Smart Media Journal / Vol.15, No.1 / ISSN:2287-1322
https://dx.doi.org/10.30693/SMJ.2026.15.1.16

Swin-V2 기반 음식 이미지 분류 및 기초대사량 연계 식단 추천 시스템

(Food Image Classification and BMR-Aware Diet Recommendation System Based on Swin-V2)

윤혜성*

(Hye-Seong Yoon)

요 약

현대인의 식생활에서 초가공식품과 고열량 식품 섭취가 증가하면서 비만 및 대사성 질환이 확산되고 있으며, 이에 따라 개인이 섭취한 음식의 종류와 영양 성분을 자동으로 기록·관리하는 기술의 필요성이 커지고 있다. 본 연구는 음식 이미지를 입력으로 받아 해당 음식의 종류를 분류하는 Swin-V2 기반 딥러닝 모델과 기초대사량(Basal Metabolic Rate, BMR) 기반 식단 추천 모듈을 구성하고, 이를 연계한 모바일 애플리케이션 구조를 제안한다. 실험에는 Food-101 데이터셋 101개 클래스 중 실제 일상 식단에서 자주 등장하는 10개 클래스를 선정하여 클래스당 1,000장씩을 사용하였으며, 연산 자원과 학습 시간의 제약으로 전체 클래스를 직접 평가하지 못했다는 한계가 있다. 제안 모델은 ImageNet 사전학습 가중치를 사용한 Swin-V2-Tiny를 전층 미세조정한 것으로, Swin-V1과 여러 CNN 기반 모델과 비교했을 때 Top-1 정확도 0.926, macro F1-Score 0.926으로 모든 비교 모델보다 우수한 성능을 보였다. 또한 자동 음식 인식 및 식단 추천을 통합한 프로토타입 시스템을 제안한다. 향후 Food-101 전체 클래스 및 자가 촬영 이미지로 데이터 범위를 확장하고 음식 분량 추정 및 모델 경량화를 통해 상용 서비스에 적합한 통합 식단 관리 시스템으로 발전시키고자 한다.

■ 중심어 : 딥러닝 ; 스윈트랜스포머V2 ; 음식인식 ; 식단추천

Abstract

In modern diets, increasing consumption of ultra-processed and high-calorie foods is accelerating obesity and metabolic disorders, highlighting the need for technologies that automatically record and manage food types and nutritional intake. This study proposes a mobile application architecture that integrates a Swin-V2-based deep learning model for food image classification with a basal metabolic rate (BMR)-based meal recommendation module. Experiments use 10,000 images from 10 frequently consumed classes selected from the 101-class Food-101 dataset, with the limitation that computational and time constraints prevented evaluation on all classes. The proposed Swin-V2-Tiny model, fine-tuned from ImageNet pre-trained weights, achieves a Top-1 accuracy of 0.926 and a macro F1-Score of 0.926, outperforming Swin-V1 and several CNN-based baselines. The prototype system shows that automatic food recognition and diet recommendation can be effectively combined, and future work will extend the dataset, incorporate portion-size estimation, and apply model compression to build an integrated diet-management system suitable for real-world services.

■ keywords : Deep Learning ; SwinTransformerV2 ; Food recognition ; Diet recommendation

Ⅰ. 서 론

현대인의 식생활에서 초가공식품 섭취가 전 세계적으로 증가함에 따라, 비만과 당뇨병의 유병률 또한 가파르게 상승하는 추세이다[1][2]. 비만은 고혈압, 심혈관 질환 등 다수의 만성질환의 핵심 위험 요인일 뿐만 아니라, 전체 사망 위험 증가와도 깊은 연관이 있다. 대규모 메타분석 연구에서는 체질량지수 증가와 모든 원인에 의한 사망률 사이에 유의미한 상관관계가 있음이 보고되었으며[3] 국제 암 연구소

* 준회원, 호남대학교 컴퓨터공학과

접수일자 : 2025년 12월 11일
수정일자 : 2026년 01월 05일
게재확정일 : 2026년 01월 14일
교신저자 : 윤혜성 e-mail : remisen326@naver.com

(IARC) 검토에 따르면 과도한 체지방은 대장암, 직장암, 위문분암, 간암, 담낭암, 췌장암 등 다수 암종의 발병 위험을 높인다고 경고했다[4].

이처럼 식단이 건강에 미치는 영향은 지대하지만, 현대인의 식사는 갈수록 복잡해지고 있어 섭취하는 음식의 영양 성분을 정확히 파악하기는 더욱 어려워졌다[5]. 식단 관리의 첫걸음은 무엇을 얼마나 먹는지 정확히 파악하고 기록하는 것이지만, 수동 기록 방식은 번거롭고 지속하기 어렵다는 한계가 있다[6].

이러한 배경 속에서, 지난 수년간 눈부신 발전을 거듭한 딥러닝 기술은 이미지 분야에 혁신적인 해결책을 제시하고 있다. 특히 Vision Transformer의 단점을 보완하며 등장한 Swin Transformer는 계층적 구조와 윈도우 개념을 도입하여 CNN의 지역적 특징 포착 능력과 Transformer의 전역적 관계 학습 능력을 효과적으로 결합했다. 이를 통해 다양한 컴퓨터 비전 벤치마크에서 SOTA를 달성했다. 이후 나온 Swin-V2 역시도 SOTA를 달성하였다.

본 연구는 음식 이미지 데이터셋에 맞게 Swin-V2를 전이 학습하고 파인튜닝하여, 높은 정확도로 음식을 분류하는 모델을 개발하는 것 뿐만 아니라 음식 인식을 위한 실용적인 애플리케이션을 구축하는 것을 목표로 한다. 이렇게 개발된 모델은 공신력 있는 영양 성분 데이터베이스와 연계될 경우, 사용자가 음식 사진을 촬영하는 것만으로 간편하게 칼로리와 영양 정보를 추정할 수 있는 모바일 애플리케이션의 핵심 기술로 활용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 CNN부터 Swin Transformer에 이르기까지 딥러닝 기반 음식 인식 관련 연구 동향을 살펴보고, 3장에서는 연구에서 사용된 Swin-V2 아키텍처와 데이터셋, 학습 방법을 상세히 기술한다. 4장에서는 실험 결과와 성능을 분석하며, 마지막 5장에서는 연구를 요약하고 향후 연구 방향을 제시하며 결론을 맺는다.

본 논문의 주요 기여는 다음과 같이 정리할 수 있다.

첫째, 대규모 ImageNet 사전학습 가중치를 활용한 Swin-V2-Tiny 기반 음식 인식 모델을 설계하고, Food-101 데이터셋에서 실제 일상 식단에서 자주 등장하고 시각적으로 구분이 가능한 하위 10개 클래스(총 10,000장)를 대상으로 전이학습 및 파인튜닝을 수행하여 Top-1 정확도 0.926, macro F1-Score 0.926의 성능을 달성하였다. Swin-V2는 timm 라이브러리에서 제공하는 표준 파인튜닝 설정을 그대로 사용하고, Swin-V1과 EfficientNetB0, ResNet50, MobileNetV2, VGG16 비교 대상 모델에는 각 프레임워크에서 권장되는 분류 헤드 구성과 비교적 공격적인 학습 전략을 허용하여 성능을 최대한 끌어올린 뒤, 동일한 데이터 분할과 입력 해상도 조건에서 정량적으로 비교하였다. 이를 통해 제안 모델의 성능을 보수적인 기준에서 평가하면서도, 최신 Transformer 계열 구조가 강력한 CNN·Transformer 기반 기준선보다 우수한 표현력과 일반화 성능을 제공함을 정량적으로 입증하였다.

둘째, 기초대사량(Basal Metabolic Rate, BMR)과 하루 누적 섭취 열량, 후보 식품 집합의 열량·영양 정보에 기반하여 남은 칼로리 예산 내에서 두 끼 혹은 한 끼 분할 식단을 구성하는 식단 추천 알고리즘을 수식으로 정식화하고, Swin-V2 기반 음식 분류 모듈과 연계 가능한 모바일 애플리케이션 아키텍처로 구체화하였다. 이를 통해 음식 인식 결과를 단순 분류 정확도 평가를 넘어, 실제 사용자 시나리오에서 남은 끼니의 식단을 구성하는 데 활용할 수 있는 응용 예시를 제시하고, 향후 음식 인식·영양 관리 연구에서 재현 가능한 실험·시스템 설계 기준선(baseline)을 제공하였다.

부가적으로, 학습률 워밍업과 코사인 스케줄링, 레이블 스무딩(label smoothing), 혼합 정밀도 학습(mixed-precision training) 등을 결합한 학습 전략을 도입하여, 비교적 제한된 연산 자원 환경에서도 Swin-V2를 안정적으로 수렴시키는 실무적인 파인튜닝 설정을 정리하였다.

본 연구에서는 연산 자원과 학습 시간의 제약을 고려하여, Food-101의 101개 전체 클래스가 아닌 하위 10개 클래스를 대상으로 탐색적 실험을 수행하였다. 이러한 설계로 인해 Food-101 전체 클래스에 대한 일반화 성능을 직접적으로 평가하지 못한다는 한계가 존재하나, 후속 연구에서는 본 연구에서 구축한 모델과 실험 환경을 기반으로 전체 101개 클래스 및 자가 촬영 이미지로 범위를 확장할 계획이다.

Ⅱ. 관 련 연 구

딥러닝 기반 시각 인식은 GPU의 연산 성능의 비약적 향상, 대규모 공개 데이터셋의 등장 그리고 프레임워크 및 최적화 기술의 고도화로 발전했다. 특히, CNN의 등장은 이미지 인식 분야에 큰 발전을 가져왔다. CNN의 합성곱 커널로 인한 뛰어난 특징 추출 능력을 기반으로 다양한 이미지 인식 분류 작업에 사용되었다. KAGAYA, Hokuto; AIZAWA, Kiyoharu; OGAWA, Makoto 등의 경우에는 CNN를 이용하여 음식 이미지를 감지하였다[7]. POULADZADEH, Parisa; SHIRMOHAMMADI, Shervin 등은 FooDD 데이터셋을 사용한 CNN으로 94.11%의 정확도를 달성하였다[8]. MARTINEL, Niki; FORESTI, Gian Luca; MICHELONI, Christian 등은 Food-101 데이터세트에서 Wide-Slice Residual Networks를 도입하여 Top-1 정확도 90.27%를 달성했다[9]. ZAHISHAM, Zharfan; LEE, Chin Poo; LIM, Kian Ming 등은 사전 훈련된 ResNet 50을 사용하여 ETHZ-FOOD101, UECFOOD10, UECFOOD256 데이터 세트에서 음식을 인식했다[10]. PHIPHIPHATPHAISIT, Sirawan; SURINTA, Olarik 등은 전역 풀링 계층, 배치 정규화, 소프트맥스를 적용하여 정확도를 끌어올린 MobileNet 아키텍처로 음식 이미지를 분류했다[11]. PARVATHAVARTHINI, S., et al 등은 AlexNet, EfficientNet을 활용하여 식품을 인식했다[12].

하지만 CNN은 합성곱 연산의 특성상 이미지 내의 지역적 특징을 포착하는 데는 강하지만, 이미지 전반에 걸친 전역적 맥락이나 장거리 의존성을 학습하는 데는 본질적인 한계를 가진다.

2020년, 자연어 처리 분야에서 성공을 거둔 Transformer 아키텍처를 이미지에 적용한 Vision Transformer를 DOSOVITSKIY, Alexey, et al 등이 발표하면서 새로운 패러다임이 열렸다. Vision Transformer는 이미지를 여러 개의 패치로 분할하고, 이를 토큰 시퀀스로 변환한 뒤 자기 어텐션 매커니즘을 통해 패치 간의 전역적 관계를 학습한다. 이러한 접근법은 CNN에 비해 Inductive bias가 적어 대규모 데이터셋으로 학습했을 때 CNN을 능가하는 성능을 보였으며 다수의 이미지 인식 벤치마크에서 SOTA를 달성했다[13]. GAO, Xinle; XIAO, Zhiyong; DENG, Zhaohong 등은 Augmentplus와 모델을 더 깊게 쌓을 때의 성능 저하를 방지하기 위한 LayerScale을 통해 ViT의 성능을 한층 더 끌어올린 AlsmViT로 음식 이미지 분류 정확도 95.17%를 달성했다.

그러나 Vision Transformer는 고정된 크기의 패치를 사용하기 때문에 다양한 크기의 객체를 처리하기 어렵고, 모든 패치 간의 관계를 계산하는 과정에서 입력 이미지의 해상도가 커질수록 연산량이 기하급수적으로 증가하는 문제점을 안고 있었다. 2021년 LIU, Ze, et al.가 제안한 Swin Transformer는 이러한 Vision Transformer의 한계를 극복했다. Swin Transformer는 계층적 특징 맵을 생성하여 다양한 스케일의 정보를 처리하며, Shifted Window 내에서만 어텐션을 계산하여 연산 효율을 크게 향상시켰다. 이 구조는 CNN의 장점인 지역성과 계층 구조를 Transformer에 효과적으로 통합하여, 적은 데이터로도 높은 성능을 낼 수 있게 만들었다. 동시에 다수의 이미지 인식 벤치마크에서 SOTA를 달성했다[14]. 이어서 LIU, Ze, et al.는 후속 연구에서 Swin Transformer V2를 제안하여, 안정적인 학습을 위한 스케일링 전략과 개선된 상대적 위치 인코딩 기법 등을 도입함으로써 보다 대규모 입력 해상도와 모델 용량까지 확장 가능한 구조

를 제시하였다[15]. Swin-V2는 여러 이미지 인식 벤치마크에서 추가적인 성능 향상을 달성하며, 대규모 비전 과제에 적용 가능한 범용 백본으로 활용되고 있다.

III. 본 론

1. 데이터셋 구성

본 연구에서는 음식 이미지를 입력으로 받아 해당 음식을 분류하는 모델을 구축하였다. 실험에 사용한 데이터셋은 Food-101 공개 데이터셋의 일부로, 총 101개 음식 클래스 중 10개 클래스를 선정하여 사용하였다. 각 클래스당 1,000장의 이미지를 사용하여, 전체 데이터 수는 10,000장으로 구성하였다.

파이토치 기반 Swin 계열 실험에서는 전체 이미지를 하나의 디렉터리에 클래스별 하위 폴더 구조로 배치한 뒤, StratifiedShuffleSplit를 이용해 학습/검증 데이터를 8:2 비율로 층화 분할하였다. 이렇게 분할된 인덱스를 기준으로, 학습용과 검증용 데이터셋에 각각 서로 다른 변환(transform)을 적용하였다.

Swin-V1의 경우 입력 크기를 224×224로 맞추기 위해 Resize(256) 후 RandomResizedCrop(224, scale=(0.9, 1.0)), RandomHorizontalFlip() 및 RandAugment 또는 ColorJitter를 사용하였다. CNN 계열 모델(Keras 기반)은 ImageDataGenerator를 사용하여, 224×224 크기로 리사이즈한 뒤 0~1 구간으로 스케일링(rescale=1/255)하였으며, 동일한 디렉터리 구조에서 validation_split=0.2 옵션을 사용해 8:2 비율로 학습/검증 데이터를 분할하였다.

그림 1. 사용된 데이터셋

표 1. 사용된 데이터셋 목록

사용된 데이터셋 목록	
1. 와플	2. 폭찹
3. 피자	4. 팬케이크
5. 나초	6. 애플파이
7. 아이스크림	8. 도넛
9. 추로스	10. 비빔밥

2. Swin-V2 기반 음식 분류 모델

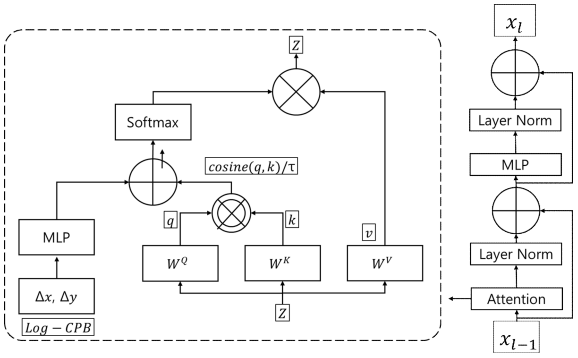

그림 2. Swin-V2 아키텍처

본 연구에서는 음식 이미지 인식 모듈로 LIU, Ze, et al.[15]이 제안한 Swin-V2 아키텍처를 기반으로 한 구성을 백본으로 사용하였다. 제안하는 모델은 PyTorch 및 timm 라이브러리를 활용하여 구현한 Swin-V2 계열 모델이다. 구체적으로는 swinv2_tiny_window16_256 아키텍처를 사용하였으며, 이는 256×256 해상도의 입력 이미지를 16×16 크기의 패치 단위 윈도우로 분할하여, 계층적(feature pyramid) 구조와 Shifted Window Self-Attention을 결합한 형태의 모델이다. 모델은 ImageNet-1K 데이터셋으로 사전 학습된 가중치를 초기값으로 사용하고, Food-101 하위 10개 음식 클래스에 맞추어 최종 분류기 헤드를 num_classes=10으로 재구성하였다.

Swin-V2의 모든 파라미터는 전층 미세조정(full fine-tuning) 방식으로 학습하였으며, 네트워크 깊이와 DropPath 비율은 timm 기본 설정을 따르되, 과적합 완화를 위해 DropPath 비율을 0.1로 설정하였다. 학습 시에는 PyTorch의 자동 혼합 정밀도 기능(torch.amp.autocast)과 그라디언트 스케일러(GradScaler)를 사용하여 GPU 메모리 사용량을 절감하면서 연산 속도를 향상시켰다.

3. 식단 추천 모듈

본 절에서는 앱에 사용된 식단 추천 모듈에 대해 설명한다. 식단 추천 모듈은 사용자의 기초대사량(Basal Metabolic Rate, BMR)과 하루 동안의 누적 섭취 열량을 이용하여, 잔여 열량 예산 안에서 영양성분이 균형 잡힌 식단을 자동으로 추천하는 방식으로, 상세한 내용은 아래와 같다.

사용자가 앱에서 BMR을 설정하면, 시스템은 안전 마진 E_{safe}을 제외한 1일 목표 열량과 남은 열량 예산을 다음 수식 (1), (2)과 같이 계산한다.

$$E_{target} = \max(0, BMR - E_{safe}), \quad (1)$$

$$E_{remain}(t) = \max(0, BMR - E_{consummed}(t) - E_{safe}) \quad (2)$$

수식 (2)의 $E_{consummed}(t)$는 시간 t까지 누적 섭취 열량이며, 수식 (1)의 E_{safe}는 200kcal로 설정하였다.

앱 내부에는 미리 정의된 식품 집합 $F = \{f_i, ..., f_n\}$이 저장되어 있다.
각 식품 f_i에 대해, 1회 제공량 기준의 에너지와 4대 영양소 정보를 다음 수식 (3)과 같이 둔다.

$$f_i = (e_i, c_i, p_i, f_i, fib_i) \quad (3)$$

위 수식 (3)에서 e_i는 열량을 의미하고, c_i는 탄수화물, p_i는 단백질, f_i는 지방, fib_i는 식이섬유를 의미한다.

하나의 끼니에 대해 목표 열량E이 주어졌을 때 본 연구는 탄수화물/단백질/지방 의 비율을 미리 지정하고, 식이섬유는 일정치 이상 만족하는 조합을 탐색한다.

$$rc = 0.45, \ rp = 0.25, \ rf = 0.30 \quad (4)$$

수식 (4)는 한 끼에서 사용되는 탄수화물/단백질/지방의 비율을 정의한 식이다.

수식 (4) 에 따라 한 끼에서 목표로 하는 탄수화물, 단백질, 지방은 다음 수식 (5)으로 계산한다.

$$C^* = \frac{rcE}{4}, \ P^* = \frac{rpE}{4}, \ F^* = \frac{rfE}{9} \quad (5)$$

또한 한 끼당 권장 최소 식이섬유량을 다음 수식 (6)로 두고, 식이섬유가 부족할 경우에만 패널티를 부여한다.

$$Fib_{\min} = 8g \quad (6)$$

특정 식단 조합 $S \subset F$에 대해 실제 합계는 다음 수식 (7)과 같이 정의한다.

$$\begin{aligned} E(S) &= \sum_{i \in s} e_i, \quad (7) \\ C(S) &= \sum_{i \in S} c_i, \\ P(S) &= \sum_{i \in S} p_i, \\ F(S) &= \sum_{i \in S} f_i, \\ Fib(S) &= \sum_{i \in S} fib_i \end{aligned}$$

이때 조합 S의 품질을 평가하는 점수 함수 $J(S)$를 다음 수식 (8)과 같이 정의한다.

$$\begin{aligned} J(S) = {} & w_E |E - E(S)| + \quad (8) \\ & w_C |C^* - C(S)| + w_p |P^* - P(S)| \\ & + w_F |F^* - F(S)| + \\ & w_{fib} \max(0, Fib_{\min} - Fib(S)) \end{aligned}$$

구현에서는 다음 수식 (9)을 사용하였다.

$$w_E = 1.0, \ w_c = 0.5, \ w_p = 0.6, w_F = 0.5, \ w_{fib} = 0.4 \quad (9)$$

한끼 목표 열량 E가 주어졌을 때, 본 연구는 후보 식품 중 1~3개를 선택하는 모든 조합을 열거하여 점수 $J(S)$가 최소가 되는 조합을 다음 수식 (10)처럼 선택한다.

$$S^* = \arg \min_{S \subset F, 1 \le |S| \le 3} J(S) \quad (10)$$

단, 구현에서 사용한 식품 수n가 적고, 조합 크기를 3개 이하로 제한하므로 전체 조합 수$O(n^3)$는 스마트폰 환경에서 충분히 처리 가능한 수준이다.

동일한 점수의 후보가 여러 개 존재할 경우, 열량이 목표보다 과도하게 높아지는 것을 방지하기 위해 목표 열량 이하인 조합을 우선 선택한다. 이 과정을 함수로 구현하여 한 끼 추천에 사용한다.

전체 예산 B를 두 끼로 나누어 추천할 때는, 평균값 $B/2$를 각 끼의 목표 열량으로 사용한다.

먼저 첫 번째 끼니를 구하는 방법은 다음 수식 (11)과 같다.

$$S_1^* = \arg\min J(S; E = B/2) \quad (11)$$

두 번째 끼니는 첫 끼니에 사용된 식품을 제외한 집합 $F' = F \setminus S_1^*$에 대해 같은 방식으로 다음 수식 (12)과 같이 계산한다.

$$S_2^* = \arg \min_{S \subset F'} J(S; E = B/2) \quad (12)$$

만약 두 번째 끼니에서 적절한 조합을 찾지 못하거나 두 끼니의 구성S_1^*, S_2^*이 완전히 동일하면, 남은 예산 $B - E(S_1^*)$를 기준으로 재탐색하여 가능한 다른 조합을 선택한다.

요약하자면, 제안된 식단 추천 모듈은 다음과 같은 순서를 갖는다.

(1) 사용자별 BMR과 실제 섭취 기록을 바탕으로 남은 칼로리 예산을 계산한다.

(2) 제한된 후보 식품 집합에서 동적 계획법을 사용하여 해당 예산을 초과하지 않는 범위에서 목표 칼로리에 가장 근접하는 조합을 생성한다.

(3) 필요 시 0.5 서빙 단위까지 고려하여 잔여 칼로리를 정밀하게 조정한다.

이를 Swin-V2 기반 음식 분류 모델과 연계할 경우, Swin-V2 기반 음식 분류 모델의 예측값을 바탕으로 남은 끼니의 식단을 구성함으로써, 음식 인식 결과는 단순한 분류 정보를 넘어 사용자의 에너지 균형을 고려한 식단 추천 기능으로 확장된다.

4. 앱 구조 설명

본 논문에서 구현한 Swin-V2 기반 음식 앱의 음식 인식 프로세스는 [그림 3]과 같다. 사진을 SwinV2로 분석 후 이에 해당하는 음식 정보를 사용자에게 화면으로 보여주는 과정이다.

본 연구에서 구현한 음식 인식 기반 모바일 애플리케이션은 크게 (1) 사용자 정보 입력 모듈, (2) 음식 이미지 인식 모듈, (3) 식단 추천 모듈의 세 부분으로 구성된다. 사용자는 먼저 키, 체중, 연령, 성별 등 기본 정보를 입력하여 기초대사량을 계산하고, 이후 “음식 촬영” 또는 “식단 추천” 기능을 선택한다. 음식 촬영 기능을 선택하면 사용자가 촬영한 음식 이미지가 서버 혹은 단말 내 2. Swin-V2 기반 분류 모델 절에 기재한 대로 학습한 모델을 ONNX로 변환한 파일에 입력값으로 들어가고, Swin-V2 기반 음식 분류 모델의 예측 결과에 따라 해당 음식의 명칭과 영양 성분(칼로리, 탄수화물, 지방, 단백질 등)이 조회된다. 사용자는 음식을 인식하거나 인식하지 않은 상태에서도 자유롭게 식단 추천 기능을 사용할 수 있으며, 기초대사량 및 기존 섭취량 정보를 바탕으로 남은 칼로리 예산 내에서 적절한 음식 조합을 제안하는 방식으로 앱은 동작한다. 식단 추천 기능은 3. 식단 추천 모듈 절의 구조로 이루어져 있다.

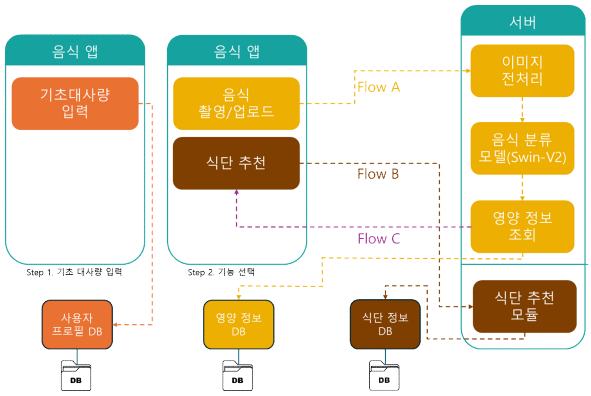

그림 3. 제안하는 앱의 주요 프로세스

Ⅳ. 실 험

1. 실험 환경

모든 실험은 Google Colab 기반의 단일 GPU 환경에서 수행하였다. Swin 계열 모델은 PyTorch 2.x 및 timm 라이브러리를 사용하여 구현하였고, CNN 계열 비교 모델은 TensorFlow 2.x 및 Keras를 사용하여 구현하였다. 데이터셋 로딩과 전처리에는 PyTorch의 torchvision 및 Keras의 ImageDataGenerator를 각각 사용하였다. 재현성을 확보하기 위해 파이썬의 random, NumPy, PyTorch의 시드 값을 고정하고, PyTorch의 CuDNN 관련 옵션 중 비결정적 연산을 비활성화하였다.

2. 비교 대상

제안 모델의 성능을 검증하기 위해, Swin-V1과 대표적인 CNN 기반 분류 모델(EfficientNetB0, ResNet50, MobileNetV2, VGG16)을 비교 대상으로 선정하였다. Swin-V1은 torchvision에서 제공하는 swin_s(Swin-Small) 모델을 사용하였으며, 마찬가지로 ImageNet 사전학습 가중치를 불러온 뒤, 최종 분류 헤드를 음식 클래스 수에 맞게 교체하였다. Swin-V1의 경우 계층별 특성을 고려하여 stage1~4에 서로 다른 학습률을 부여하고, 최상위 분류 헤드에는 상대적으로 큰 학습률을 적용하는 layer-wise learning rate 전략을 사용하였다. 반면 Swin-V2는 timm 라이브러리에서 제공하는 대표적인 파인튜닝 레시피를 그대로 사용하는 것을 목표로 하였다. , Swin-V2에 대해서는 추가적인 layer-wise 튜닝을 적용하지 않고, 널리 사용되는 기본 설정만으로도 어느 수준의 성능을 달성하는지를 확인하고자 하였다. 이로 인해 동일 계열 모델 간에 세부 학습 전략이 완전히 일치하지는 않지만, 이전 아키텍처(Swin-V1)에는 비교적 공격적인 하이퍼파라미터 튜닝을 허용하는 반면, 최신 구조인 Swin-V2는 “표준 설정”만 사용한 상태에서 비교를 수행하였다는 점에서, 제안 모델의 우수성을 다소 보수적으로 평가한 결과로 해석할 수 있다.

CNN 계열 비교 모델은 TensorFlow/Keras 환경에서 구현하였다. EfficientNetB0, ResNet50, MobileNetV2, VGG16 모두 ImageNet 사전학습 가중치를 불러온 뒤, 최상단 분류층을 제거하고 다음과 같은 공통 구조의 분류 헤드를 추가하였다.

Global Average Pooling
512차원 완전연결층 + ReLU 활성함수
배치 정규화
Dropout(0.5)
256차원 완전연결층 + ReLU 활성함수
최종 Softmax 분류층(클래스 수 = 10)

본 연구에서는 CNN 계열 비교 모델(EfficientNetB0, ResNet50, MobileNetV2, VGG16)의 분류 헤드를 다음과 같이 통일하여 설계하였다. 이러한 구성은 서로 다른 백본(backbone) 구조 위에서 가능한 한 유사한 분류 헤드 구조를 유지함으로써, 백본 자체의 표현력 차이를 상대적으로 공정하게 비교하기 위함이다. 먼저 백본 네트워크의 최종 특징 맵에 Global Average Pooling을 적용하여 고정 길이의 특징 벡터를 얻고, 이에 512차원 완전연결층과 256차원 완전연결층을 순차적으로 연결하였다. 각 완전연결층 뒤에는 ReLU 활성함수, 배

치 정규화(batch normalization), 드롭아웃(dropout, p = 0.5)을 적용하였으며, 마지막으로 10개 음식 클래스에 대응하는 Softmax 출력층을 연결하여 최종 예측 확률을 계산하였다. 이러한 구성은 AlexNet, VGG 등 기존 이미지 분류 연구에서 널리 사용되어 온 다단계 완전연결 분류기 구조와 드롭아웃 설정을 참고한 것이다[16, 17]. 본 연구에서는 데이터셋 규모와 모델 복잡도, 연산량을 종합적으로 고려하여 중간 완전연결층의 차원을 512와 256으로 설정하였으며, 이는 파라미터 수와 표현력, 정규화 효과 간의 균형을 맞추기 위한 실험적 선택이다.

3. 평가 지표

모델 성능 평가는 Top-1 분류 정확도(Accuracy)와 macro F1-Score를 주요 지표로 사용하였다. Accuracy는 모델이 예측한 확률이 가장 높은 클래스가 실제 정답과 일치하는 비율을 의미한다. 반면 macro F1-Score는 각 클래스별 F1-Score를 산출한 뒤 산술평균하는 방식으로 계산되며, 클래스별 표본 수가 동일하더라도 각 클래스에 대해 균등한 중요도를 부여한다. 본 연구에서는 클래스 간 데이터 수를 동일하게 맞추었기 때문에, Accuracy와 macro F1-Score가 유사한 값을 보이는 것이 자연스러운 결과이며, 이는 모델이 특정 클래스에 편향되지 않고 전반적으로 균형 잡힌 분류 성능을 보였음을 시사한다.

4. Swin-V2 학습 곡선 분석

그림 4. Swin-V2 Loss

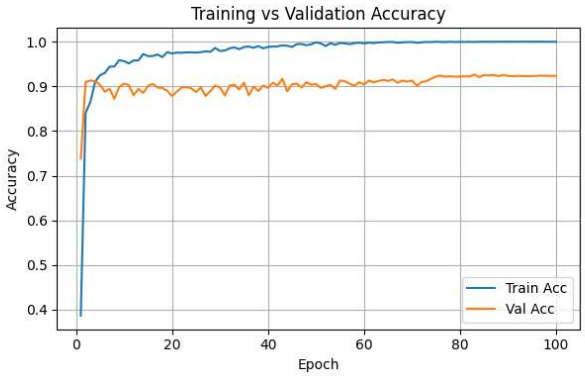

그림 5. Swin-V2 Accuracy

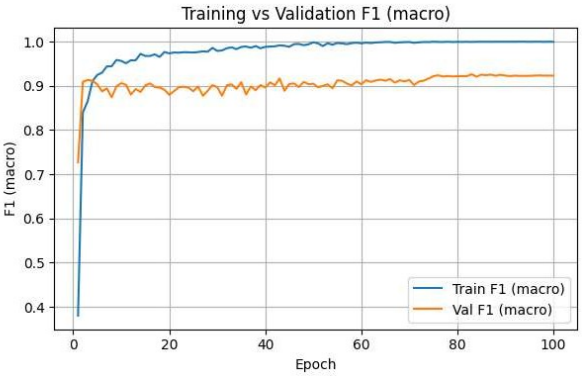

그림 6. Swin-V2 F1 Score(macro)

Swin-V2 모델을 100에폭 동안 학습한 결과, 학습 손실과 검증 손실은 초반 급격히 감소한 뒤 일정 수준에서 수렴하는 양상을 보였다[그림 4]. 학습 정확도와 검증 정확도 역시 에폭이 증가함에 따라 우상향하며, 중후반부에는 0.9 이상에서 안정적으로 유지되었다[그림 5]. macro F1-Score 또한 유사한 추세를 보이며, 초기 10에폭 이내에 급격한 성능 향상이 이루어진 뒤 점진적 개선을 거쳐 최종적으로 0.926에 도달하였다[그림 6].

레이블 스무딩과 데이터 증강을 적용했음에도 불구하고, 학습 곡선 상에서 과도한 overfitting 징후(예: 학습 정확도는 상승하는데 검증 정확도가 하락하거나 정체되는 양상)는 뚜렷하게 관찰되지 않았다. 이는 (1) 클래스별 데이터 수를 균일하게 맞춘 점, (2) 수평 뒤집기, 회전, 색상 변형과 같은 적절한 수준의 데이터 증강을 도입한 점, (3) AdamW 기반의 가중치 감쇠(Weight Decay)와 Cosine Annealing 스케줄을 통해 학습 후반부의 과도한 파라미터 진동을 억제한 점 등이 복합적으로 기여한 결과로 해석할 수 있다.

5. 비교 모델 성능 비교

표 2는 Food-101 하위 10개 클래스 데이터셋에 대해 Swin-V2 모델과 Swin-V1, EfficientNetB0, ResNet50, MobileNetV2, VGG16을 동일한 학습/검증 분할 설정 하에서 파인튜닝한 결과를 정리한 것이다.

표 2. Swin-V1, EfficientNetB0, ResNet50, MobileNetV2, VGG16 과의 비교

Model	Accuracy	F1 Score(Macro)
Proposed Method	0.926	0.926
Swin-V1	0.905	0.905
EfficientNetB0	0.877	0.878
ResNet50	0.883	0.882
MobileNetV2	0.873	0.873
VGG16	0.865	0.865

먼저, Swin-V2는 모든 비교 모델 중 가장 높은 Accuracy와 macro F1-Score를 달성하였다. 기존 CNN 기반 모델들과 비교했을 때, ResNet50과 EfficientNetB0는 여전히 강력한 성능을 보이지만, Swin-V2는 이들을 약 4~5%p 상회하는 성능을 보여준다. 이는 음식 이미지와 같이 다양한 배경과 조명, 촬영 각도를 가지는 복잡한 장면에서, Swin-V2의 윈도우 기반 자기어텐션 구조가 전역적 문맥과 지역적 특징을 함께 포착하는 데 더 유리하다는 점을 시사한다.

또한 Swin-V1과 Swin-V2의 비교에서, 두 모델 모두 Transformer 기반 구조임에도 불구하고 Swin-V2가 약 2%p 높은 성능을 보였다. 이는 Swin-V2에서 도입된 안정적인 학습을 위한 스케일링 전략과 향상된 상대적 위치 인코딩 기법 등이, 비교적 적은 데이터 개수를 가진 하위 데이터셋 환경에서도 긍정적인 효과를 발휘했다는 간접적인 증거로 해석할 수 있다.

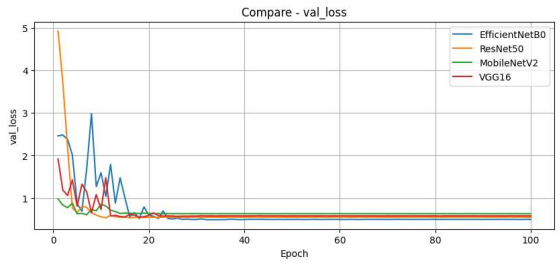

그림 7. EfficientNetB0, ResNet50, MobileNetV2, VGG16간의 Validation Loss 비교

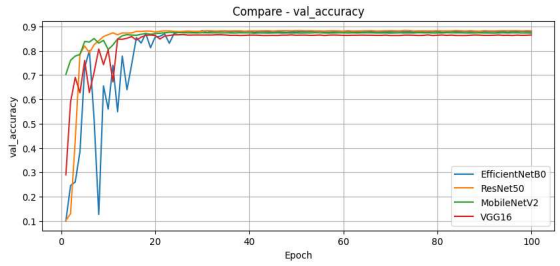

그림 8. EfficientNetB0, ResNet50, MobileNetV2, VGG16간의 Accuracy 비교

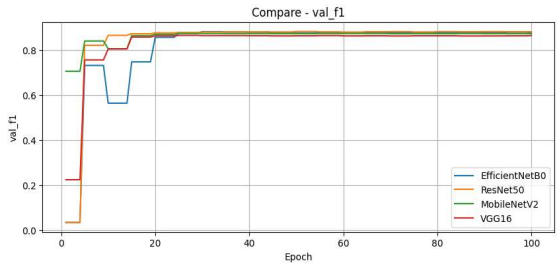

그림 9. EfficientNetB0, ResNet50, MobileNetV2, VGG16간의 F1 Score(macro) 비교

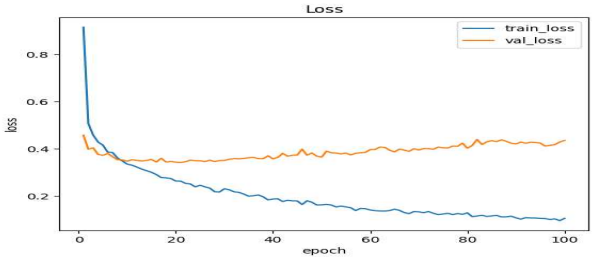

그림 10. Swin-V1 Loss

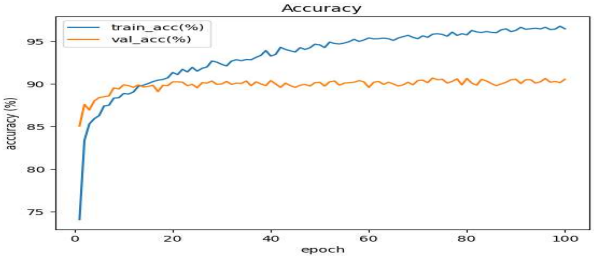

그림 11. Swin-V1 Loss

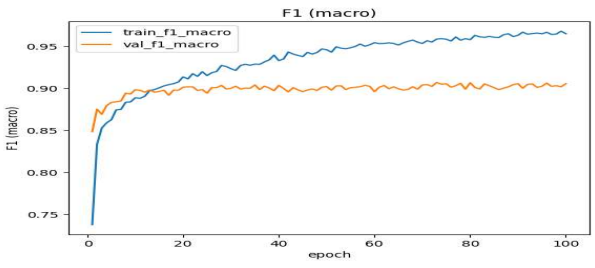

그림 12. Swin-V1 F1 Score(macro)

EfficientNetB0, ResNet50, MobileNetV2, VGG16, Swin-V1의 학습 곡선을 비교한 결과, 공통적으로 학습 정확도는 빠르게 100%에 근접하는 반면, 검증 F1-macro는 약 0.88 수준에서 포화되는 양상을 나타내었다[그림 7, 8, 9, 10, 11, 12]. EfficientNetB0의 경우, 학습 정확도는 초기 epoch에서 99% 수준까지 급격히 증가하는 동안, 검증 F1-macro는 epoch 5에서 0.7325, epoch 20에서 0.8573, epoch 30에서 0.8827로 점진적으로 향상된 이후 약 0.88 부근에서 더 이상 유의미한 개선이 나타나지 않았다. ResNet50 역시 학습 정확도는 거의 100%까지 도달하였으나, 검증 F1-macro는 epoch 5에서 0.8217, epoch 10에서 0.8663까지 빠르게 상승한 뒤, epoch 35에서 0.8820, epoch 50에서 0.8834 수준으로 서서히 수렴하며 추가적인 성능 향상이 거의 발생하지 않는 패턴을 보였다.

MobileNetV2는 경량 구조임에도 불구하고 학습 정확도가 점차 높은 수준까지 수렴하면서, 검증 F1-macro가 초반에 빠르게 상승한 뒤 0.87 - 0.88 전후에서 포화되는 비교적 완만한 곡선을 나타냈다. VGG16은 대규모 파라미터를 바탕으로 학습 정확도가 가장 빠르게 100%에 도달하는 반면, 검증 F1-macro는 다른 모델들과 유사한 0.88 전후에서 수렴하여, 학습·검증 성능 간 격차가 상대적으로 크게 벌어지는 전형적인 과적합 양상을 보였다. Swin-V1의 경우, self-attention 기반 구조 특성상 초기 몇 epoch 동안은 완만한 워밍업 구간을 거친 뒤, 검증 F1-macro가 점차 증가하여 결국 다른 비교 모델과 유사한 0.88 인근에서 수렴하는 안정적인 학습 곡선을 나타냈다.

종합하면, EfficientNetB0, ResNet50, MobileNetV2, VGG16, Swin-V1 모두 동일한 데이터 및 학습 설정 하에서 학습 정확도는 거의 완전한 적합에 도달하지만, 검증 F1-macro는 약 0.88 부근에서 공통적인 일반화 상한에 도달하는 것으로 해석할 수 있다. 이에 비해 제안하는 Swin-V2는 검증 F1-macro가 이러한 상한을 소폭 상회하여 최고 0.9068까지 도달하였으며, 포화 구간에 진입하기 전까지 검증 성능이 지속적으로 증가하는 양상을 보여, 동일한 조건에서 다른 백본들보다 우수한 표현력과 일반화 성능을 제공함을 시사한다.

6. 식단 추천 모듈 사례 분석

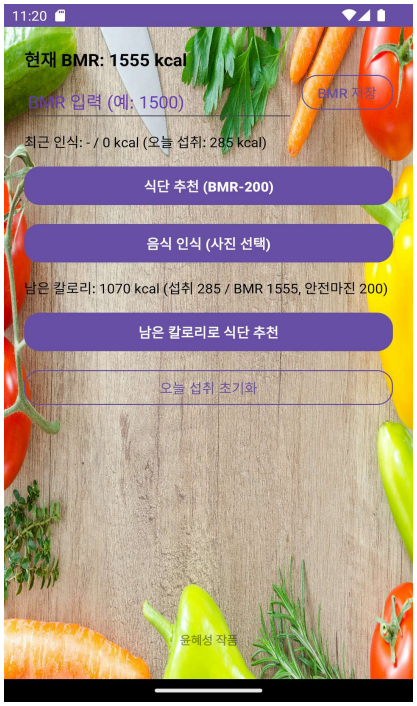

그림 13. 앱 홈 화면

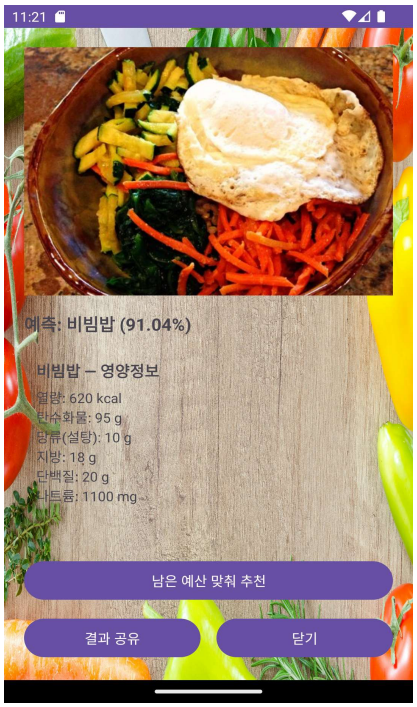

그림 15. 음식 인식 화면

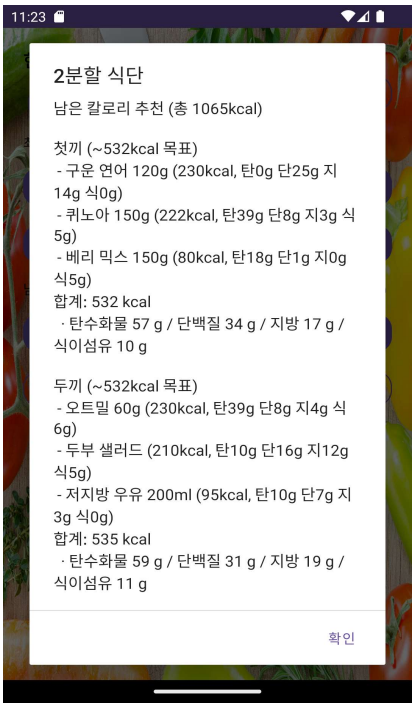

그림 14. 식단 추천 화면

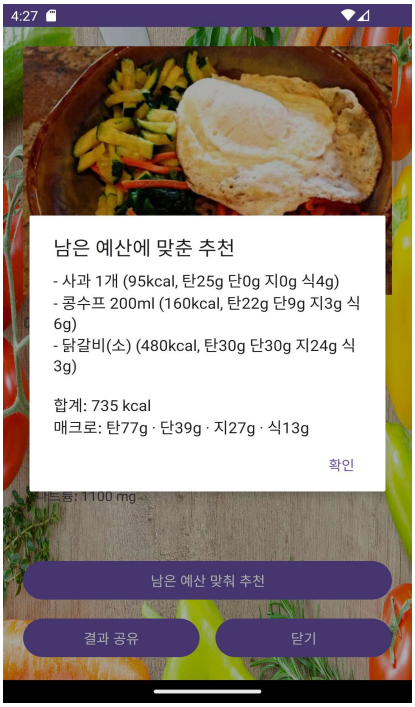

그림 16. 음식 인식 후 남은 예산에 맞춘 식단 추천 화면

표 3. 사례 1: 예시 사용자에 대한 칼로리 예산 설정

항목	값
BMR	1,555kcal
안전 마진	200kcal
남은 예산	1,355kcal
1끼 목표 열량	약 678kcal
추천 식단 총 열량	1,355kcal

표 4. 사례 1: 예시 사용자에 대한 2분할 식단 추천 결과

끼니	음식명	서빙	열량 (kcal)	끼니 합계
첫끼	구운 연어	120g	230	
	통밀빵	2조각	190	
	김밥	1/2줄	260	
	소계			680
두끼	두부 샐러드	1그릇	210	
	코티지 치즈	100g	110	
	토마토 파스타	200g	360	
	소계			680
	총합			1,360

표 5. 사례 2: 예시 사용자에 대한 칼로리 설정

항목	값
BMR	1,555kcal
이미 섭취한 열량	290kcal
안전 마진	200kcal
남은 예산	1,065kcal
추천 식단 총 열량	1,065kcal

표 6. 사례 2: 예시 사용자에 대한 남은 예산 기반 식단 추천 결과

음식명	서빙	열량(kcal)
샐러드	1그릇	약 80
고구마 중간	1개	약 180
오트밀 + 우유	오트밀 40g, 우유 한 컵	약 220
현미밥	1공기	약 300
두부 반모	150g	약 120
닭가슴살	100g	약 165
합계	1,065	

본 절에서는 제안한 식단 추천 모듈이 실제 사용자 설정에 대해 어떻게 동작하는지 확인하기 위해, 가상의 사용자 시나리오에 대한 사례 분석을 수행하였다.

사례 1에서는 BMR이 1,555kcal인 사용자가 아직 아무것도 섭취하지 않은 상태에서 안전 마진 200 kcal를 적용하면, 남은 칼로리 예산은 1,355kcal가 된다. 제안한 알고리즘은 이를 두 끼로 균등 분할하여 한 끼당 약 677kcal를 목표로 식품 조합을 탐색하고, 그 결과 [표 3]과 같은 2분할 식단을 추천하였다. 첫 끼와 두 끼의 합계는 각각 680kcal 그리고 전체 추천 열량은 1,360kcal로 설정한 목표 예산(1,355kcal)에 근접한 값을 보인다. 이 결과는, 677kcal라는 목표에 오차 3으로 가장 근접한 680kcal가 가장 목표에 가깝다고 판단하여 최상위 후보가 된 까닭이다.

사례 2에서는 사용자가 하루 동안 이미 일정량의 에너지를 섭취한 상태에서, 남은 칼로리 예산 E_{target}에 맞추어 식단을 추천하는 상황을 가정하였다. 예를 들어, BMR과 누적 섭취 열량, 안전 마진을 반영한 결과 남은 예산이 1,065kcal로 계산된 경우, 제안된 식단 추천 모듈은 [표 6]과 같이 샐러드, 고구마, 오트밀, 현미밥, 두부, 닭가슴살로 구성된 단일 식단을 제안하였다. 각 식품의 열량을 합산한 결과, 총 열량은 1,065kcal로 목표 예산과 일치함을 확인할 수 있다.

[그림 16]을 통해 Swin-V2 모델의 음식 인식 결과에 따른 식단 추천 역시 총 열량은 735kcal로 목표 예산과 일치함을 확인 가능하다.

V. 결 론

연구에서는 현대인의 건강한 식생활 관리를 돕기 위한 자동 음식 인식 시스템의 핵심 기술로서 Swin Transformer 모델 적용 가능성을 탐구했다. 대규모 데이터셋으로 학습된 Swin-V2-Tiny 모델을 Food-101 데이터셋에 파인튜닝하여, 100 에폭 학습한 결과만으로 0.926의 높은 Top-1 분류 정확도, 0.926의 F1 Score를 달성하였다. 이는 Swin Transformer의 효율적인 아키텍처가 복잡하고 다양한 음식 이미지를 분류하는 데 매우 효과적임을 입증하는 결과이다.

뿐만 아니라, Swin-V2 분류 모델과 연계한 식단 추천 모듈 역시 데이터베이스에 등록된 제한된 후보 식품 집합에서 최적의 식단 조합을 생성하는 것을

확인할 수 있었다.

본 연구는 음식 종류를 성공적으로 분류했지만, 실제 영양 관리 애플리케이션으로 발전하기 위해서는 다음과 같은 후속 연구가 필요하다. 첫째, 음식의 양(portion)을 추정하는 기술이 결합되어야 한다. 객체 탐지(Object Detection)나 인스턴스 분할(Instance Segmentation) 모델을 활용하여 이미지 내에서 음식 영역을 정확히 분리하고, 뎁스 카메라나 다중 시점 이미지를 통해 부피를 추정하는 연구가 필요하다. 둘째, 제한된 연산 자원을 가진 모바일 환경에서의 실시간 구동을 위해 모델 경량화 및 최적화 연구가 병행되어야 한다. 셋째, 데이터베이스에 등록한 후보 식품 집합을 다양하게 구성할 필요성이 있다.

이러한 과제들이 해결된다면, 본 연구에서 제안한 고성능 음식 인식 모델은 사용자가 손쉽게 식단을 관리하고 건강한 삶을 영위하는 데 기여하는 강력한 도구가 될 수 있을 것이다.

REFERENCES

[1] World Health Organization, “Obesity and overweight,” WHO Fact sheet, May 7, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight. (accessed Sep., 8, 2025).

[2] International Diabetes Federation. (n.d.). Diabetes facts and figures. https://idf.org/about-diabetes/diabetes-facts-figures/. (accessed Sep., 8, 2025).

[3] DI ANGELANTONIO, Emanuele, et al. Body-mass index and all-cause mortality: individual-participant-data meta-analysis of 239 prospective studies in four continents. The lancet, 388.10046:, pp. 776-786, 2016.

[4] LAUBY-SECRETAN, Béatrice, et al. Body fatness and cancer—viewpoint of the IARC Working Group. *New England journal of medicine*, 375.8:, pp. 794-798, 2016.

[5] SALIM, Nareen OM, et al. Study for food recognition system using deep learning. In: *Journal of Physics: Conference Series*. IOP Publishing, p. 012014, 2021.

[6] CHOPRA, Megha; PURWAR, Archana. Recent studies on segmentation techniques for food recognition: A survey. *Archives of Computational Methods in Engineering*, 29.2:, pp. 865-878, 2022.

[7] KAGAYA, Hokuto; AIZAWA, Kiyoharu; OGAWA, Makoto. Food detection and recognition using convolutional neural network. In: *Proceedings of the 22nd ACM international conference on Multimedia,* pp. 1085-1088, 2014.

[8] POULADZADEH, Parisa; SHIRMOHAMMADI, Shervin. Mobile multi-food recognition using deep learning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13.3s:, pp. 1-21, 2017.

[9] MARTINEL, Niki; FORESTI, Gian Luca; MICHELONI, Christian. Wide-slice residual networks for food recognition. In: *2018 IEEE Winter conference on applications of computer vision (WACV)*. IEEE, pp. 567-576, 2018.

[10] ZAHISHAM, Zharfan; LEE, Chin Poo; LIM, Kian Ming. Food recognition with resnet-50. In: *2020 IEEE 2nd international conference on artificial intelligence in engineering and technology (IICAIET)*. IEEE, pp. 1-5, 2020.

[11] PHIPHIPHATPHAISIT, Sirawan; SURINTA, Olarik. Food image classification with improved MobileNet architecture and data augmentation. In: *Proceedings of the 3rd International Conference on Information Science and Systems.*, pp. 51-56, 2020.

[12] PARVATHAVARTHINI, S., et al. A Deep Learning Approach for Food Recognition and Nutritional Analysis. In: *2025 8th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, pp. 1010-1017, 2025.

[13] DOSOVITSKIY, Alexey, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[14] LIU, Ze, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF international conference on computer vision,* pp. 10012-10022, *2021*.

[15] LIU, Ze, et al. Swin transformer v2: Scaling up capacity and resolution. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* pp. 12009-12019, *2022*.

[16] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems,* 25, *2012*

[17] SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

저 자 소 개

윤혜성(준회원)

2026년 호남대학교 컴퓨터공학과 학사 졸업

<주관심분야 : 딥러닝, 컴퓨터 비전>