

2025 오픈소스프로그래밍 프로젝트 최종보고서

프로젝트명	시각장애인을 위한 편의점 과자 인식 프로그램			
팀명	바나나킥	팀장	학번	이름
			2410470	김경민
수행기간	2025. 05. 15 ~ 2025. 06. 22 (학기말까지)	팀원	2410310	김가은
			2416892	심세희

가. 목표 및 기대효과

- 기술개발 배경
 - 시각장애인들은 일상생활에서 과자류 제품의 종류나 브랜드를 식별하기 어려워 문제를 겪고있다.
 - 현재 점자 표기나 음성안내 서비스는 일부 대형 매장에만 한정되며, 실제 편의점과 같은 소형매장에서는 보조 장비 없이 제품 정보 접근이 매우 제한적이다.
 - 과자 회사가 점자를 표기해야한다는 규정도 없을 뿐더러 과자회사의 입장도 고려를 해야하므로 점자를 무조건적으로 표시하게 할 수는 없다.
 - 이를 해결 또는 해소하기 위하여 AI 기술이 효과적인 기술적 대안이 될 수 있다.
- 기술개발 목표
 - 시각장애인이 편의점 내에서 독립적으로 제품을 식별하고 구매 결정을 내릴 수 있도록 보조 기기를 활용하여 과자류 제품의 이미지를 인식하고, 이를 기반으로 제품명에 대한 정보를 음성으로 안내하는 시스템을 개발하고자 한다.
- 기대효과
 - [기술적 기대효과]**
 - 이미지 인식 및 음성 변환 기술을 통합한 보조 시스템 개발을 통한 실시간 제품 인식 및 음성 안내 솔루션 확보
 - 시각장애인을 위한 실사용 환경 중심의 기술 구현 사례 확보
 - 응용 가능성이 높은 확장성 있는 모델 개발 (향후 다양한 상품 약, 화장품 등으로 확장 가능)
 - [사회적 기대효과]**
 - 시각 장애인의 정보 접근성 향상을 통한 소비 자율성, 삶의 질 향상에 기여함으로써 사회적 통합 및 기술기반 복지 구현을 도움
 - 일상적인 식품 선택, 구매 과정에 대한 불편해소를 통한 사회참여 및 독립적 생활 지원
 - 비장애인 중심으로 설계된 소매 유통 환경에 대한 포용적 개선 방향 제시

나. 프로젝트 개발 내용

< 개발 과정 >

- 데이터 전처리
 - Roboflow를 통해 과자 구별을 위한 과자 이미지 데이터셋을 새롭게 만들었다.
 - 인터넷 후기를 통하여 이미지를 수집하고 Roboflow에서 수동으로 바운딩 박스를 라벨링하였다.
 - 데이터 수량 : 403장(이미지 수), 99(클래스 수)
 - 전처리 후 : 965장

Dataset Details

965 Total Images

[View All Images →](#)



Dataset Split

TRAIN SET

87%

843 Images

VALID SET

8%

82 Images

TEST SET

4%

40 Images

Preprocessing

Auto-Orient: Applied

Resize: Stretch to 640x640

Augmentations

Outputs per training example: 3

Flip: Horizontal, Vertical

Rotation: Between -14° and $+14^\circ$

Saturation: Between -32% and $+32\%$

Exposure: Between -10% and $+10\%$

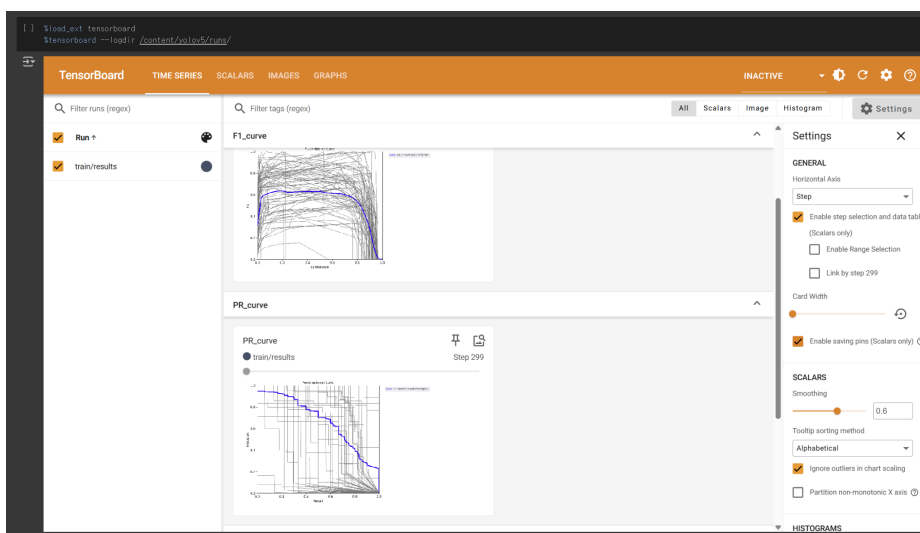
Blur: Up to 2.7px

Noise: Up to 1.92% of pixels

YOLOv5 모델 학습시키기

앞서 만든 데이터셋을 사용하여 Google colab에서 YOLOv5를 과자 탐지 모델로 학습시켰다.

1. 구글 코랩 환경으로 데이터셋 다운로드
2. YOLOv5 환경 세팅하기
3. YOLOv5 학습하기
4. 최종 학습된 모델 저장하기 (best.pt)

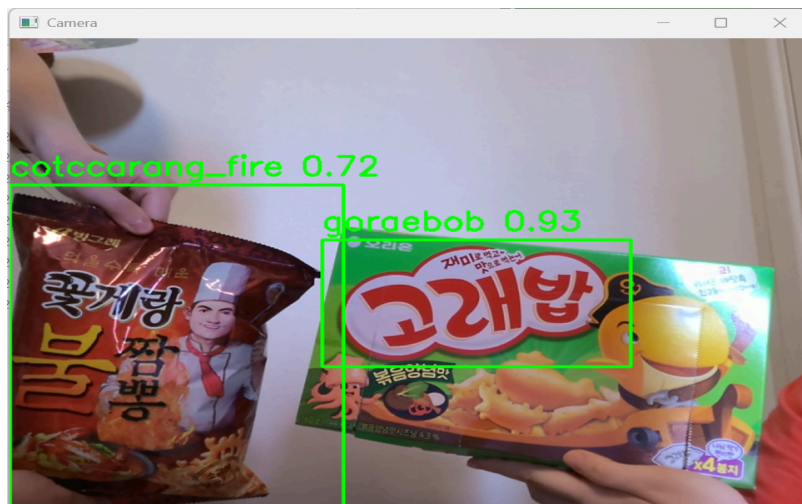


<tensorboard로 학습결과 상세하게 확인하기>

주요 기능 구현

기능 1: 실시간 과자 인식

- 노트북 내장 카메라를 통해 실시간으로 프레임을 받아옴
- 학습된 YOLOv5 모델(best.pt)을 통해 받아온 이미지 내 과자를 탐지 및 분류
- 프레임 상에 탐지된 과자 객체에 네모 표시 및 텍스트 라벨링 표시

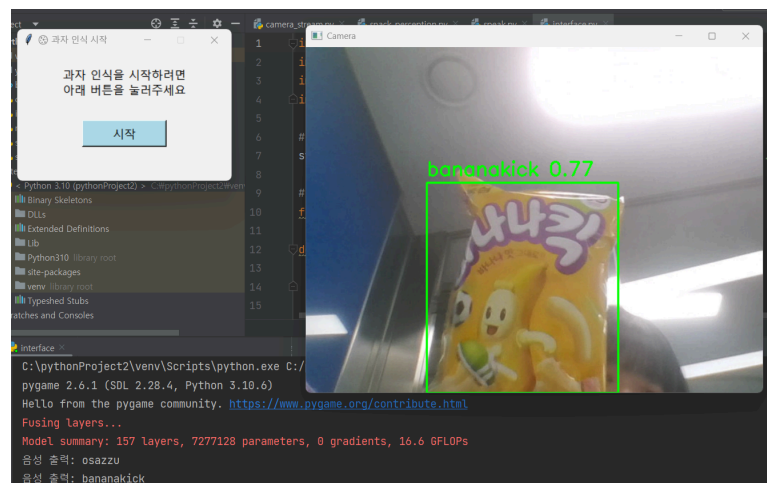


기능 2: TTS 음성 안내 기능

- 인식된 과자 이름을 **gTTS(Google Text-to-Speech)**를 이용해 음성으로 변환 후 출력 (ex “이 제품은 고구마깡입니다.”, “이 제품은 바나나킥입니다.”)
- 동일 과자 반복 인식을 방지하기 위해 이전 인식 과자와 비교 후 음성 출력 여부 결정
- **pygame**을 이용해 음성 재생 후 임시 음성 파일 자동 삭제

기능 3: 인터페이스 기능

- GUI 실행을 위한 인터페이스 기능을 추가 개발하여 사용자가 버튼 클릭만으로 인식 기능 프로그램을 손쉽게 시작할 수 있도록 하였음



(시작 버튼을 누르면 노트북 내장 카메라가 켜지며 과자 인식 프로그램이 시작된다.)

- 개발 서비스 목표 및 기능 구현 세부 내용

해당 프로그램의 최종 목표는 시각장애인이 스스로 과자 종류를 구별할 수 있도록 돕는 것이다. 이를 위해 인공지능 기반 객체 탐지와 음성 안내 기능을 결합한 직관적이고 자동화된 인식 시스템을 구현하였다. 실제 구현은 다음과 같은 알고리즘 중심 구조를 따른다.

1. 실시간 영상 입력 처리

- OpenCV의 `cv2.VideoCapture(0)`을 사용하여 노트북 내장 카메라에서 프레임 단위로 영상을 수신한다.
- 프레임은 YOLOv5 모델이 요구하는 포맷으로 `letterbox`, `transpose`, `tensor` 변환 과정을 거친다.

```
33 def get_camera_frame():
34     cap = cv2.VideoCapture(0) # 0번 장치를 통해 노트북 카메라 켜기
35     if not cap.isOpened():
36         print("카메라 연결 실패")
37     return
```

(snack_perception.py의 일부 코드 예시)

2. 딥러닝 기반 객체 인식

- `DetectMultiBackend` 클래스를 사용하여 `best.pt` 모델 가중치를 불러오고, 프레임마다 과자 객체를 탐지한다.
- YOLOv5의 `non_max_suppression()`을 통해 다중 탐지를 정리하고, 클래스별로 `confidence`가 가장 높은 탐지 결과만 추출한다.
- 탐지 결과는 `cv2.rectangle()` 및 `cv2.putText()`를 통해 화면에 실시간 표시된다.

```
# 3. 후처리 및 시각화
for det in pred:
    if len(det):
        det = filter_highest_confidence_per_class(det) # 클래스별 confidence가 가장 높은 것만 남기기
        det[:, :4] = scale_boxes(img.shape[2:], det[:, :4], frame.shape).round()
        for *xyxy, conf, cls in reversed(det):
            label = f'{names[int(cls)]} {conf:.2f}'
            x1, y1, x2, y2 = map(int, xyxy)
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(frame, label, (x1, y1-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
            snack_name = names[int(cls)] # 마지막으로 인식된 과자 이름
```

(snack_perception.py의 일부 코드 예시)

3. TTS(Text-to-Speech) 음성 안내

- 탐지된 과자명이 이전 프레임과 다를 경우에만 음성 출력이 작동하도록 하여 반복 방지.
- gTTS로 음성 mp3 파일을 생성하고, `pygame.mixer`로 재생 후 자동 삭제되도록 처리하였다.
- 파일 접근 충돌 방지를 위해 `uuid`를 활용한 고유한 mp3 파일명을 사용하였다.

```
24 # 명시적으로 언로드 및 초기화 후 삭제
25 pygame.mixer.music.stop()
26 pygame.mixer.music.unload()
27 pygame.mixer.quit() # mixer 완전 해제
```

(speak.py의 일부 코드 예시)

4. 최종 사용자 인터페이스

- `tkinter` 기반으로 GUI 버튼을 제공하여 사용자가 별도 명령어 입력 없이 인식 기능을 손쉽게 실행할 수 있도록 하였다.

- GUI 이벤트 루프를 유지하기 위해 `threading.Thread`로 인식 함수(`get_camera_frame()`)를 병렬 실행시켰다.

```
1 import tkinter as tk
2     import threading
3     import sys
4     import os
```

(interface.py의 일부 코드 예시)

→ 이와 같이 각 기능은 모듈화되어 있으며, 최종 사용자는 GUI 상의 버튼 하나만으로 영상 입력, 과자 인식, 음성 출력까지 모든 과정을 자동으로 손쉽게 편리하게 실행할 수 있다.

- 사용된 SW 기술

1. 개발 환경 및 플랫폼

- 학습 플랫폼: Roboflow + Google Colab 환경에서 데이터셋 정리 및 모델 학습 수행 (YOLOv5 모델 fine-tuning)
- 개발 환경: Visual Studio Code + GitHub

2. 주요 기술 및 라이브러리

- 딥러닝 모델: YOLOv5 (DetectMultiBackend 사용, CPU 기반 실행)
- 영상 처리: OpenCV (cv2)를 통해 실시간 카메라 프레임 처리 및 시각화
- 음성 안내 기능: gTTS를 통해 텍스트를 음성으로 변환, pygame으로 mp3 재생 후 삭제
- GUI 인터페이스: tkinter 라이브러리로 간단한 버튼 기반 인터페이스 제공 (interface.py)
 - 쓰레드를 활용하여 GUI 이벤트 루프를 멈추지 않고 인식 기능 실행 가능하게 함.

- 데이터 수집 및 모델 학습

- 데이터 수집 : 약 100종의 대표적인 편의점 과자 이미지 403장을 수집하여 라벨링하였음.
- 라벨링 도구: Roboflow 를 활용해 라벨링 및 데이터 증강
- 학습 모델: YOLOv5
- 학습 결과: confidence 기반 필터링 처리를 통해 높은 정확도로 각 과자 분류가 가능해짐.
- cf) 바운딩 박스가 여러 개일 경우, 가장 확신(confidence)이 높은 클래스 하나만 선택하여 출력할 수 있도록 하였음.

- 시스템 구성도

사용자 → (카메라) → OpenCV → YOLOv5 모델 추론 → Snack Name 추출 → gTTS + pygame → 음성안내

- 에러 해결 및 기술 이슈 대응

- playsound 라이브러리 오류:
Windows 환경에서 재생 실패 오류 발생 (Error 259, Error 263)

→ pygame 으로 대체하여 해당 문제를 해결하였음

- **TTS** 재생 중 파일 접근 오류:

파일 삭제 시점 조절 및 고유한 임시 파일 이름 (uuid) 생성으로 충돌 방지

- **YOLOv5** 모듈 import 경로 오류:

sys.path.append() 를 사용하여 상대경로 기반으로 내부 모듈 호출

- **GitHub** 활용 방식

- 버전 관리: **GitHub** 저장소를 통해 각자 코드 버전 관리
- 파일 구조 설계

Snack-Perception/

├─ yolov5/ # YOLOv5 모델 코드 (git clone한 yolov5)

├─ best.pt # 커스텀 학습된 과자 인식 모델

├─ snack_perception.py # 메인 실행 파일

├─ speak.py # 텍스트를 음성으로 출력하는 모듈

└─ README.md # 프로젝트 설명 문서

(yolov5 디렉토리는 YOLOv5 공식 저장소를 클론해서 사용 - git clone <https://github.com/ultralytics/yolov5.git>)

- 협업 방식

- 각자 담당 기능을 로컬에서 개발하고, 기능 구현이 완료되면 **GitHub** 에 푸시
- **README** 파일 작성 및 주석 정리를 통해 협업 가독성 강화
- 기능별로 파일을 모듈화하여 분리하고, **GitHub**를 통해 지속적으로 기능별 코드 상황을 업데이트

- **snack_perception.py**: 인식 및 음성 안내 메인 코드

- **speak.py**: 음성 변환 및 재생 담당 모듈

- **interface.py**: tkinter 기반 GUI 실행 인터페이스

- **README.md**: 각 코드 소개 및 실행 방법 기술

- 파일 업로드 시 **commit** 메세지 명확화를 통해 팀원 간 이해도 향상 도움.

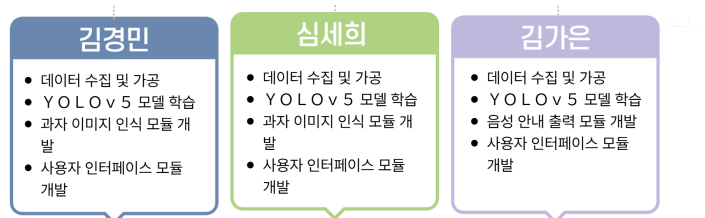
- 실습 정리 및 제출 활용

- 최종 구현된 코드와 실행 방법을 **README.md** 에 정리하여 팀원들뿐만 아니라 외부 사용자도 바로 실행 가능한 형태로 문서화하였음.

- 기획 보고서 및 최종 보고서 제출 전, **GitHub** 저장소 기준으로 파일 정리 및 기록을 완료함.

다. 추진일정 및 업무분담

1) 역할 분담 계획



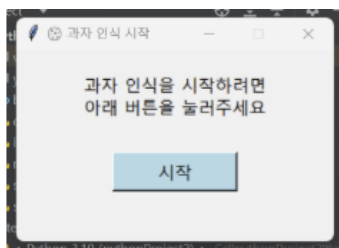
구분	성명	담당 업무	세부개발 사항
팀장	김경민	데이터 수집 및 가공, YOLOv5 모델 학습, 과자 이미지 인식 모듈 개발	과자 인식 모듈 개발에서 프레임을 학습시킨 모델에 입력하여 추론, 추론 결과에서 탐지된 객체의 위치, 이름 추출, bounding box 및 label 시각화 처리
팀원-1	김가은	데이터 수집 및 가공, YOLOv5 모델 학습, 음성 안내 출력 모듈 개발, 사용자 인터페이스 모듈 개발	객체 탐지 결과를 실시간으로 받아 음성으로 안내하는 기능 구현, TTS 기술을 활용하여 자연스러운 한국어 음성으로 출력하도록 처리, 사용자의 간단한 조작 가능하게 하는 ui 구현
팀원-2	심세희	데이터 수집 및 가공, YOLOv5 모델 학습, 과자 이미지 인식 모듈 개발	과자 인식 모듈 개발에서 웹캠으로부터 실시간 영상 스트림 수신 코드 구현, YOLOv5에 전달할 수 있는 형식으로 이미지 전처리, TTS 와 연결

2) 일정

세부 개발내용	과제 수행 기간					
	5월 2주	5월 3주	5월 4주	6월 1주	6월 2주	6월 3주
1. 주제선정 및 역할분담						
2. 데이터 수집 및 전처리						
3. YOLOv5 모델 학습						
4. 과자 이미지 인식 모듈 개발						
5. 음성 안내 출력 모듈 개발						
6. 사용자 인터페이스 모듈 개발						
7. 동영상 촬영, 최종 보고서 작성						

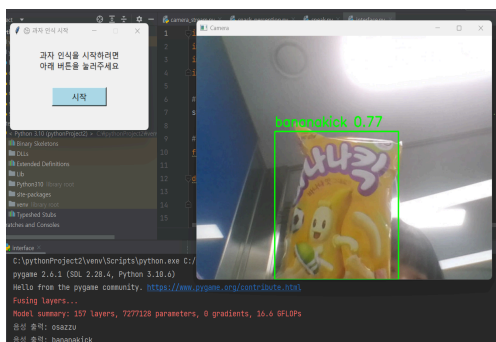
다. 프로젝트 개발 결과

- 최종결과물 내용

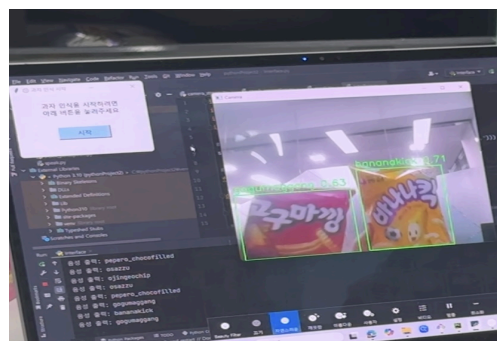


◦결과물 예시

→ `interface.py` 실행 → 시작 버튼 누르기
→ 사용자가 카메라에 과자 비추기



(이미지1: 바나나킥 인식 시 모습)



(이미지2: 두 과자를 동시에 비쳤을 때)

이미지1: → “이 제품은 바나나칩입니다.” 음성 안내 출력됨

이미지2: → “이

제품은 바나나칩입니다. 이 제품은 고구마깡입니다.” 음성 안내 출력 + q 누를 시 종료

<잘 되지 않는 경우>

맛동산, 고래밥, 빅파이, 고구마깡와 같이 특정 과자들은 잘 인식하지만 빼빼로, 칸초는 잘 인식하지 못한다는 문제가 있다. 모든 물체를 과자로 인식한다는 문제가 있다.

◦Github 사이트

- 25-1 오픈소스프로그래밍 16조 <https://github.com/25-1-16/project>

• 프로젝트 수행 결과의 한계점

◦개발 기술의 한계점

사전에 학습한 제품 외에는 인식하지 못하며, 새로 출시된 제품은 별도의 학습이 필요함. 또한 같은 제품도 포장 디자인이 주기적으로 바뀌거나 한정판이 있는 경우. 미등록된 이미지로 인해 인식에 문제가 발생함. 카메라 실행 시 렉이 걸리고 작동 속도가 느려 보여준 과자를 바로바로 인식하지는 못하는 문제가 발생함. 또한 인식 후 음성 안내까지 시간이 오래걸림. 시각 장애인이 직접 들고 사용하는 경우, 제품이 카메라에 정확히 잡히지 않아 인식이 실패할 가능성이 있음. 과자 여러개를 인식시켰을 때 인식 시간이 오래걸리고 정확도가 낮아지는 문제가 있음.

◦개선 방향

- 데이터셋 보강 및 클래스

불균형 해소

현재 인식이 잘 되지 않는 빼빼로, 칸초 등은

학습 이미지 수가 부족하거나 다양한 각도/조명/배경에서의 샘플이 적을 가능성이 큼. → 추가 촬영 및 증강(Data Augmentation)을 통해 해당 과자 클래스의 데이터를 보강하고, 전체 클래스 간의 데이터 균형을 맞춰 인식률을 개선할 수 있습니다.

- Class 수 제한 및 Background 클래스 도입 고려

학습 시 '과자 외

배경 클래스(background)'를 명시적으로 학습시키지 않아, 모든 물체를 과자로 인식하게 되는 문제가 발생할 수 있습니다. → 향후 학습 시 배경 또는 기타 물체 클래스를 포함하거나, 명시적 negative sample(과자가 아닌 물체 사진)을 포함하여 과자 이외 물체를 구분하는 능력을 강화할 수 있습니다.

◦프로젝트 수행의 어려움 및 느낀 점

[프로젝트 수행의 어려움]

모델 학습을 위한 데이터 수집의 어려움이 있었음. 팀원 모두가 데이터 전처리 작업이 처음이다 보니 과자 이미지를 수집하고 전처리 방법을 익히는 데 많은 시간이 소요되었음. 이미지 클래스를 만드는 과정에서도 시행착오가 발생했음. 음성 안내 모듈, 인식 시스템, 사용자 인터페이스 간의 연동이 복잡했음. 팀원들이 각자 역할을 나누어 모듈을 개발한 후 이를 통합하는 과정에서, 변수명 불일치나 모듈 간 동기화 오류가 발생하여 수정에 시간이 걸렸음. 또한 각자 다른 환경에서 코드를 실행하면서 서로 다른 오류가 발생했고, Git을 통한 협업 과정에서 버전 충돌이나 경로 문제를 해결하는 데에도 어려움이 있었음.

[느낀 점]

가은: 이번 프로젝트를 하면서 git을 활용한 협업의 중요성과 실제 적용의 어려움을 느낄 수 있었음. 각자 코드 작업 후 github에 업로드하고 병합하는 과정에서 충돌이 자주 발생했고 이를 해결한 뒤에 뿌듯함을 느꼈음. 특히 각자의 로컬환경에서 발생하는 오류가 달라 이를 공유하고 함께 해결해가는 과정에서 팀워크의 중요함을 느꼈음. 또한 팀원들과 협력해서 모델을 학습시킨 뒤 인터페이스와 연결하고 음성안내까지 연동함으로써 사람이 실제로 사용할 수 있는 형태로 만든 경험을 하면서 뿌듯함을 느꼈음.

경민: 딥러닝 모델을 학습하여 시각장애인을 위한 서비스를 만들면서 인공지능 기술이 실제 생활에 얼마나 큰 도움이 될 수 있는지 생각해보게 되었습니다. 수업시간에 배운 데이터셋 생성과 딥러닝 모델을 학습시키는 과정을 이번 프로젝트를 통해 직접 해보며 어떻게 구현되고 활용되는지 경험할 수 있었습니다. 프로그램을 실제로 테스트해서 과자 종류를 구별하는 결과를 확인할 때마다 성취감을 느낄 수 있었습니다. Github를 활용하여 팀원들과 협업하며 협력능력과 문제해결력을 기를 수 있었습니다.

세희: 이번 프로젝트를 통해 딥러닝 모델을 실제 서비스로 구현하는 전 과정을 처음부터 끝까지 경험할 수 있어 굉장히 뜻깊은 시간이었다. 단순히 모델을 학습시키는 것을 넘어, '카메라 입력-> 모델 추론-> 결과 해석-> 음성 출력'까지 하나의 흐름으로 이어지는 시스템을 구축해보며, 기술을 현실 문제 해결에 연결하는 법을 배울 수 있었다. 무엇보다도, 팀 프로젝트를 진행하며 효율적인 GitHub 활용법과 협업 능력을 기를 수 있었던 거 같다. 끝으로, 우리가 만든 서비스가 시각장애인의 일상 속 불편함을 조금이나마 해소할 수 있는 방향으로 설계되었다는 점에서 보람을 느낀다.