

Hear Eat

**시각장애인의 편의점 이용 접근성을
높이는 애플리케이션**

4조

김민성, 박준서, 원현선, 임소현

01

개발동기

02

모델 선택 및 학습

03

앱 소개

04

기대 효과 및 Q&A

편의점 과자 점자 미표기

개발 배경



- 시중에 판매되는 과자에 점자 표기 X
- 포장 용기가 다르지 않다면 시각장애인이 구별 불가

유사한 앱의 문제점

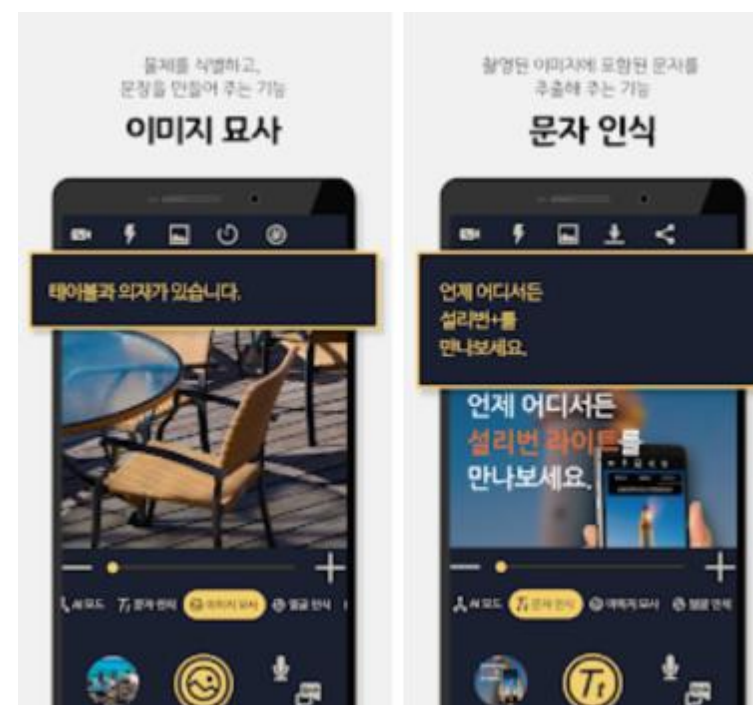
Be My Eyes



영상통화 형식의 애플리케이션

- 과자를 고르는 시간 동안 통화 필요
- 많은 옵션들을 통화 상대가 읽어야 함

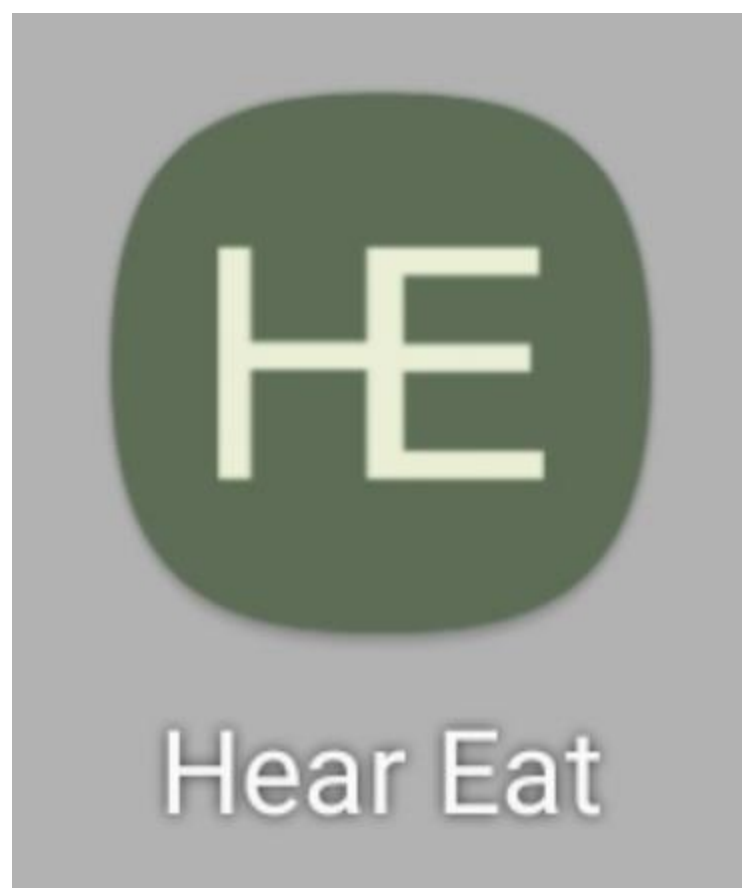
설리번 플러스



객체인식 형태의 애플리케이션

- 이미지 묘사로는 과자 종류를 구별할 수 없음
- 문자 인식으로는 포장지 위에 글자가 너무 많고 폰트 인식이 어려움

Hear Eat 앱 소개



시각장애인이 편의점을 이용할 때 사용할 수 있는 애플리케이션

- 앱을 켜면 바로 카메라 화면이 보이고, 화면 위로 과자들이 인식됨
- 앱 화면을 아무 곳이나 터치하면 음성으로 과자의 이름을 읽어 줌
- 과자 여러 개를 동시에 인식하여 과자를 고를 때 실시간으로 사용 가능

Faster R-CNN

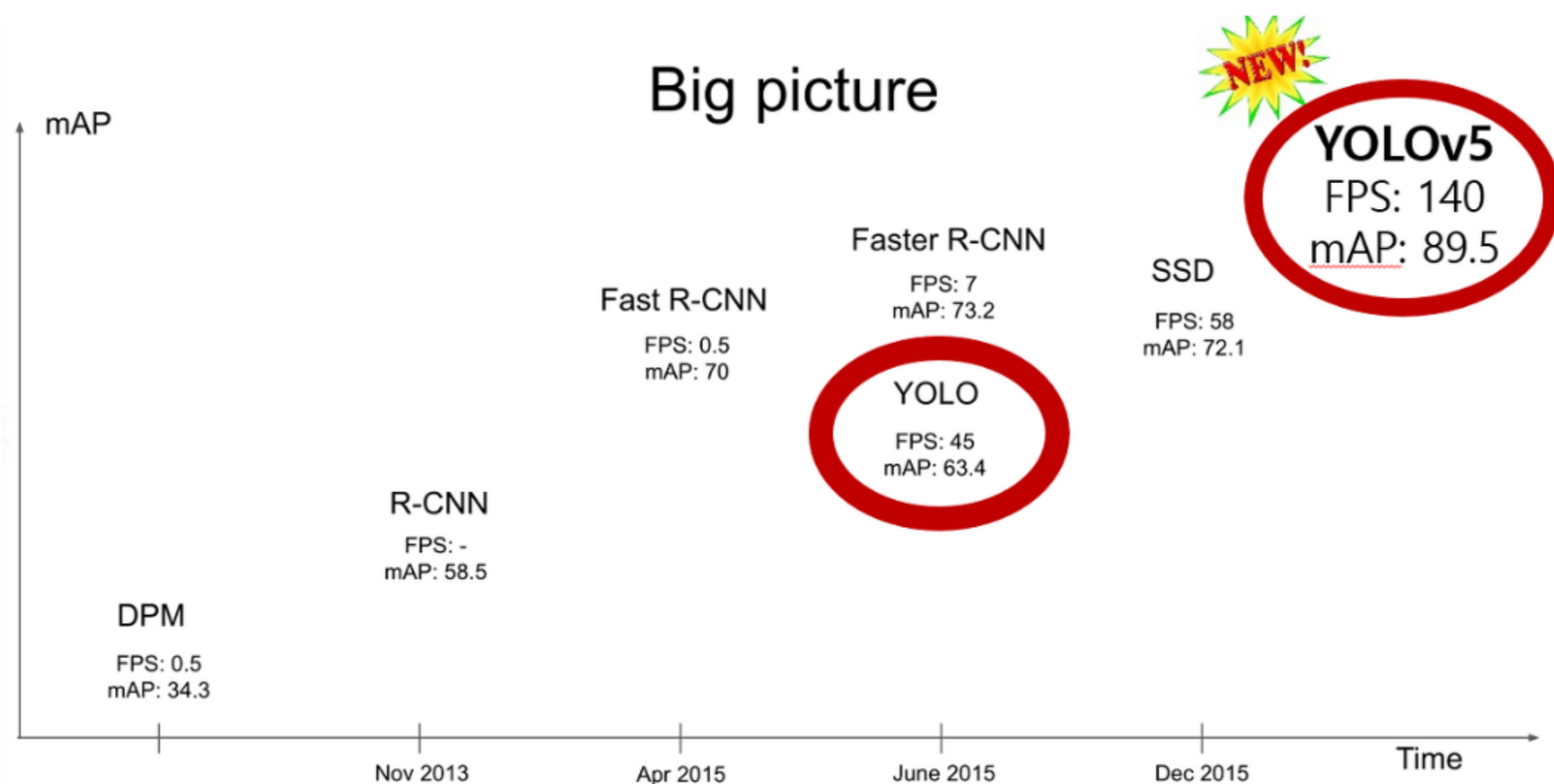
- + 높은 탐지 정확도. 작은 물체에 대해서도 탐지력 우수
- 긴 모델 학습 시간. 객체 탐지 속도가 느림.

SSD(single shot multi box detector)

- + Faster R-CNN에 비해 비교적 빠르고 정확함
- 복잡한 사용성, 짧은 기간 내에 숙련되기 어려움

YOLOv5

You Only Look Once



적은 학습 시간, 간편한 사용법

준수한 객체 탐지 성능

실시간 탐지 성능은 매우 좋음

서울시 상품 표지 이미지 AI 학습 데이터셋

JPG 서울특별시_상품 표지 이미지 AI 학습 데이터셋

데이터 수집자가 소유한 스마트폰 활용하여 편의점, 마트의 카테고리별 상품의 영상데이터 확보하였습니다.

마트, 편의점에서 촬영한 영상을 이미지 프레임 추출 후 항목, 식품 명칭별 클래스를 지정하여 Bounding Box(구역표시)

※ 해상도 : 1920×1080

시중에 판매되는 상품 표지 이미지의 학습 데이터를 활용한 공공, 민간 서비스 연계, 상품 이미지 데이터 POOL 구축에 대한 가이드라인 활용할 수 있습니다.

[바로가기](#)
[오류신고 및](#)
[담당자 문의](#)


0



0

[관심](#)


팀원 개인별 편의점 동영상 촬영 후 라벨링 **roboflow**



최종 학습 데이터

클래스 : 83개

Train : 1664

Valid : 474

Test : 240

+

Generate New Version

VERSIONS

Final2

v6 Jul 18, 2022

2022-07-18 11:36pm

v5 Jul 18, 2022

Final1

v4 Jul 18, 2022

2022-07-18 11:27pm

v2 Jul 18, 2022

2022-07-18 11:24pm

v1 Jul 18, 2022

Final2

Version 6 Generated Jul 18, 2022

Export Edit

TRAINING OPTIONS


Use Roboflow Train

Let us train your model and get results within 24 hours along with a hosted API endpoint for making predictions. [Learn More >](#)

Start Training

Available Credits: 3

IMAGES



2378 images

[View All Images >](#)

TRAIN / TEST SPLIT

Training Set

70%

1.7k images

Validation Set

20%

474 images

Testing Set

10%



240 images

08


<https://app.roboflow.com/productdetection-3rxyc/final-5jluu/images/>

08


```
!python /content/yolov5/train.py --img 640 --batch 32 --epochs 300 --data /content/yolov5/Final-6/data.yaml --weights yolov5s.pt
```

train: weights=yolov5s.pt, cfg=, data=/content/yolov5/Final-6/data.yaml, hyp=yolov5/data/hyps/hyp.scratch-low.yaml, epochs=300, batch_size=32, imgsz=640, rect=False, resum
github: up to date with <https://github.com/ultralytics/yolov5> 
YOLOv5  v6.1-316-g916bdb1 Python-3.7.13 torch-1.12.0+cu113 CUDA:0 (Tesla T4, 15110MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, obj=

Weights & Biases: run 'pip install wandb' to automatically track and visualize YOLOv5  runs (RECOMMENDED)

TensorBoard: Start with 'tensorboard --logdir yolov5/runs/train', view at <http://localhost:6006/>

Downloading <https://ultralytics.com/assets/Arial.ttf> to /root/.config/Ultralytics/Arial.ttf...

100% 755k/755k [00:00<00:00, 145MB/s]

Downloading <https://github.com/ultralytics/yolov5/releases/download/v6.1/yolov5s.pt> to yolov5s.pt...

100% 14.1M/14.1M [00:00<00:00, 104MB/s]

Batch : 32

Epochs : 300

```

Epoch 0/299   gpu_mem    box      obj      cls    labels  img_size
              7.4G     0.08612  0.07333  0.1052   431      640: 100% 52/52 [00:51<00:00, 1.01it/s]
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 0% 0/8 [00:00<?, ?it/s] WARNING: NMS time limit 2.220s exceeded
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 100% 8/8 [00:15<00:00, 1.97s/it]
              all      474      3461    0.00496  0.233    0.0165    0.00588

Epoch 1/299   gpu_mem    box      obj      cls    labels  img_size
              9.2G     0.05967  0.05677  0.09496   344      640: 100% 52/52 [00:46<00:00, 1.13it/s]
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 12% 1/8 [00:02<00:20, 2.89s/it] WARNING: NMS time limit 2.220s exceeded
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 100% 8/8 [00:14<00:00, 1.77s/it]
              all      474      3461    0.0126   0.395    0.028     0.0132

Epoch 2/299   gpu_mem    box      obj      cls    labels  img_size
              9.2G     0.05475  0.05526  0.08763   356      640: 100% 52/52 [00:46<00:00, 1.11it/s]
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 100% 8/8 [00:13<00:00, 1.65s/it]
              all      474      3461    0.0113   0.525    0.0531    0.0251

Epoch 3/299   gpu_mem    box      obj      cls    labels  img_size
              9.2G     0.04385  0.04999  0.08118   362      640: 100% 52/52 [00:46<00:00, 1.11it/s]
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 100% 8/8 [00:11<00:00, 1.39s/it]
              all      474      3461    0.923    0.0757   0.0952    0.0449

Epoch 4/299   gpu_mem    box      obj      cls    labels  img_size
              9.2G     0.0389   0.04785  0.07393   377      640: 100% 52/52 [00:46<00:00, 1.11it/s]
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 100% 8/8 [00:11<00:00, 1.39s/it]
              all      474      3461    0.868    0.1     0.114     0.0572

Epoch 5/299   gpu_mem    box      obj      cls    labels  img_size
              9.2G     0.03588  0.04546  0.06911   393      640: 100% 52/52 [00:46<00:00, 1.11it/s]
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 100% 8/8 [00:10<00:00, 1.26s/it]
              all      474      3461    0.896    0.127    0.153     0.0939

Epoch 6/299   gpu_mem    box      obj      cls    labels  img_size
              9.2G     0.03473  0.04417  0.06449   357      640: 100% 52/52 [00:46<00:00, 1.12it/s]
              Class    Images    Labels      P      R    mAP@.5 mAP@.5:.95: 100% 8/8 [00:09<00:00, 1.25s/it]
              all      474      3461    0.892    0.134    0.207     0.136

```

총 소요시간 : 4.5시간

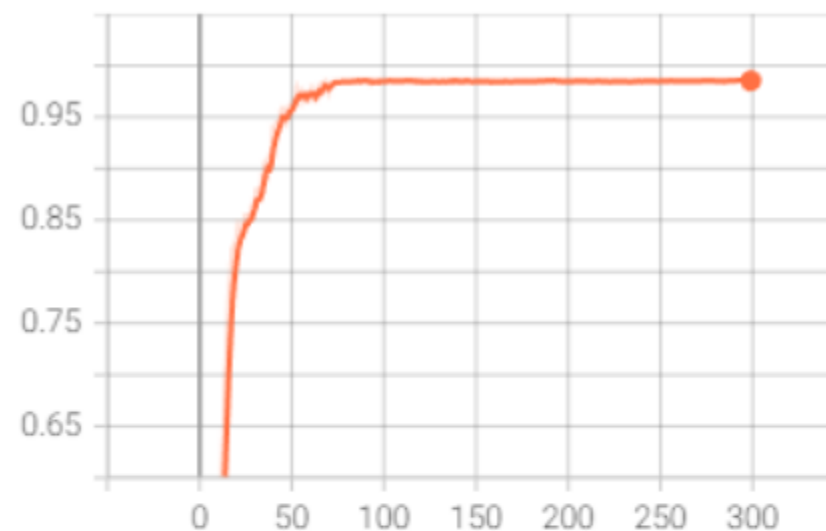
Precision : 0.951

Recall : 0.975

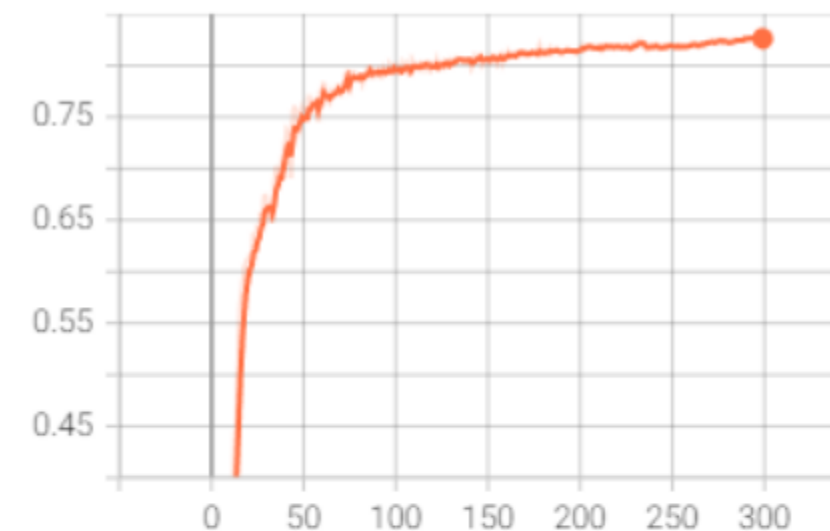
mAP@.5 : 0.985

mAP@[.5: .95] : 0.826

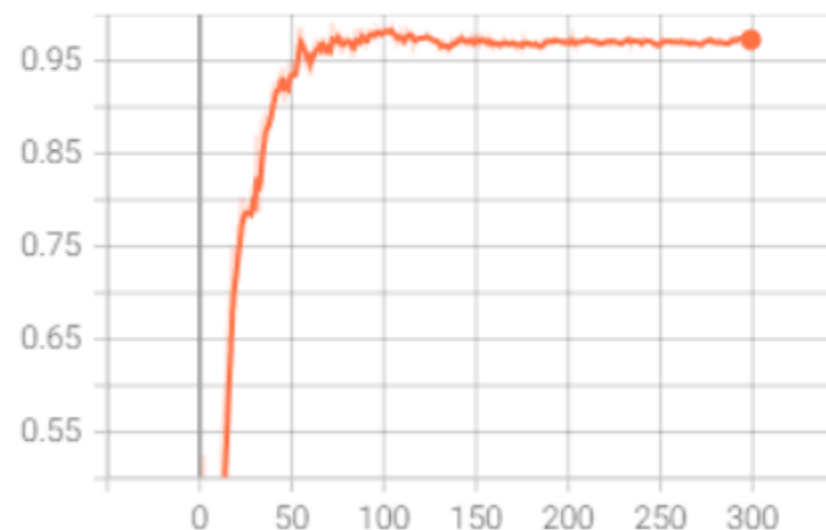
metrics/mAP_0.5
tag: metrics/mAP_0.5



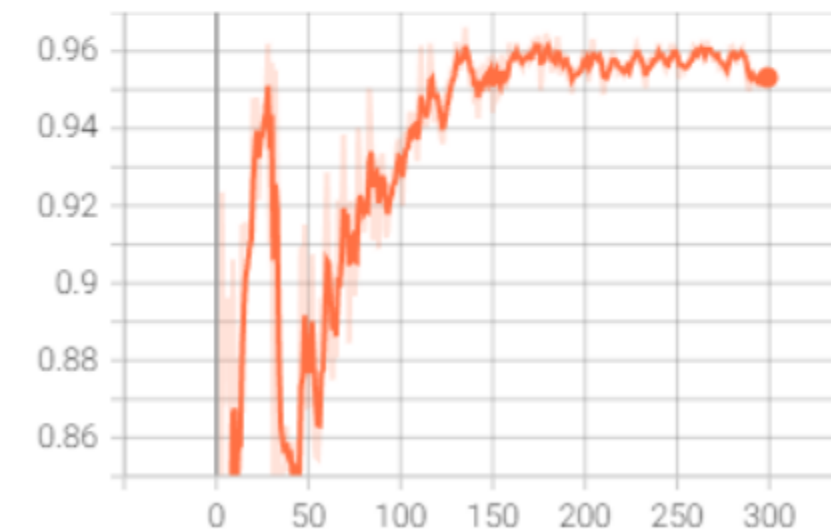
metrics/mAP_0.5:0.95
tag: metrics/mAP_0.5:0.95

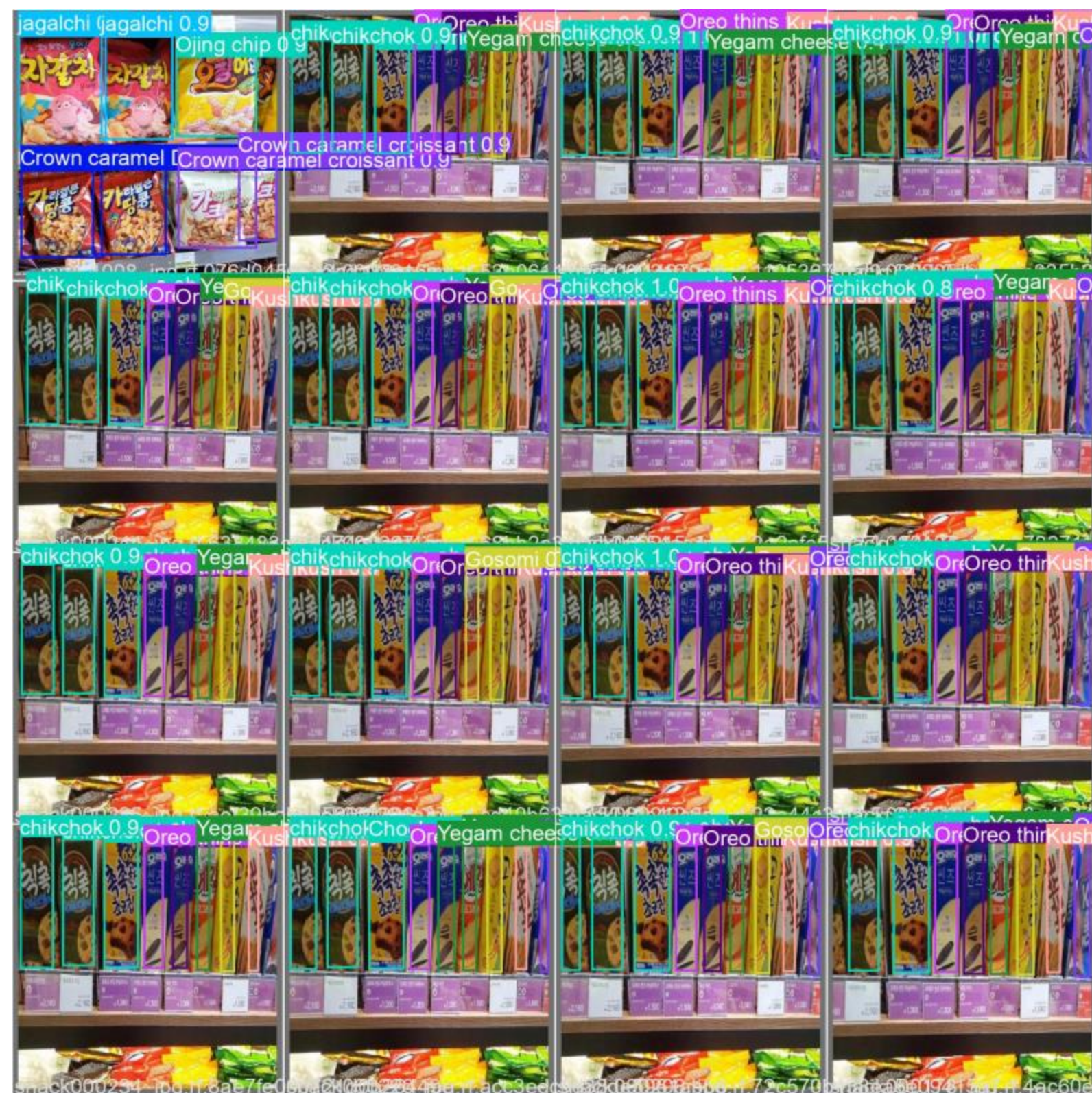
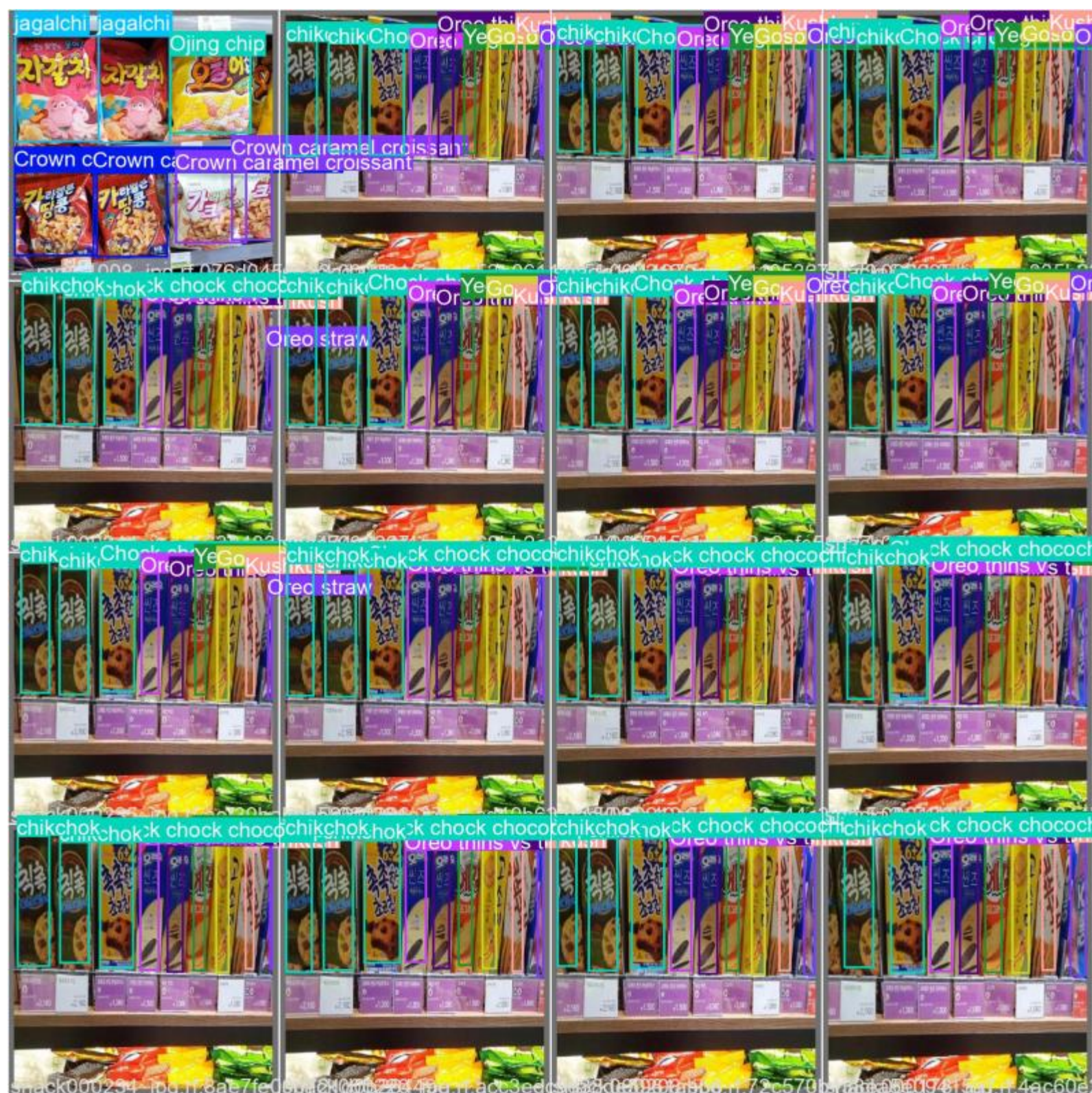


metrics/recall
tag: metrics/recall



metrics/precision
tag: metrics/precision





개발 환경



개발 툴



프로그래밍 언어



**딥러닝 모델
프레임워크**

주요 기능: Text To Speech

tts process



인식한 사물을 이용자에게 알리기
위하여 구글 TTS API를 활용,
음성 출력 기능 구현

앱 UI 소개



카메라 촬영부분

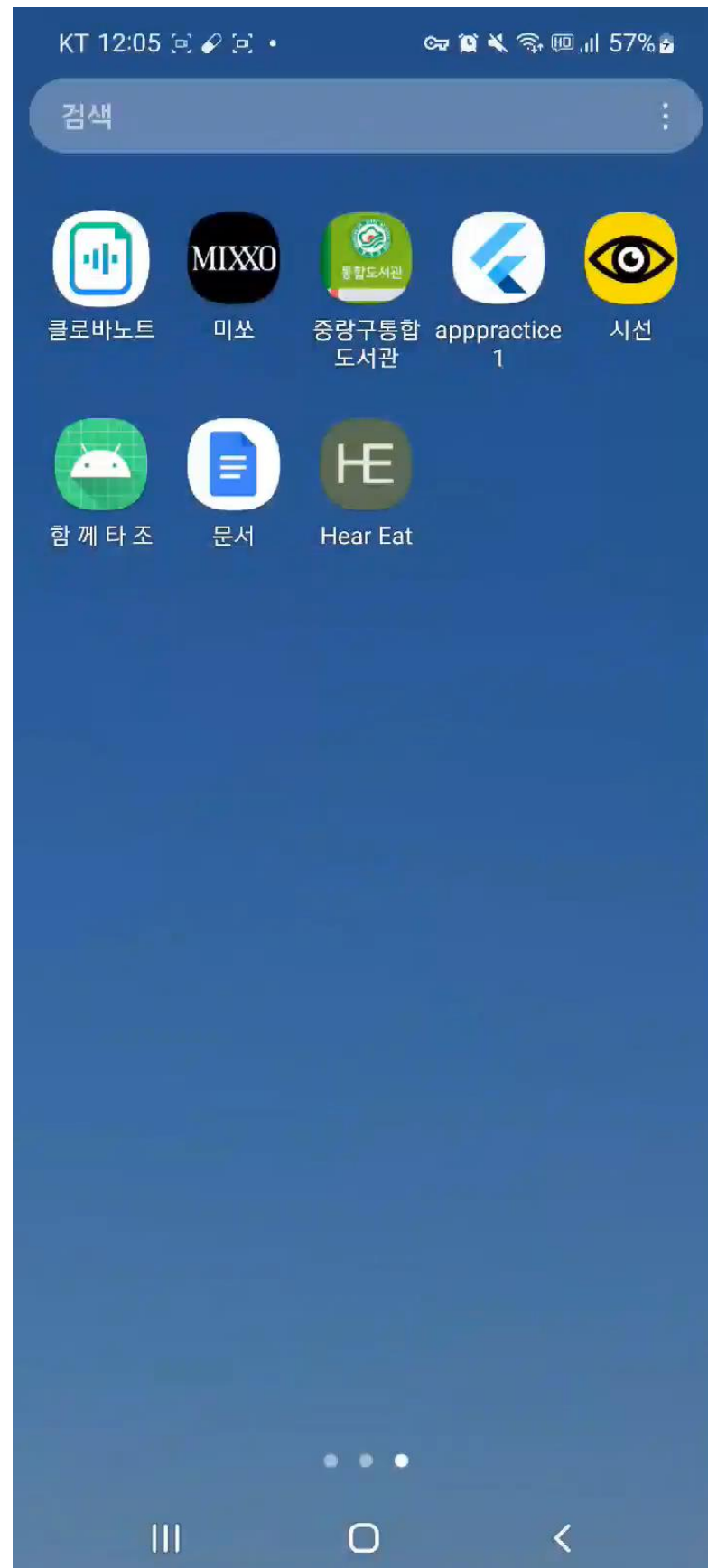
별도의 화면 전환 등의 불필요한 요소 배제,
직관성을 유지하고자 함

바운딩 박스(식별 표시 기능)

글자 색상 (흰색, #FFFFFF)과, 식별 박스 배경 색상 (녹색, #5C6E56)간의 명도 차이가 5.49:1로 (기준: 4.5:1)
저시력 장애인의 가독성을 높임

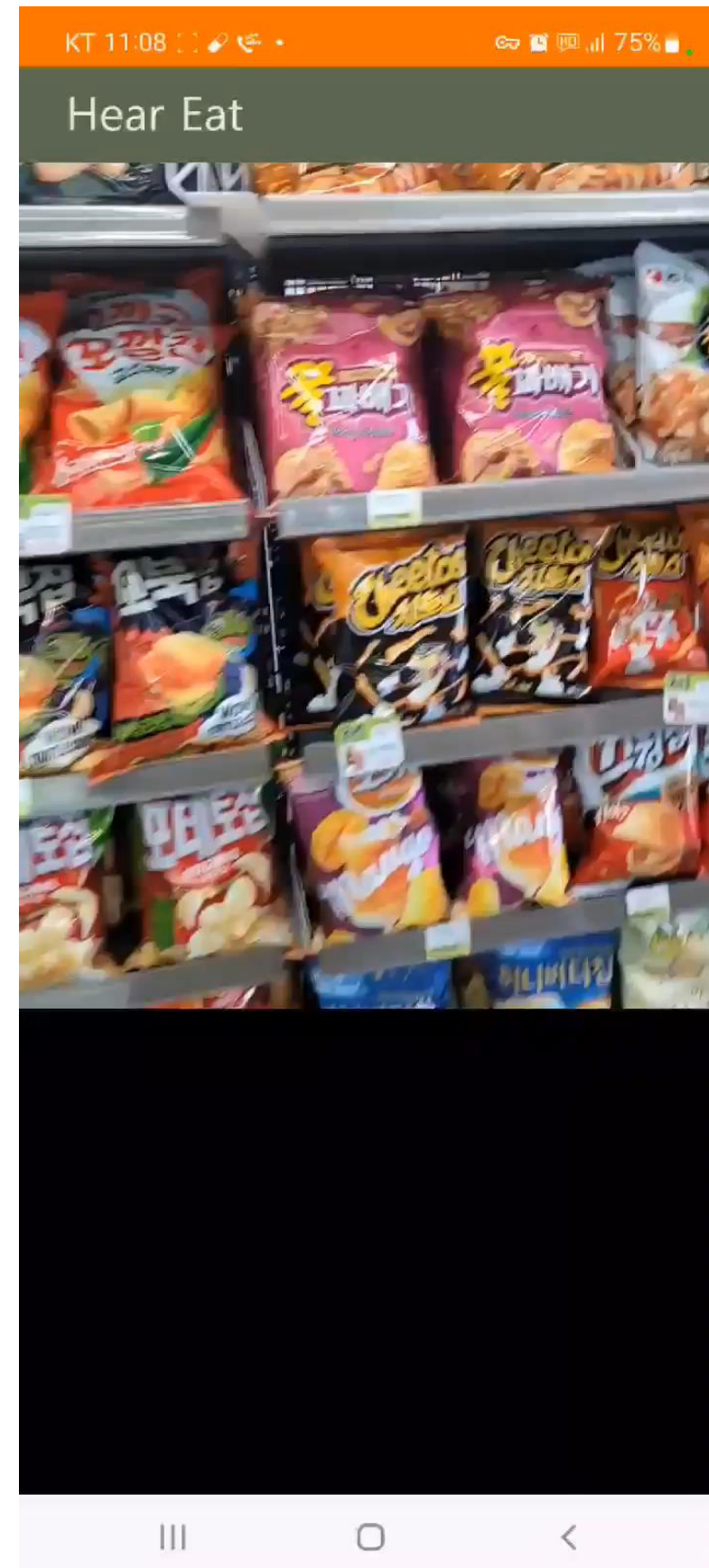
터치 버튼(화면 전체)

주 사용자의 특성을 고려하여,
화면 전체가 클릭 가능한 버튼으로 기능



Hear Eat

실행 예시 (동영상)

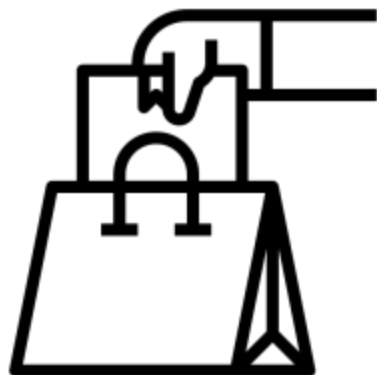




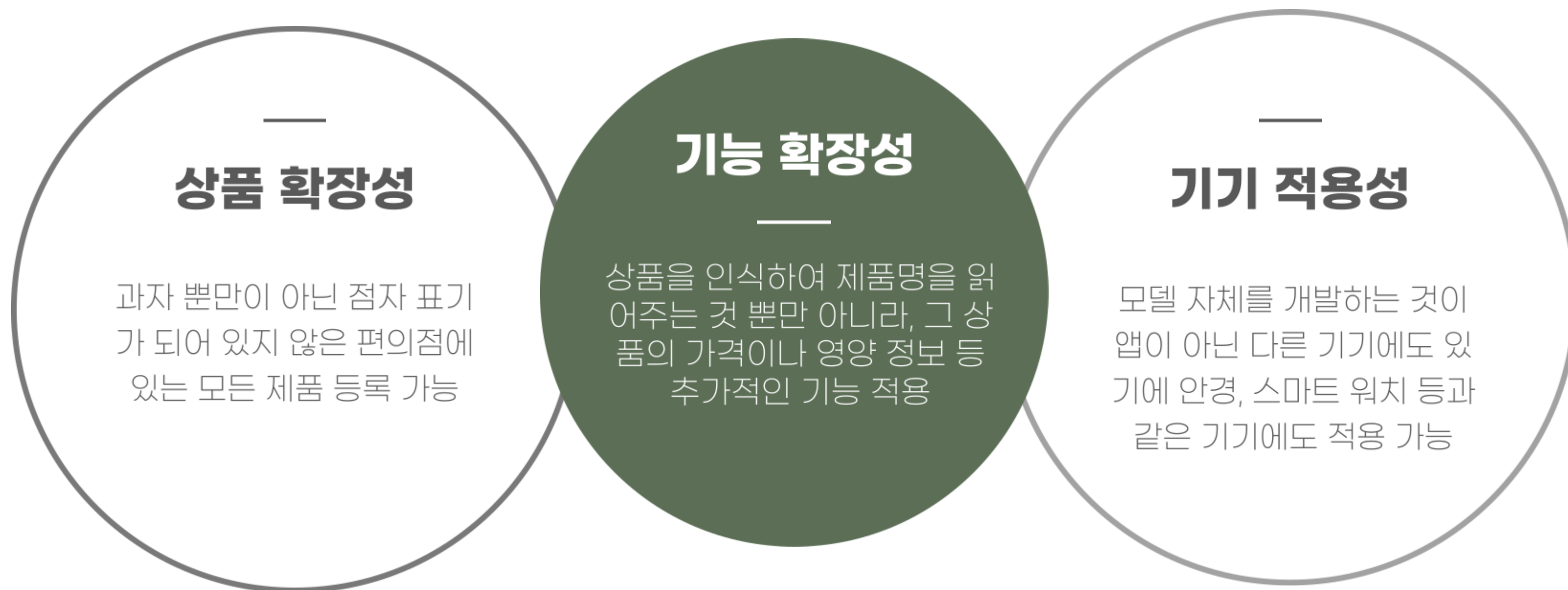
타인의 도움 필요 없이 시각장애인들이 혼자서 사용 가능



점자 표기가 되지 않은 상품 또한 구별 가능



시각장애인에 대한 소비자의 권리 보장



Q & A

감사합니다