공공데이터포털의 식품의약품안전처_식품 영양성분 정보의 1회 제공량 데이터를 활용하였다. 인식된 모든 음식의 1회 제공량 총 칼로리를 합산하고 추가적으로 탄수화물, 단백질, 지방과 같은 영양정보를 제공한다.

3.3. 인식된 음식에 대한 영양정보의 정보 취득

인식된 음식 데이터에 대한 영양성분 데이터를 제공하기 위해 공공데이터포털의 식품의약품안전처_식품 영양성분 정보를 사용하였다. 표 6은 제공받은 식품의약품안전처_식품 영양성분 정보를 요약하였다.

표 6. 음식영양정보 데이터

분류체계	보건-식품의약품안전
제공기관	식품의약품안전처
데이터포맷	JSON+XML
등록	2016-12-31
이용허락범위	이용허락범위 제한 없음

음식의 영양성분 정보는 REST API로 받으며 데이터 포맷은 JSON+XML으로 제공받는다. 그림 13은 공공데이터포털에서 식품 영양성분 정보를 받아 오는 데이터의 예시다.

```
{
  DESC_KOR: '김치찌개',
  SERVING_WT: '100',
  NUTR_CONT1: '61.00',
  NUTR_CONT2: '3.00',
  NUTR_CONT3: '3.80',
  NUTR_CONT4: '3.80',
  NUTR_CONT5: 'N/A',
  NUTR_CONT6: '491.00',
  NUTR_CONT7: 'N/A',
  NUTR_CONT8: 'N/A',
  NUTR_CONT9: 'N/A',
  BGN_YEAR: '2011',
  ANIMAL_PLANT: ''
},
```

그림 13. 음식영양정보 데이터 예시

식품 영양성분 데이터 정보는 식품 이름, 1회 제공량, 열량, 탄수화물, 지방, 당류, 콜레스테롤, 포화지방산 등 다양한 항목으로 구성되어 있다. 그중에서도 식품의 주요 영양성분인 탄수화물(g), 단백질(g), 지방(g) 데이터와 식품 이름, 1회 제공량(g), 열량(kcal), 나트륨(mg) 데이터만 받아 온다. 출력 결과 데이터의 예시는 표 7과 같다.

표 7. 음식영양정보 데이터 출력 결과 예시

항목명(국문)	항목명(영문)	샘플데이터
식품이름	DESC_KOR	김치찌개
1회제공량 (g)	SERVING_WT	100
열량 (kcal)	NUTR_CONT1	61.00
탄수화물 (g)	NUTR_CONT2	3.00
단백질 (g)	NUTR_CONT3	3.80
지방 (g)	NUTR_CONT4	3.80
당류 (g)	NUTR_CONT5	N/A
나트륨 (mg)	NUTR_CONT6	491.00
콜레스테롤(mg)	NUTR_CONT7	N/A
포화지방산(g)	NUTR_CONT8	N/A
트랜스지방산(g)	NUTR_CONT9	N/A
구축년도	BGN_YEAR	2011
가공업체	ANIMAL_PLANT	

식품의약품안전처_식품 영양성분 정보를 받아오는 방법은 다음과 같다. 먼저 딥러닝 모델을 통해 인식한 음식 클래스 값을 한국어로 변환한다. 예를 들어 딥러닝 모델을 통해 ‘10’ 이라는 클래스 데이터를 받아 오면 10에 대응되는 식품인 ‘김치찌개’ 로 바꾸고, 식품의약품안전처_식품 영양성분 정보 REST API에서 김치찌개에 대응하는 식품 영양성분 정보를 받아 온다. 만약 식품 영양성분 데이터 정보에서 한 식품에 대한 여러 개의 식품 영양성분 데이터가 있으면, 구축년도를 기준으로 최근에 업데이트된 식품 영양성분 데이터를 받아오도록 설정하였다.

4. 음식인식 및 칼로리 추정 앱 구현

본 장에서는 음식인식 및 칼로리 추정 앱 개발환경 및 구현에 대하여 서술하고, 앱에 대한 성능을 평가한다. 앱은 플러터(Flutter)를 통해 개발되었으며, 앱의 서버에서 이미지 인식 딥러닝 모델이 실행된다.

4.1절에서는 앱의 개발환경에 대하여 서술한다, 4.2절에서는 앱의 구현 화면에 대하여 서술한다. 4.3절에서는 앱의 성능평가에 대하여 서술한다.

4.1. 앱 개발환경

앱은 플러터를 사용하여 개발되었으며, 플러터는 단일 소스코드로 iOS와 안드로이드 등 다양한 플랫폼에서 실행 가능한 UI 프레임워크이다. 이미지 인식 딥러닝 모델은 우분투(Ubuntu) 20.04.5 환경에서 학습시켰다. 딥러닝 모델은 파이토치(PyTorch)를 통해 학습시켰다. 학습시킨 딥러닝 모델을 앱에서 사용하기 위해 텐서플로우(TensorFlow.js)를 사용하여 모델을 변환하였다. 변환된 딥러닝 모델은 앱의 서버에서 실행된다. 식품 영양성분 정보를 앱에서 제공하기 위해 공공데이터포털에서 제공하는 REST API를 활용하였다. 사용자가 앱 내에서 음식 사진을 업로드하면, 해당 이미지를 이용하여 음식을 인식하고 인식된 음식에 대한 영양성분 정보를 얻기 위해 REST API를 호출한다. 이를 통해 사용자는 음식 사진을 통해 인식된 음식에 대한 영양성분 정보를 앱 내에서 확인할 수 있다.

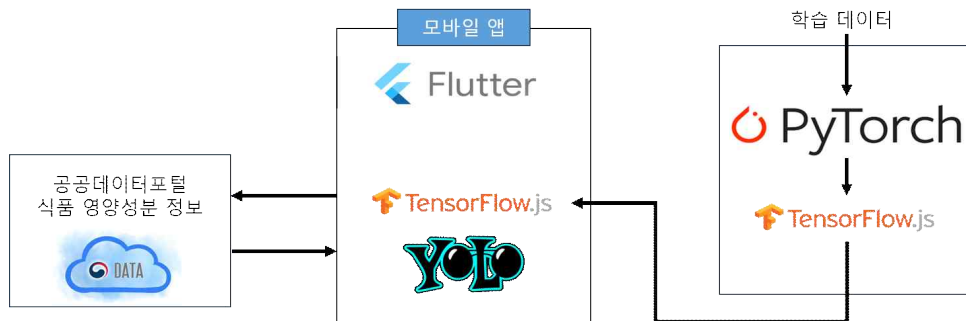


그림 14. 앱 개발환경

4.2. 앱 구현 화면

본 시스템은 스마트폰 앱으로 iOS 16 및 안드로이드 SDK 21 대상으로 개발되었다. 그림 15의 앱 초기화면을 보면 사진 촬영 버튼과 사진 선택 버튼이 있다. 사진 촬영 버튼을 누르면 실시간으로 음식 사진을 찍어 음식에 대한 영양정보를 제공받을 수 있다. 사진 선택 버튼을 누르면 스마트폰의 갤러리에서 음식 영양정보를 제공받고 싶은 사진을 선택하여 가져올 수 있다.



그림 15. 앱 초기 화면

위의 방법을 통해 화면 상단에 이미지를 가져온다. 그 후 미리 학습된 딥러닝 모델을 통해 음식을 인식하고, 참조 객체가 있을 경우 공공 데이터포털에서 식품 영양성분 정보를 REST API로 받아 인식된 음식의 칼로리를 추정한다. 참조 객체가 없을 경우, 인식된 음식의 1회 제공량에 대한 정보를 받아 온다.



그림 16. 참조 객체가 존재하는 경우

그림 16의 식단정보 확인 결과는 참조 객체가 존재하는 경우이다. 사진 촬영이나 사진 선택을 통해 가져온 음식의 양을 추정하고 인식된 음식의 총 칼로리의 합과 영양정보를 제공한다. 인식된 각각의 음식은 객체인식 딥러닝 모델이 몇 퍼센트의 확률로 음식을 예측하였는지 알 수 있다.

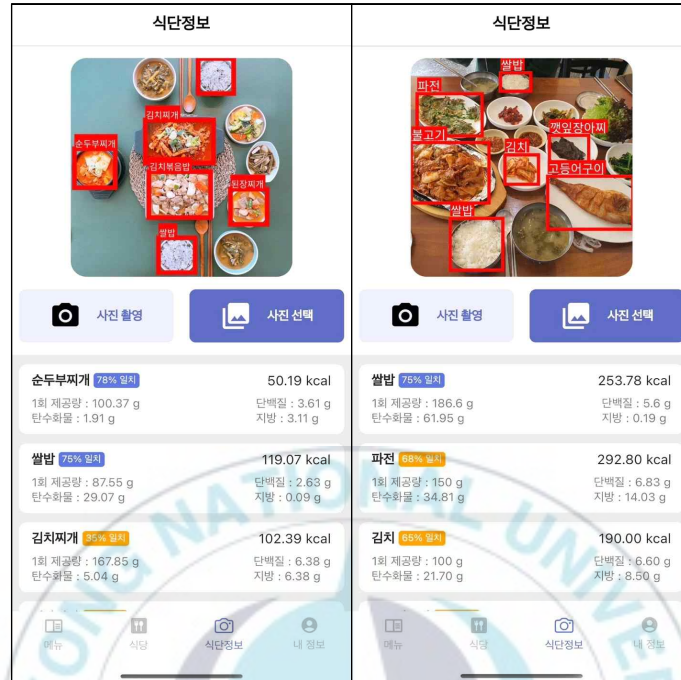


그림 17. 참조 객체가 존재하지 않는 경우

위의 그림 17의 식단정보 확인 결과는 참조 객체가 존재하지 않는 경우이다. 참조 객체가 존재하는 경우와 달리 인식된 음식의 1회 제공량을 공공데이터포털 영양성분 정보에서 받아 온다. 참조 객체가 존재하는 경우와 같이 인식된 모든 음식의 총 칼로리와 영양정보를 제공한다.

4.3. 성능평가

본 논문은 YOLOv5 객체 인식 딥러닝 모델을 학습시켜 음식인식 및 칼로리를 추정하는 기법을 앱을 통해 제안하였다. 본 절에서는 음식인식 및 칼로리를 추정하는 기법에 대한 성능을 평가한다. 성능 평가로 정성적 비교 평가와 정량적 비교 평가를 진행하였다. 앱의 식단정보 성능에 대해 정성적 비교평가를 하였고, 앱에서 사용한 딥러닝 모델에 대

해 정량적 비교 평가를 하였다.

4.3.1. 정성적 비교 평가

정성적 비교 평가로 다중 음식 인식을 하는지, 음식의 영양성분을 제공하는지, 음식의 칼로리 정보를 제공하는지에 대하여 평가하였다.

표 8. 정성적 비교 평가

	Liu, M.Chun	V.B.Kasyap, B.Alshujaa	다이어트 앱	제안한 시스템
다중 음식 인식	×	○	×	○
영양성분 제공	×	×	○	○
칼로리 정보 제공	×	○	△	○

표 8은 본 연구와 다른 연구에 대해서 정성적으로 비교 평가한 표이다. Liu[5], M.Chun[6]의 연구에서는 한 가지 음식이 있는 사진에 대한 딥러닝 모델을 설계하여 한 가지 음식에 대한 음식 분류를 한다. 다중 음식 인식, 영양성분, 칼로리에 대한 정보는 제공하지 않는다. V.B. Kasyap[7], B. Alshujaa[9]의 연구는 다중 음식 인식과 칼로리를 제공하지만 영양성분 정보는 제공하지 않는다. 앱스토어에 있는 다이어트 앱은 섭취한 음식과 섭취량을 사용자가 직접 기록하고 섭취량에 따라 음식의 영양성분과 칼로리를 계산해 정보를 제공한다. 제안한 시스템에서는 다중 객체 인식 딥러닝 모델을 통해 다중 음식 인식을 하고, 인식한 음식에 대한 영양성분 데이터를 제공하며, 참조 객체의 유무에 따라서 칼로리 데이터를 제공한다.

4.3.2. 정량적 비교 평가

커스텀 데이터로 전이 학습시킨 YOLOv5m 모델의 성능을 평가한다. 약 50,000장의 이미지 데이터를 Train : Validation : Test = 8 : 1 : 1의 비율로 분리했으며 이미지 크기는 320, 배치크기(batch size)는 128로 하였으며, 200 에포크동안 YOLOv5m 모델을 학습시켰다.

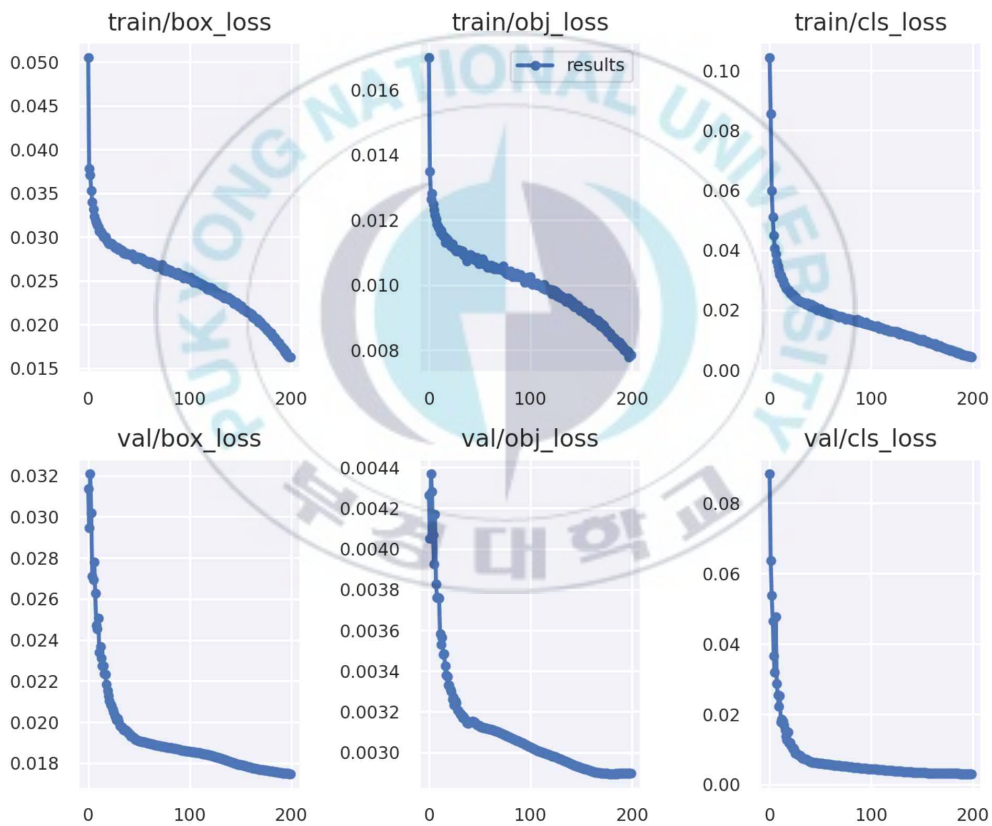


그림 18. YOLOv5 모델의 학습과정 및 성능지표 1

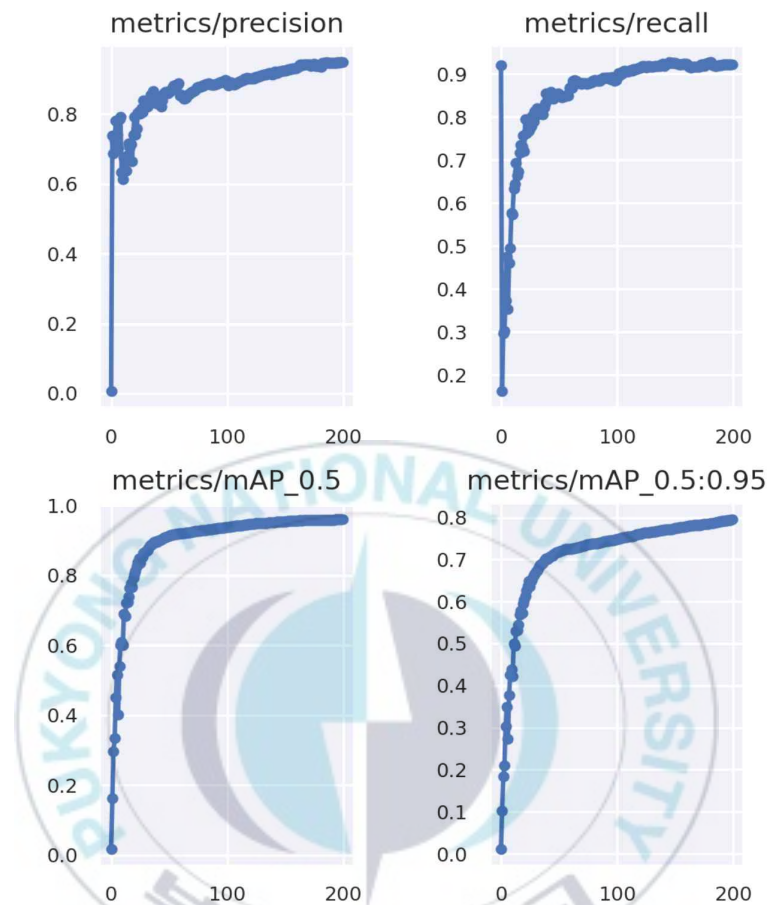


그림 19. YOLOv5 모델의 학습과정 및 성능지표 2

그림 18과 그림 19는 커스텀 데이터로 YOLOv5m 모델을 학습하는 과정과 성능지표를 보여준다. 각각의 성능지표에 대한 설명은 표 9에서 한다.

표 9. 모델의 성능 평가 지표 설명

지표	의미
box_loss	예측한 객체의 Bounding_box와 실제 객체의 Bounding_box의 오차
obj_loss	예측한 객체와 실제 객체의 오차
class_loss	예측한 객체의 Class와 실제 객체의 Class에 대한 오차
metrics/precision	모델의 정밀도
metrics/recall	모델의 재현율
metrics/mAP_0.5	IOU값이 50% 이상인 평균 정밀도
metrics/mAP_0.5:0.95	IOU값이 50%~95% 사이의 평균 정밀도

Precision은 정밀도를 의미하며 인식한 모든 결과들 중 옳게 인식한 비율을 의미한다.

$$Precision = \frac{TP}{TP+FP}$$

TP는 True Positive의 약자로 실제 값과 예측값이 True인 것을 의미한다. FP는 False Positive의 약자로 실제 값은 False이지만 예측값은 True인 것을 의미한다. 정밀도의 계산 방법은 옳게 인식한 값을 인식한 모든 결과값으로 나눈 값이다.

Recall은 재현율을 의미하며 실제 값들 중 예측한 결과가 올바르게 인식한 값들의 비율을 의미한다.

$$Recall = \frac{TP}{TP+FN}$$

FN은 False Negative 약자로 실제 값은 True이나 예측은 False인 것을 의미한다. 재현율의 계산 방법은 옳게 인식한 값을 실제 값이 Positive인 모든 값으로 나눈 값이다.

정밀도는 예측한 값이 얼마나 정확한지 판단하는 방법이고, 재현율을 실제로 정답을 얼마나 맞췄는지 판단하는 방법이다.

표 10. 모델의 성능 평가 지표

P	R	mAP50	mAP50-95
93.3%	92.5%	95.5%	78.7%

표 10은 본 연구에서 사용한 YOLOv5m 모델의 성능 지표이다. 재현율의 변화에 따른 정밀도를 나타낸 곡선을 PR 곡선(Precision-Recall Curve)이라고 한다. AP는 Average Precision의 약자로 Precision-Recall Curve의 아래 면적을 의미한다. 그림 20은 본 연구에서 사용한 YOLOv5m 모델의 PR 곡선이다.

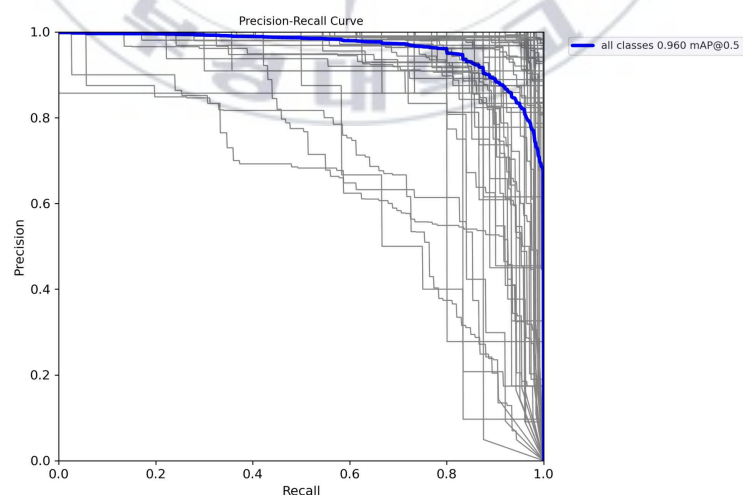


그림 20. Precision-Recall Curve

mAP(mean Average Precision)는 객체 검출 딥러닝 모델의 성능을 평가할 때 사용하는 지표이며 클래스별 AP의 평균값을 의미한다. IoU는 딥러닝 모델이 예측한 경계 상자와 실제 경계 상자의 겹치는 부분을 의미한다. IoU 값이 높을수록 객체 인식 딥러닝 모델이 객체의 경계 상자를 정확하게 예측했다고 볼 수 있다. mAP0.5는 객체 검출 딥러닝 모델의 정확성을 측정하는 데 사용되며, IoU의 임계값(Threshold)이 50% 이상에서의 평균 정밀도를 나타낸다. mAP0.5-0.95는 0.05 단계의 크기로 0.5부터 0.95까지의 IoU에 대한 평균 AP 값이다.

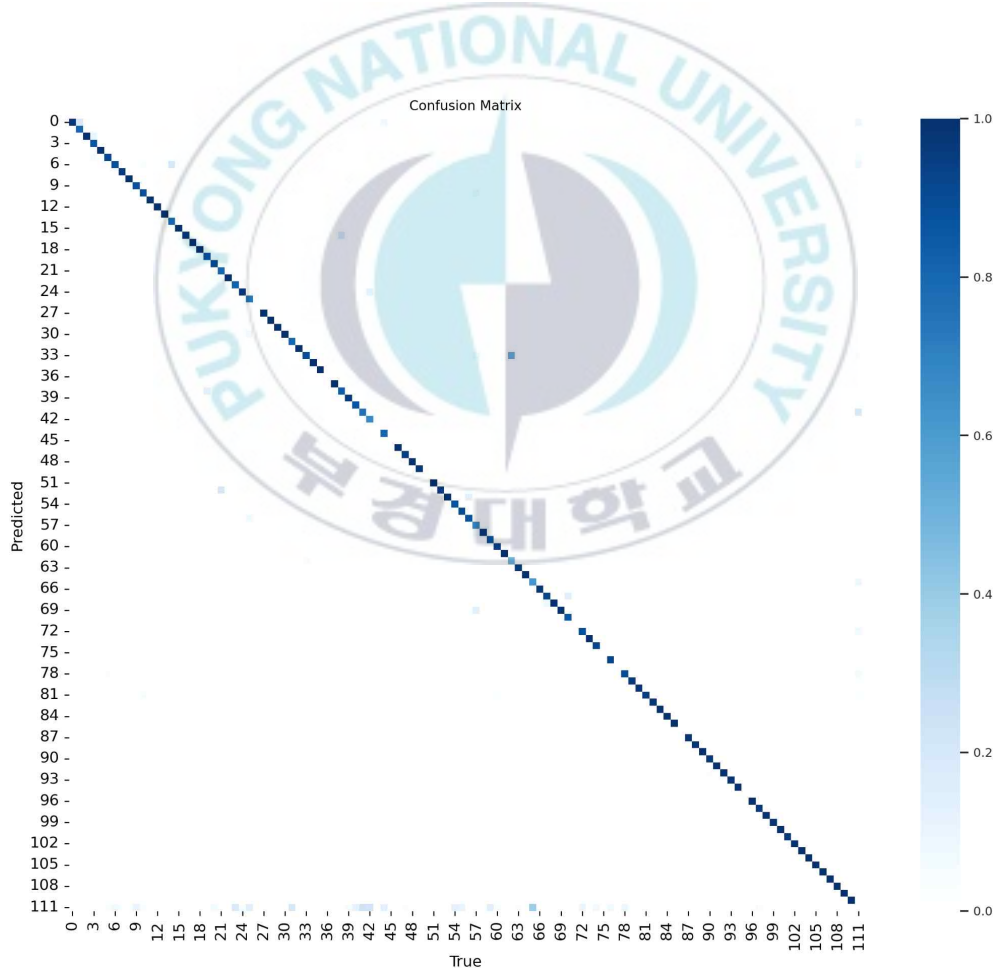


그림 21. Confusion matrix

그림 21은 본 연구에서 학습시킨 YOLOv5m 모델의 오차행렬(Confusion matrix)이다. 오차행렬은 학습된 딥러닝 모델의 예측 성능을 측정하기 위해 예측값과 실제 값을 비교하여 시각화한 표이다. 그림20의 y축은 딥러닝 모델의 예측값을 의미하고, x축은 실제 값을 의미한다. 그림에 표시된 사각형의 색깔이 진할수록 높은 빈도를 의미하고 사각형의 색깔이 연할수록 낮은 빈도를 한다.

4.3.3. 한계점

본 연구에 사용된 YOLOv5 딥러닝 모델 같은 경우 각 클래스 별 최소 1,000장 이상의 데이터를 학습시키는 것을 권장한다. 하지만 본 연구는 음식 이미지 데이터의 부족으로 각 클래스 별 평균 350장 정도의 데이터를 학습시켰다. 그 결과 그림 22와 같이 과전을 가자미구이로 분류하거나 총각김치를 김치로 잘못 분류하는 경우가 있다. 그림 21의 오차행렬의 결과를 보면 냉면을 우동으로 잘못 분류하는 것을 볼 수 있다.

그 이유는 두 가지로 정리할 수 있는데, 첫 번째는 이미지가 100개 이하인 클래스의 경우 딥러닝 모델이 잘못 예측할 확률이 높다. 그리고 음식의 색깔이 비슷한 경우이다. 학습 데이터 개수가 100개 이상임에도 불구하고 찜뽕을 김치찌개로 분류하거나 각두기를 총각김치로 잘못 분류하는 등의 경우가 있다. 이미지 인식 딥러닝 모델은 이미지의 색깔 요소도 학습하기 때문에 비슷한 색깔을 가진 음식은 구분을 잘 못하는 경우도 있다.



그림 22. 딥러닝 모델의 예측 결과

둘째로는 한국인의 가정식을 분류하는 모델을 만들었지만, 분류하는 클래스의 수가 제한적이다. 그림 22를 보면 가정식이지만 분류를 하지 못한 객체들도 있다. 한계점을 개선하기 위해서는 클래스의 가짓수와 이미지 데이터를 추가할 필요가 있다.

본 연구에서는 한국 가정식을 타깃으로 칼로리를 추정하는 방법을 제안하였다. 주로 국류나 밥류가 식단의 칼로리를 차지한다고 가정하여 반찬류는 1회 제공량에 대한 영양정보만 제공하였다. 김치, 고사리, 김 등의 반찬류는 칼로리를 크게 고려해 볼 필요는 없지만 제육볶음, 불고기, 갈비와 같은 반찬은 칼로리와 영양정보를 고려해 볼 필요가 있다. 하지만 본 연구에서는 반찬류는 1회 제공량으로 영양성분 정보를 제공하는 한계점이 있다.

5. 결론

본 논문은 딥러닝 기반 객체탐지 분류 모델을 활용하여 음식 사진에서 객체를 탐지 및 분류하고, 기준 물체를 활용하여 대략적인 음식의 열량을 추정하고 영양정보 데이터를 제공하는 기법을 제안하였다. 추정된 음식의 크기와 분류 결과를 기반으로 음식의 대략적인 열량을 추정하고, 해당 음식의 영양정보 데이터를 제공한다. 이를 통해 사용자는 자신이 섭취한 음식의 열량과 영양소를 모니터링할 수 있다.

본 논문은 **한식 위주의 가정식을 목표로 한 것이므로** 인식하는 음식의 종류가 제한적이다. 추후 연구에서는 중식, 일식, 양식을 포함하는 더욱 다양한 음식 데이터를 학습시키고, **3차원** 딥러닝 모델을 응용하여 정확도를 높인 음식 인식 및 열량 추정에 대한 연구를 진행할 예정이다.

참고 문헌

- [1] "필립스, '아시아 국가 개인건강관리 실태 조사' 결과 발표." Medcom. 2022년07월25일 수정, 2023년03월20일 접속, <http://mdon.co.kr/mobile/article.html?no=32829>
- [2] "비만의 진단과 평가." 대한비만학회. n.d. 수정, 2023년03월23일 접속, <http://general.kosso.or.kr/html/?pmode=obesityDiagnosis>
- [3] Yusuf, Salim, "Obesity and the risk of myocardial infarction in 27 000 participants from 52 countries: a case-control study.", The Lancet 366.9497, p.1640-1649, 2005
- [4] Ma, Y., Yang, Y., Wang, F., Zhang, P., Shi, C., Zou, Y., & Qin, H., "Obesity and risk of colorectal cancer: a systematic review of prospective studies." , PloS one, 8(1), e53916, 2013
- [5] Liu, Chang, "Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment." , ICOST, Proceedings 14. Springer International Publishing, p. 37-48, 2016
- [6] M. Chun, H. Jeong, H. Lee, T. Yoo and H. Jung, "Development of Korean Food Image Classification Model Using Public Food Image Dataset and Deep Learning Methods", IEEE Access, vol. 10, p. 128732-128741, 2022
- [7] V. B. Kasyap and N. Jayapandian, "Food Calorie Estimation using Convolutional Neural Network", 3rd International Conference on Signal Processing and Communication, ICPSC, p. 666-670, 2021

- [8] Sutedja, Indrajani, Raymond Bahana, and Ida Bagus Kerthyayana Manuaba. "Foods diary mobile application for diabetics." International Conference on Information Management and Technology, ICIMTech, IEEE, p.228–232, 2020
- [9] B. Alshujaa, F. AlNuaimi, W. Mansoor and S. Atalla, "Comparison of Food Calorie Measurement Using Image Processing and Machine Learning Techniques," International Conference on Signal Processing and Information Security ICSPIS, IEEE, p. 123–128, 2022
- [10] S. Naritomi and K. Yanai, "CalorieCaptorGlass: Food Calorie Estimation based on Actual Size using HoloLens and Deep Learning," IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VRW, IEEE, p. 818–819, 2020
- [11] Redmon, Joseph, "You only look once: Unified, real-time object detection.", Proceedings of the IEEE conference on computer vision and pattern recognition, p. 779–788, 2016
- [12] Lin, Tsung-Yi, "Microsoft coco: Common objects in context." Computer Vision-ECCV 2014: 13th European Conference, Part V 13. Springer International Publishing, p.740–755, 2014
- [13] "식품의약품안전처_식품 영양성분정보." 공공데이터포털. 2021년10월13일 수정, <https://www.data.go.kr/data/15057436/openapi.do>
- [14] "한국 이미지(음식)." AI-Hub. n.d. 수정, 2023년 03월 06일 접속, <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=79>

Thanks to

대학원에 입학한 게 엇그제 같은데 벌써 졸업을 앞두고 있습니다. 좋은 사람들을 많이 만난 덕분에 즐겁게 5학기를 보낼 수 있었습니다. 학부를 마치고 진로에 대한 많은 고민을 했었는데 대학원으로 진학하게 된 건 큰 행운이었던 것 같습니다. 이 논문이 나오기까지 수많은 사람들의 도움을 받았습니다. 도움이 없었다면 논문이 나올 수 없었을 것이기에 이 자리를 빌려 감사의 인사를 드리고자 합니다.

먼저 본 논문이 완성되기까지 세심한 지도와 많은 격려로 이끌어 주신 송하주 교수님께 진심으로 감사드립니다. 대학원에 있는 동안 아낌없는 조언과 가르침을 주신 덕분에 제가 한층 더 성장할 수 있었습니다. 논문을 작성하고 논문을 작성하는 과정에서 아낌없는 지도로 많은 가르침을 주신 권오흠 교수님, 김영봉 교수님께도 감사의 말씀드립니다.

본 연구에 많은 도움을 준 원준, 대학원 생활에 도움을 주신 정명환 선배님, 김종민 선배님, CHAN언니, 즐거운 대학원 생활을 할 수 있도록 도와준 광이, Tanvir, 동연, 성원, 보석, 윤수민 책임연구원님, 양희봉 책임연구원님 그리고 저의 친구들에게도 감사의 말씀드립니다.

마지막으로 저의 선택을 존중하고 지원해 주신 가족들에게도 감사의 인사를 드립니다. 저를 도와주신 많은 분들 덕분에 저는 무사히 석사 논문을 마무리할 수 있었습니다. 저에게 아낌없이 도움을 주신 분들께 진심으로 감사드립니다. 항상 좋은 일만 가득하시길 바랍니다.

2023년 7월

허지원