

<https://doi.org/10.7236/JIIBC.2023.23.5.83>
JIIBC 2023-5-13

영상처리 및 머신러닝 기술을 이용하는 운동 및 식단 보조 애플리케이션

Application for Workout and Diet Assistant using Image Processing and Machine Learning Skills

이치호*, 김동현*, 최승호*, 황인웅*, 한경숙**

Chi-Ho Lee*, Dong-Hyun Kim*, Seung-Ho Choi*,
In-Woong Hwang*, Kyung-Sook Han**

요약 본 논문에서는 홈 트레이닝 인구가 늘어나면서 증가한 운동과 식단 보조 서비스에 대한 수요를 충족시키기 위해 운동 및 식단 보조 애플리케이션을 개발하였다. 애플리케이션은 카메라를 통해 실시간으로 촬영되는 사용자의 운동 자세를 분석하여, 안내선과 음성을 이용해 올바른 자세를 유도하는 기능을 가진다. 또한, 사용자가 사진을 촬영하면 사진에 포함된 음식들을 분류하고 각 음식의 양을 추정하여, 칼로리 등의 영양 정보를 계산하여 제공하는 기능을 가진다. 영양 정보 계산은 외부의 서버에서 수행되도록 구성했다. 서버는 계산된 결과를 애플리케이션으로 전송하고, 애플리케이션은 결과를 받아 시각적으로 출력한다. 추가로, 운동 결과와 영양 정보는 날짜별로 저장하고 확인할 수 있도록 하였다.

Abstract In this paper, we developed a workout and diet assistance application to meet the growing demand for workout and dietary support services due to the increase in the home training population. The application analyzes the user's workout posture in real-time through the camera and guides the correct posture using guiding lines and voice feedback. It also classifies the foods included in the captured photos, estimates the amount of each food, and calculates and provides nutritional information such as calories. Nutritional information calculations are executed on the server, which then transmits the results back to the application. Once received, this data is presented visually to the user. Additionally, workout results and nutritional information are saved and organized by date for users to review.

Key Words : Diet Management, Home Workout, Mobile Application

1. 서론

COVID-19의 유행을 기점으로 홈 트레이닝 인구가 증가하였으며^[1], 여러 방송 매체들의 영향으로 운동에 대

한 사람들의 관심이 커졌다. 이러한 배경 속에서도 아직까지, 운동에 미숙한 초보자들 또는 기존부터 홈 트레이닝을 즐겨오던 사람들을 위한, 올바른 자세와 균형 잡힌 식단을 제시하고 보조해 주는 서비스 제공이 미흡하다.

*준회원, 한국공학대학교 컴퓨터공학부

**정회원, 한국공학대학교 컴퓨터공학부 (교신저자)

접수일자 2023년 9월 16일, 수정완료 2023년 9월 30일

계재확정일자 2023년 10월 6일

Received: 16 September, 2023 / Revised: 30 September, 2023 /

Accepted: 6 October, 2023

*Corresponding Author: khan@tukorea.ac.kr

Dept. of Computer Engineering, Tech University of Korea, Korea

현재 배포되고 있는 서비스들은 운동과 식단을 동시에 관리할 수 있는 기능을 제공하지 않으며, 서비스 중 다수가 한글을 지원하지 않거나 UI가 복잡하여 사용하기 어렵다는 단점을 가지고 있다.

본 논문에서는 운동 및 식단 분석 결과를 연계해 통합 결과를 제시하여 운동과 식단을 동시에 관리할 수 있도록 하며, 영상 및 사진 촬영으로 편리하게 보조 기능을 사용할 수 있도록 지원하는, 사용성과 편의성을 확보한 애플리케이션을 개발하였다.

II. 관련 연구

표 1은 본 논문의 애플리케이션(HomeFit)과 유사한 애플리케이션들의 주요 기능을 나타낸다. 하우핏에는 다양한 운동에 대해 올바른 자세를 유도하는 기능, 소비 칼로리를 계산하는 기능과 운동 진행 시 정확도에 따라 세부적으로 점수를 출력하는 기능이 있지만, 서비스가 유료로 제공된다는 단점이 존재한다^[2]. Fat Secret의 경우에도 식단 보조 기능을 사용할 시에 비용을 지불해야 한다는 단점이 있다^[3]. 두잉랩은 촬영 기반의 식단 분석 기능을 제공하며 모든 서비스가 무료이지만, 식단 정보가 저장되는 기간에 제약이 존재한다^[4]. 본 애플리케이션과 같이 운동 보조와 식단 보조 기능을 동시에 제공하는 애플리케이션은 없는 것으로 확인된다.

표 1. 유사 애플리케이션과의 비교
Table 1. Comparison with Similar Applications

기능 \ 제품	하우핏	두잉랩	Fat Secret	HomeFit
운동 보조				
자세 유도	○			○
소비 칼로리 계산	○	X	X	○
정확도 점수 출력	○			X
식단 보조				
촬영 기반 식단분석		○	X	○
데이터 장기간 저장	X	X	○	○
기타				
통합 통계 출력	X	X	X	○
전 기능 무료 제공	X	○	X	○

III. 세부 설계 및 구현

1. 개발 환경

본 논문의 애플리케이션과 서버는 Windows 10에서

개발하였으며, 애플리케이션은 Android Studio Dolphin | 2021.3.1.에서 JDK 1.8을 사용하였고^[5], 서버는 VSCode 1.79.2에서 Python 3.9.7을 사용하여 개발하였다^[6]. 음식 분류를 위한 AI 학습은 Google Colab과 macOS M1 Max 환경에서 수행하였고^[7], 테스트 환경은 Android Studio 에뮬레이터와 삼성전자 갤럭시 S20(Android 12)을 선택했다.

2. 시스템 구성

시스템은 그림 1과 같이 구성되어 있다. 애플리케이션은 크게 움직임 인식(운동 보조) 모듈과 식단 정보 관리(식단 보조) 모듈로 구성된다. 움직임 인식 모듈은 애플리케이션 내부에서 작동하며 사용자 디바이스에 내장된 카메라를 통해 사용자와 상호작용한다. 식단 정보 관리 모듈의 경우, 음식 분류와 양 추정을 위한 서버와 연결된다. 애플리케이션은 서버로 사진을 전송하고, 서버는 이를 분석해 다시 애플리케이션으로 전송하는 방식으로 구현하였다. 이렇게 두 모듈을 통해 도출된 운동 정보와 영양 정보들을 저장하고 관리하기 위한 DB는 애플리케이션 내부에 두어, 개발 속도와 정보 출력 과정에서의 효율성을 높이고자 하였다.

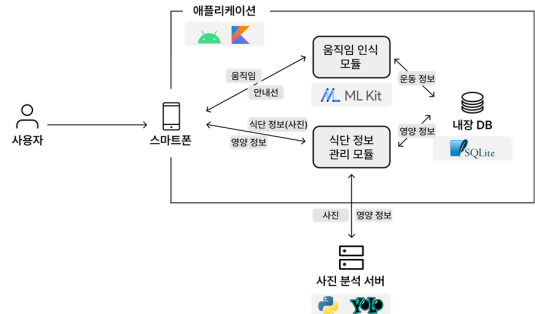


그림 1. 시스템 구성도
Fig. 1. System Architecture Diagram

3. 핵심 알고리즘

가. ML Kit을 활용한 운동 보조

ML Kit의 Pose Detection 프로젝트를 활용한 움직임 인식 모듈은 실시간으로 사용자가 취하는 자세를 분석하여 각 신체 부위별 화면의 좌표를 결과로 가져온다^[8]. 이 분석 결과 데이터를 기반으로 각 운동에 적절한 신체 부위를 선정해, 부위 간 각도 및 거리를 계산한다. 계산된 결과를 이용해 기준이 되는 운동 자세와 유사한지 검사를 수행한다. 만약 이 과정에서 기준을 통과하

지 못하게 되면, 그림 2에서처럼 해당 신체 부위에 대한 안내선을 빨간색으로 변경하고 음성 안내를 출력하여 사용자가 잘못된 자세임을 인지할 수 있도록 한다. 이후에도 운동 동작의 진행에 맞춰 동일한 과정을 수행하며 운동 횟수를 증가시킨다.

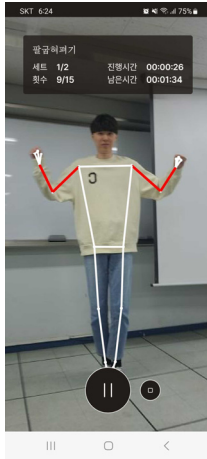


그림 2. 운동 보조
 Fig. 2. Workout Assistance

나. YOLOv5 기반의 음식 분류

음식 분류 기능을 구현하기 위해 YOLOv5를 사용하였다^[9]. YOLO는 이미지 또는 비디오에서 개체를 식별하고 찾기 위한 객체 인식 모델이며, 구현 방식에 따라 다양한 버전으로 나뉜다. 이 중 YOLOv5는 이전에 개발된 다른 모델보다 정확한 객체 탐지가 가능하다는 장점을 가지며, 이후에 개발된 다른 모델에 비해 모델의 크기가 작아 모델 학습을 위해 하드웨어적 부담이 적기 때문에 본 논문에서는 해당 모델을 선택하여 음식 분류에 필요한 모델 학습을 수행하였다.

추가로, 음식 분류의 정확도를 증대시키기 위해 AI-Hub에서 음식 이미지를 수집 후^[10], 직접 선별하는 과정을 거쳤다. 선별된 이미지들의 라벨링을 위해서는 RoboFlow 이미지 라벨링 툴을 이용하였다^[11].

음식 분류 모델은 YOLOv5 기반의 최초 학습 이후에 개선의 과정을 거쳐 완성되었다. 정확도를 높이기 위해 학습에 사용하는 이미지 수를 늘렸으며, optimizer의 종류, batch size, 학습/검증/테스트 데이터 비율을 조정해가면서 최적의 조합을 탐색하였다. 또한, 성능의 편차를 줄이기 위해 데이터 셋에 Background Image를 추가했다. 최종적으로, 표 2의 설정을 적용했을 때 모델의 mAP50 지표에서 5%, mAP50-95 지표에서 66%, 실사

용 테스트에서 37%의 정확도 향상을 이끌어 낼 수 있었다. 그 결과로, 그림 3에서처럼, 이전에는 오분류가 발생했던 음식들도 정확하게 인식되게 되었다.

표 2. 음식 분류 모델 개선 전후의 학습 조건
 Table 2. Learning Conditions Before and After Improving the Food Classification Model

개선 사항	개선 전	개선 후
데이터셋 크기	이미지 각 100장 (음식 5종류) 총 500장	이미지 각 500장 (음식 5종류), Background Image 250장 총 2,750장
map50	0.96	0.987
map50~95	0.556	0.921
epoch	30	240
TRAIN/VALIDATION/TEST SPLIT	88/8/4 (%)	84/9/7 (%)



그림 3. 개선된 음식 분류
 Fig. 3. Improved Food Classification

다. 깊이 맵 계산 기반의 음식 양 추정

본 논문에서 음식 양을 추정하는 방식은 스마트폰에 내장된 카메라의 파라미터 정보들(화각, 센서 크기, 초점 거리 등)과 이미지 처리 기술을 이용한다. 우선, 앞서 진행된 음식 분류 단계에서 결과물로 나온, 이미지 배경과 분리된, 음식들의 크롭된 이미지를 이미지 깊이 추정 모델에 입력하여 깊이 맵(depth map)을 얻는다.

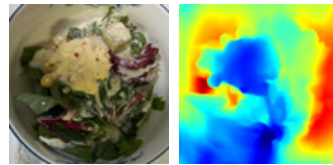


그림 4. 배경과 분리된 이미지와 깊이 맵 이미지
 Fig. 4. Separated Image From Background and Depth Map Image

깊이 맵을 그림 4와 같이 각 픽셀이 그레이 스케일 값(0~255)을 가지는 이미지로 변환하고, K-Means 클러스터링 알고리즘을 이용하여 이미지에서 가장 많은 부분을 차지하는 집단 3개를 찾는다^[12]. 이 집단들이 차지하는 부분을 이미지에서 음식이 있는 부분이라고 추정한다

다. YOLOv5 모델에서 분류되어 나온 이미지의 형태가 직사각형 혹은 정사각형의 사각형이므로, 직사각형에 내접하는 최대 크기의 타원을 해당 음식의 그릇 모양이라고 가정한다. 그릇(이미지 상의 타원 도형)의 실제 크기 및 깊이를 추정하기 위하여 몇 가지 제약사항이 존재한다. 먼저 일반적으로 실생활에서 사용하는 숟가락 및 젓가락의 크기가 20~25cm로 표준화되어 있다는 조사 내용을 기반으로^[13], 촬영된 이들의 크기를 이미지 속 그릇의 실제 크기를 추정하는 데 사용한다. 이 과정을 위해, 애플리케이션에서 사진을 찍어 전송할 때 숟가락 및 젓가락이 고정된 픽셀 크기로 찍힐 수 있도록 프레임을 제시한다. 그림 5는 프레임이 제시된 후 음식을 촬영하는 과정을 나타낸다.

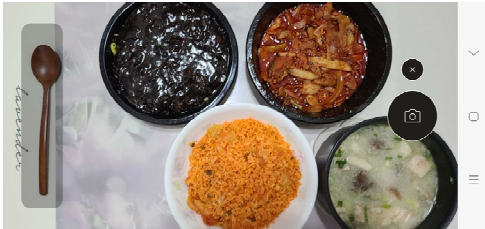


그림 5. 프레임이 제시된 카메라 화면
Fig. 5. Camera Screen with Presented Frame

이를 통해 그릇의 직경, 더 나아가 스마트폰 카메라의 파라미터 정보, 각 픽셀의 깊이 강도(depth intensity)와 결합하여 그릇의 깊이까지 추정이 가능하고, 결과적으로 해당 음식의 부피 추정이 가능해지게 된다. 음식의 부피를 추정하는 알고리즘은 Github에 공개된 오픈소스 프로젝트를 기반으로 한다^[14]. 이미지, 그릇의 직경, 깊이 정보를 입력하면 픽셀 당 길이, 픽셀 당 깊이 값을 계산하고 이를 통해 앞서 음식이 있다고 추정된 부분의 픽셀들의 부피를 계산하여 더해서 총 부피를 구할 수 있다. 총 부피는 그림 6과 같이 파이썬 디셔너리 형태로 제시된다.

```
Volume result :{'Sundaegukbap': 261.74936553301796}
unit: cm^3
```

그림 6. 부피 추정 결과 예시
Fig. 6. Volume Estimation Result Example

추정된 부피 값을 식품 별 부피와 영양 성분 환산 표^[15]에 따라 식품의 총 열량(kcal) 및 탄수화물, 단백질, 지방과 같은 영양 정보로 변환한다. 이 정보를 다시 애플리케이션의 식단 정보 관리 모듈에 전송하는 것으로 음

식 양 추정 과정은 종료된다.

양 추정 성능은 두 차례에 걸쳐 개선하였다. 우선, 순차적이던 처리 구조를 멀티 스레드 기반으로 변경하여, 여러 음식이 포함된 사진에 대한 추정 속도를 40%가량 향상시켰다. 예로, 5가지 음식이 포함된 사진에 대해서는 1분 이상의 시간이 단축되었다. 다음으로는 알고리즘 개선을 통해 정확도 향상을 시도하였다. 앞서 설명한대로 K-Means 클러스터링 알고리즘을 적용하여 음식이 있는 부분을 판단하도록 하였고, 그릇 깊이 계산식을 세부적으로 조정하여 일부분 개선함으로써 실제 값에 조금 더 근접한 양 추정 결과를 얻어낼 수 있게 되었다. 표 3은 두 가지 테스트 케이스에 대해, 알고리즘 개선 전후의 추정 결과와 실제 값을 비교하여 나타내고 있다.

표 3. 양 추정 알고리즘 개선 전후 성능 비교
Table 3. Performance Comparison Before and After Improving the Quantity Estimation Algorithm

개선 사항	테스트 케이스 1			테스트 케이스 2		
	개선 전	개선 후	실제 값	개선 전	개선 후	실제 값
무게(g)	164	587	822	116	855	821
그릇 크기(cm)	12.5	12.5	16	11.9	11.9	19
그릇 깊이(cm)	2.9	7.9	4	2.75	12.2	8
소요 시간(초)	13	14	-	11	13	-

라. TCP 소켓 기반의 서버-클라이언트 통신

Python은 TCP 소켓, UDP 소켓, HTTP 요청, WebSockets, MQTT, ZeroMQ, RabbitMQ, gRPC 등 다양한 방식으로 통신 구현을 지원한다^[16]. Android에서의 Kotlin 또한 위의 통신 방식들을 라이브러리와 API 형태로 지원한다. TCP를 이용한 통신은 신뢰성을 보장하는 과정에서의 지연을 수반한다. 또한, Python에서 TCP 소켓을 기반으로 서버를 구성하면, 네트워크 상태 변화에 따른 재연결 로직이나 스레드 관리를 직접 구현해야 한다는 불편함이 따른다. 하지만 동시에, 소켓 옵션이나 버퍼 크기 등의 매개 변수를 직접 설정하여 세밀한 제어가 가능하며, 불필요한 오버헤드 없이 효율적인 통신이 가능하고, 프로토콜을 원하는 대로 구현할 수 있다는 장점을 가진다.

이러한 장점들을 고려하였으며, Kotlin으로 구현된 Android 애플리케이션과 Python 서버 간에 지속해서 통신을 유지하고, 사용자명, 음식 사진, 분류 결과를 포함하는 JSON 파일 등 다양한 데이터를 교환하기 위해 본 논문에서는 TCP 소켓 기반의 통신 방식을 선택했다.

서버는 수신한 음식 사진을 저장하고 처리하기 위해, 우선 사용자명을 수신한다. 사용자명은 클라이언트에서 음식 촬영 화면에 접속하거나, 앨범에서 사진을 선택하면 전송된다. 그림 7은 이후의 통신 과정을 나타낸다. 사용자명이 정상적으로 전달되고 연결이 확인되면, 이어서 음식 사진이 전송된다. 처리가 끝나면, 식단 분석 결과는 JSON 파일에 저장된다. 서버는 파일을 클라이언트에게 전송하고, 클라이언트는 파일을 받아서 결과를 출력한다. 결과 출력화면으로 이동하면서 클라이언트는 서버와의 연결을 종료한다.

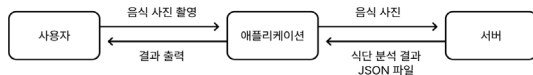


그림 7. 통신 구조 및 송수신 데이터
 Fig. 7. Communication Structure and Transmitted /Received Data

그림 8과 그림 9는 각각 클라이언트와 서버에서 상대방으로 전송하는 메시지의 형태를 정의하는 프로토콜이다. 메시지는 공통으로 시작 코드(), 종료 코드 ()와 메시지 크기를 포함하며, 서버와 클라이언트는 이를 활용해 메시지가 정상적으로 수신되었는지 확인한다. 사진 및 JSON 파일의 경우, 메시지를 먼저 전송한 후에 파일을 전송하도록 구현했으며 수신 측에서 파일 크기 정보와 실제 수신한 데이터 크기를 비교하도록 하여 정확성을 확보했다.

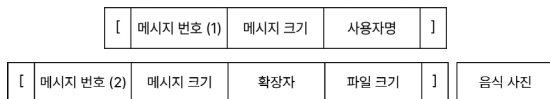


그림 8. 사용자명, 음식 사진 전송 프로토콜
 Fig. 8. Username and Food Photo Transmission Protocol



그림 9. 식단 분석 결과 전송 프로토콜
 Fig. 9. Food Classification Result Transmission Protocol

IV. 결 론

본 논문에서는 ML Kit의 자세 인식 기술을 활용한 운동 보조 시스템과 YOLOv5 기반의 이미지 분류 및 깊이 맵 계산 기반의 식단 보조 시스템을 통해 홈 트레이닝 및 식단 관리의 편의성 향상을 실현하는 애플리케이션을 개

발하였다. 운동과 식단 보조 기능을 동시에 제공하면서도, 단순화된 UI와 사용성을 고려한 보조 기능들을 제공하기 때문에 기존 애플리케이션들의 단점을 보완하고 사용자들에게 개선된 경험을 제공할 수 있을 것이다.

이후에는 운동 보조 기능에 신체 부위의 이동에 따라 파티클이 흩어지는 효과와 동작 완료 시 배경색이 변화하는 애니메이션 등 각종 시각효과를 추가할 것이다. 식단 보조 기능과 관련해서는 상용화 및 정확도 개선을 위해 선호도가 높은 30종류의 음식에 대해 각 1,000장 이상의 이미지로 음식 분류 학습을 추가 진행할 계획이다^[17]. 현재 음식 양 추정에서는 K-Means 알고리즘을 적용 후, 크기가 큰 집단 3개를 음식이라고 판단해 고정적으로 선택한다. 이 부분을 실제 음식 각각의 특징을 고려해, 음식에 따라 적절한 수의 집단을 선택할 수 있도록 개선하는 방향으로 정확도 향상을 시도할 예정이다.

References

- [1] Korea Consumer Agency, "Acceleration of Home Training Due to COVID-19 and Consumer Issues", 2021.
- [2] <https://howfit.ai/>
- [3] <https://www.fatsecret.kr/>
- [4] <https://www.doinglab.com/home>
- [5] <https://developer.android.com/studio/releases/past-releases/as-dolphin-release-notes>
- [6] https://code.visualstudio.com/updates/v1_79
- [7] <https://colab.google/notebooks/>
- [8] <https://developers.google.com/ml-kit/vision/pose-detection?hl=ko>
- [9] https://colab.research.google.com/github/pytorch/pytorch.github.io/blob/master/assets/hub/ultralytics_yolov5.ipynb
- [10] <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=realm&dataSetSn=74>
- [11] <https://docs.roboflow.com/>
- [12] Tae-Woo Kim, Hyeon-Joong Yoo, "Color Code Detection and Recognition Using Image Segmentation Based on k-Means Clustering Algorithm", Journal of Korea Academia-Industrial cooperation Society, Vol. 7, No. 6, p. 1103, 2006.
- [13] <https://thesoulofseoul.net/korean-metal-chopsticks/>
- [14] <https://github.com/WhiteXiez/food-volume-estimation#end-to-end-multi-view-supervision-for-single-view-food-volume-estimation>

- [15] Yoonna Lee, "Development of Food and Nutrient Database - Food Portion/Weight Database", Korea Health Industry Development Institute, pp. 61-168, 2007.
DOI: <https://doi.org/10.23000/TRKO201700002083>
- [16] <https://docs.python.org/3/library/ipc.html>
- [17] <https://www.gallup.co.kr/gallupdb/reportContent.asp?seqNo=614>

황 인 응(준회원)



- 한국공학대학교 컴퓨터공학부 재학
- 관심분야 : 안드로이드, 컴퓨터 비전

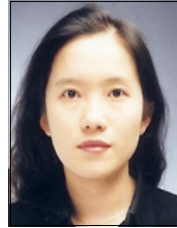
저 자 소 개

이 치 호(준회원)



- 한국공학대학교 컴퓨터공학부 재학
- 관심분야 : 안드로이드, UI/UX 디자인

한 경 속(정회원)



- (학사) 홍익대학교 컴퓨터공학과
- (석사) 홍익대학교 전자계산학과
- (박사) 홍익대학교 전자계산학과
- 한국공학대학교 컴퓨터공학부 교수
- 관심분야 : 컴파일러 최적화, 프로그래밍 언어

김 동 현(준회원)



- 한국공학대학교 컴퓨터공학부 재학
- 관심분야 : 안드로이드

최 승 호(준회원)



- 한국공학대학교 컴퓨터공학부 재학
- 관심분야 : iOS, 자료구조

※ 본 연구는 2019년 교신저자의 한국공학대학교 연구년 기간동안 수행한 결과물임