

AI를 활용한

스마트 무인계산대

[K-Digital] 멀티캠퍼스 7조
우리 똑똑하조

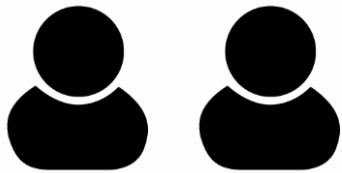


정해준 조해성 최희원 전창도 박건우 안도현 김민주 강인솔

팀원소개

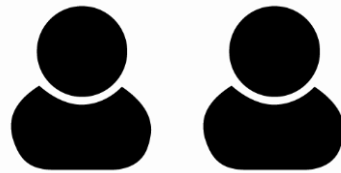


IoT



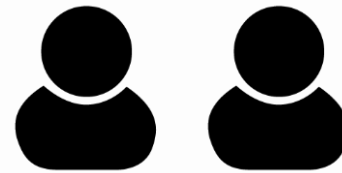
정해준 조해성

BigData



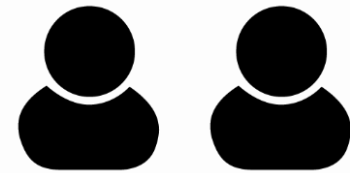
최희원 전창도

AI



박건우 안도현

Cloud



강인솔 김민주

개요 및 배경



- 코로나 19로 인한 경기 불황
- 오프라인 업체 매출은 회복 추세
 - 백화점을 제외하고 대형마켓 매출 하락
 - 편의점 매출 증가
- 최저시급 인상으로 인한 인건비 증가

4월 오프라인 매장 매출 증감

단위: %, 전년 동월 대비



자료: 산업부

The JoongAng

개요 및 배경



• 코. 코로나 끝 보이는데...마트 손님은 돌아오지 않는다

[중앙일보] 입력 2021.05.27 15:28 수정 2021.05.27 17:30




김남준 기자

- 롯데 서울 동작구에 사는 박은경(40)씨는 지난해 주로 온라인으로 장을 봤다. 신종 코로나바이러스 감염증(코로나19) 우려에 동네 마트도 가기 꺼려졌기 때문이다. 빠른 배송 덕분에 온라인 장보기가 그리 불편하지 않은 것도 이유였다. 박씨는 “온라인이 할인도 많은 데다 실제 이용해 보니 믿을 만한 업체도 많아져 계속 이용하고 있다”면서 “코로나19가 끝나도 적어도 생필품은 계속 온라인으로 살 것 같다”고 했다.

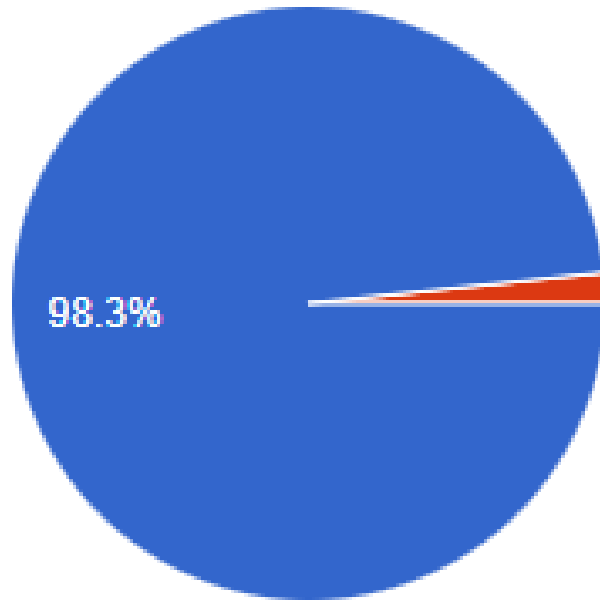


개요 및 배경

무인 캐서 설문조사 

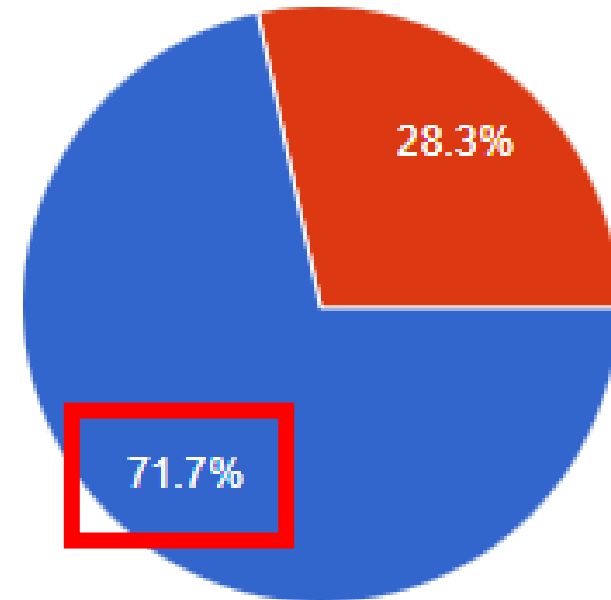
응답 115개

셀프 계산대를 이용해본 경험이 있나요?




셀프 계산대를 이용해보셨다면 불편했던 경험이 있나요?

● 예
● 아니요



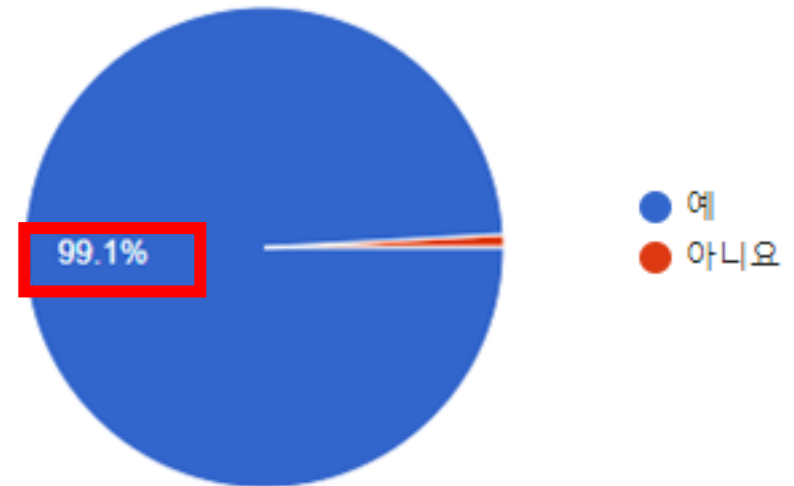
개요 및 배경

무인 캐셔 설문조사 

불편하셨다면 어떤 점이 불편하셨나요?



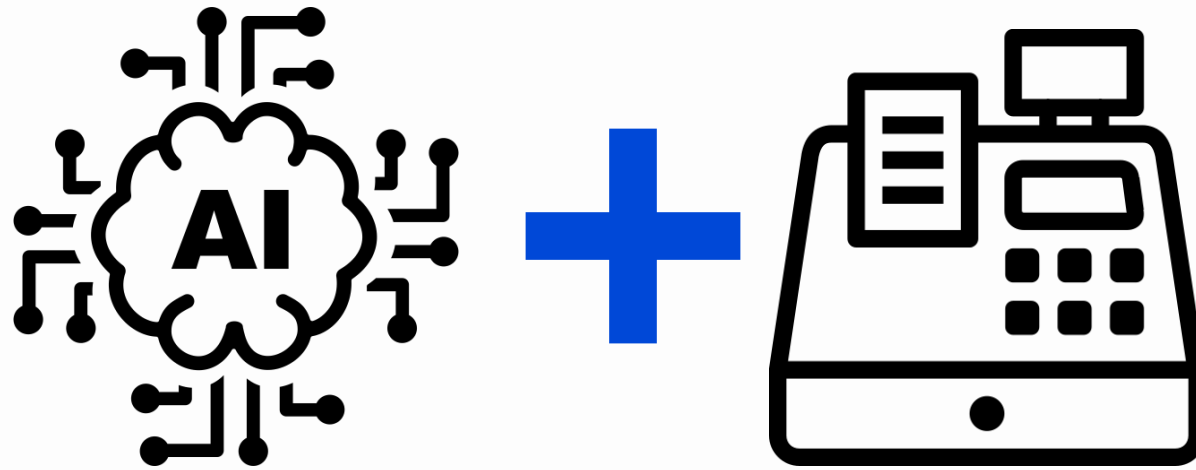
일일이 바코드를 찍지 않아도 되는
셀프 계산대가 있다면 사용하실 의향이 있나요?



주제 선정



Ai를 통해 바코드를 찍지 않고 즉시 계산 가능



번거로운 조작이 필요하지 않아 편리한
AI 스마트캐셔

타겟설정



편의점 및 구매고객이 많은 소규모 매장



공간은 협소하지만 이용 고객의 수는 많은 편의점
많은 고객들의 이용으로 인해 효율적인 자원 분배가 어려움
스마트 캐셔를 설치하여 인건비 절약 및 효율적인 자원 확보 및 분배 예상

차별점 및 사업성



기존 유인캐셔

- 효율적 자원 분배
- 매장 내 회전율 증가

기존 셀프계산대

- 시간 단축
- 사용자 편의성

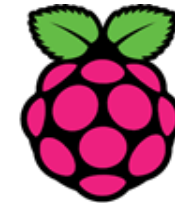
사용기술



BeautifulSoup

Numpy

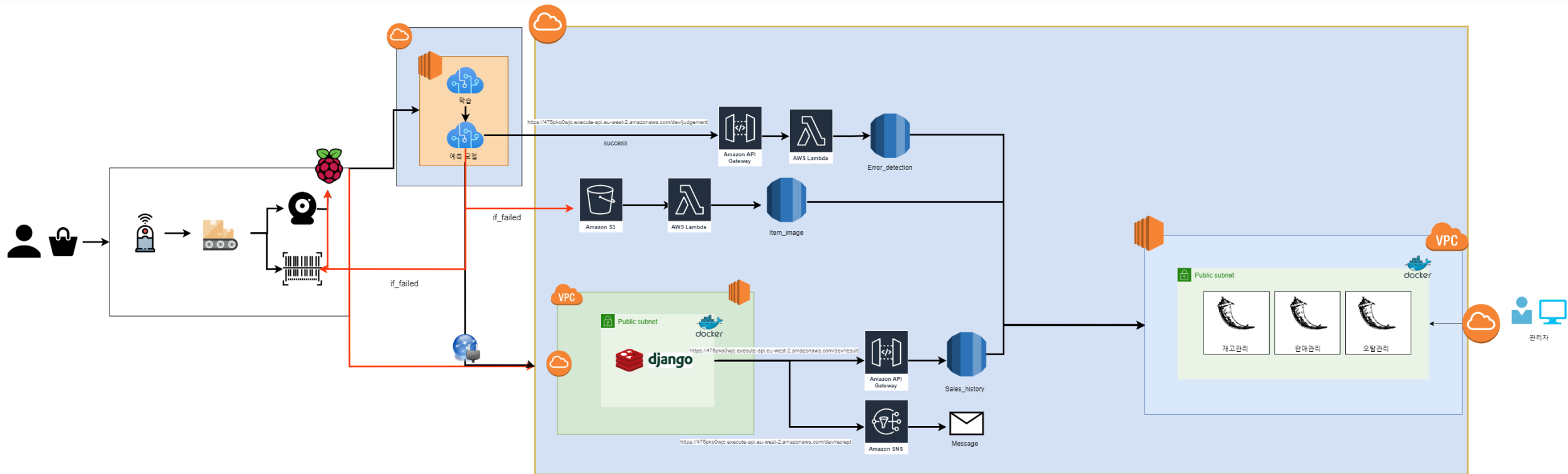
pandas



세부 내용



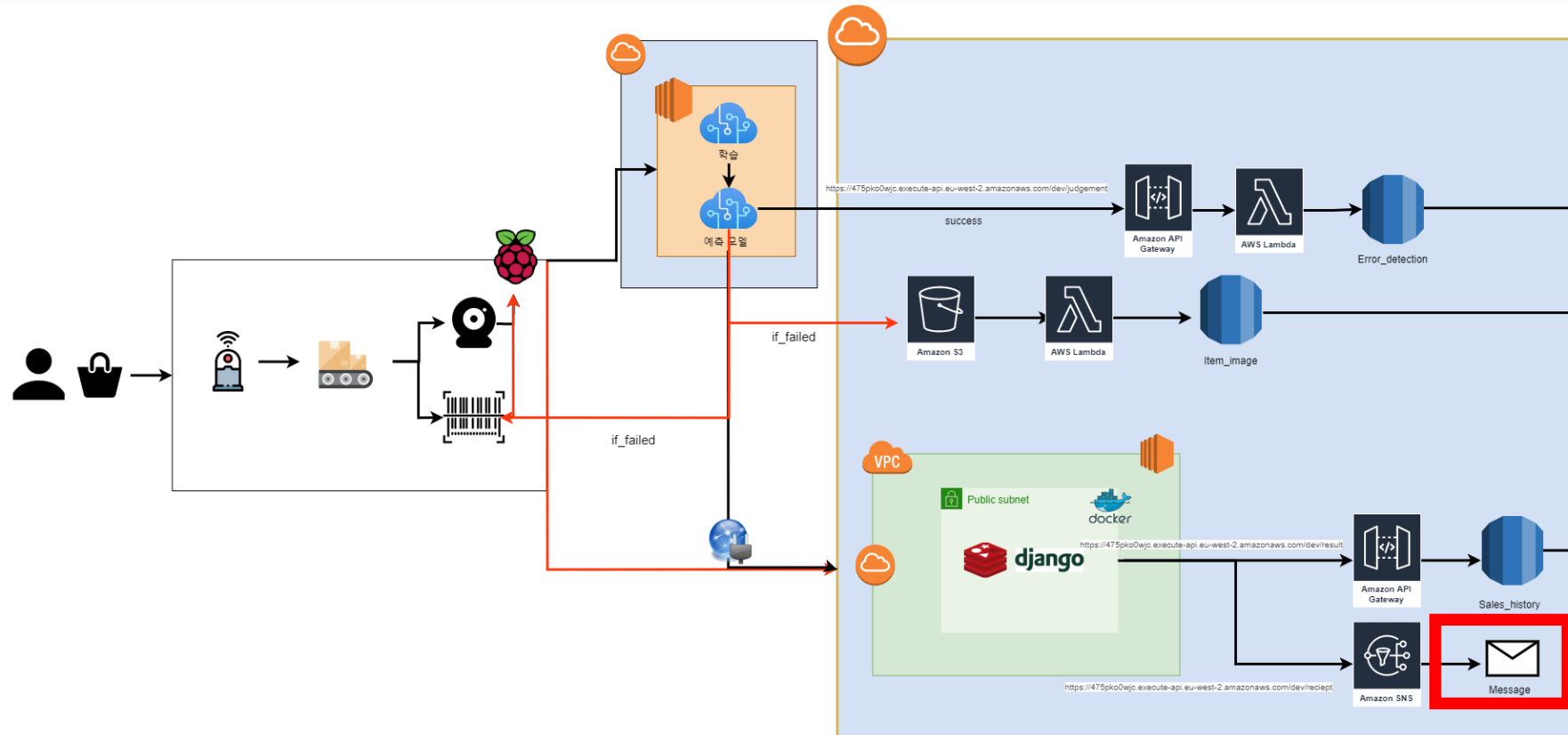
- Use-case 다이어그램



세부 내용



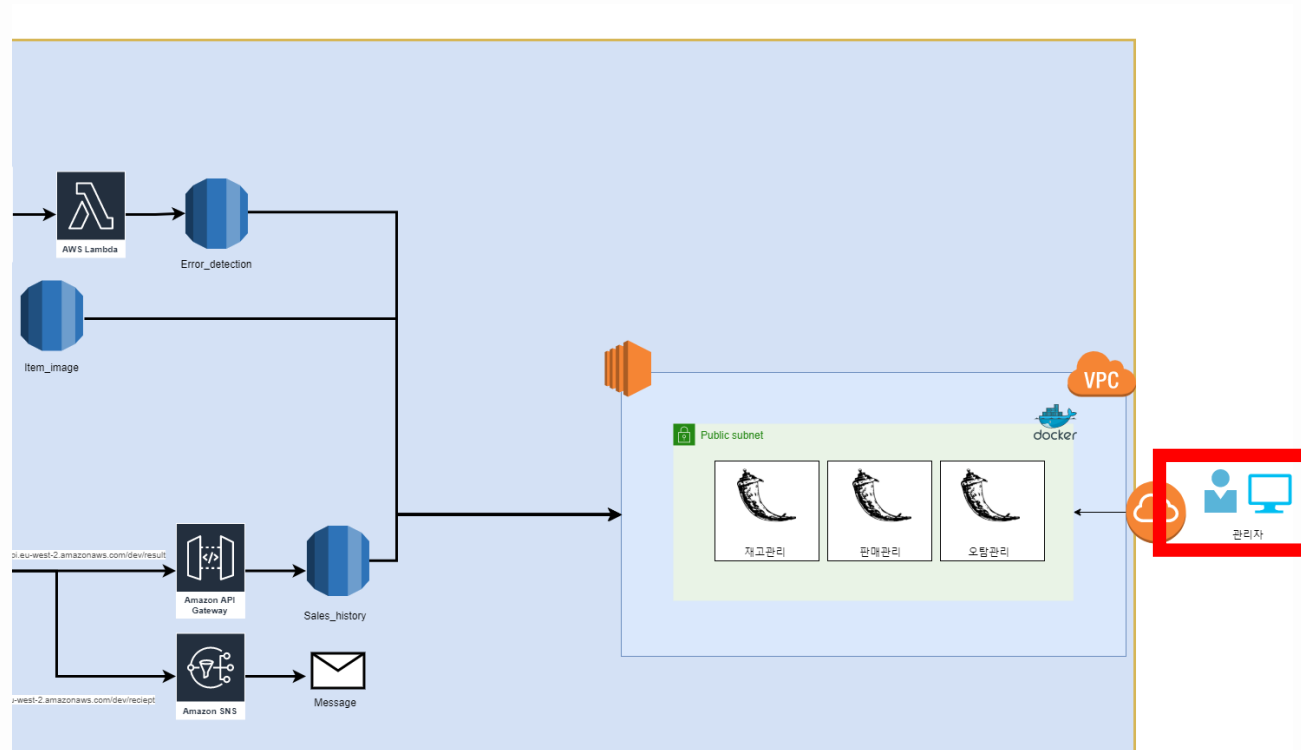
- 사용자 다이어그램



세부 내용



- 관리자 다이어그램



7조 스마트 무인 계산대

IoT

하드웨어 구성



ABOUT PROJECT

제품 접근 인식 및 컨베이어 벨트

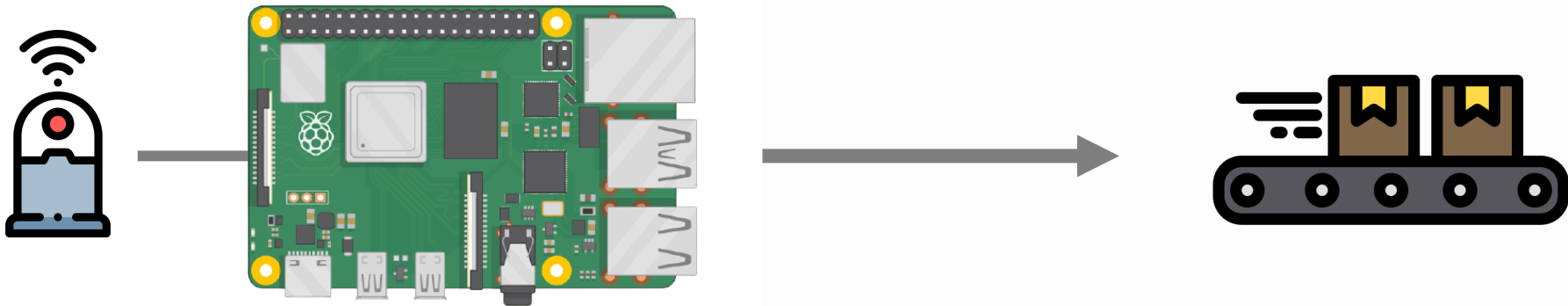
사진 촬영 및 이미지 전송

인식오류 발생 대처

IoT 하드웨어 구성



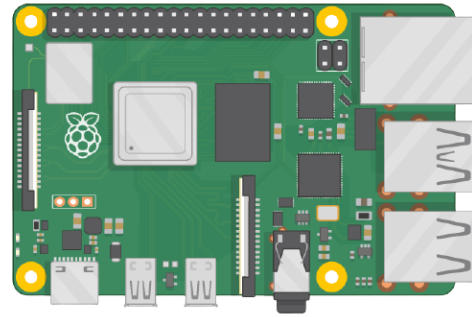
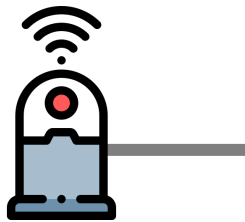
제품 접근에 따른 컨베이어 벨트 동작 설명



IoT 하드웨어 구성



적외선 센서 작동 방식



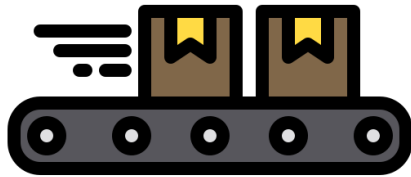
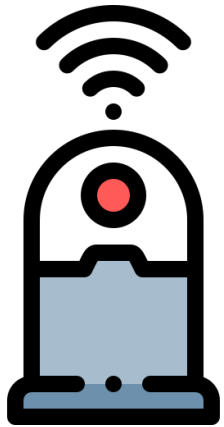
RaspberryPi

- SPI dev 라이브러리 이용
- 아날로그 신호 → 디지털 신호
- 적외선 센서의 송신부와 수신부 사이에
제품 감지 시 신호 값 0 출력

IoT 하드웨어 구성



컨베이어 벨트 작동



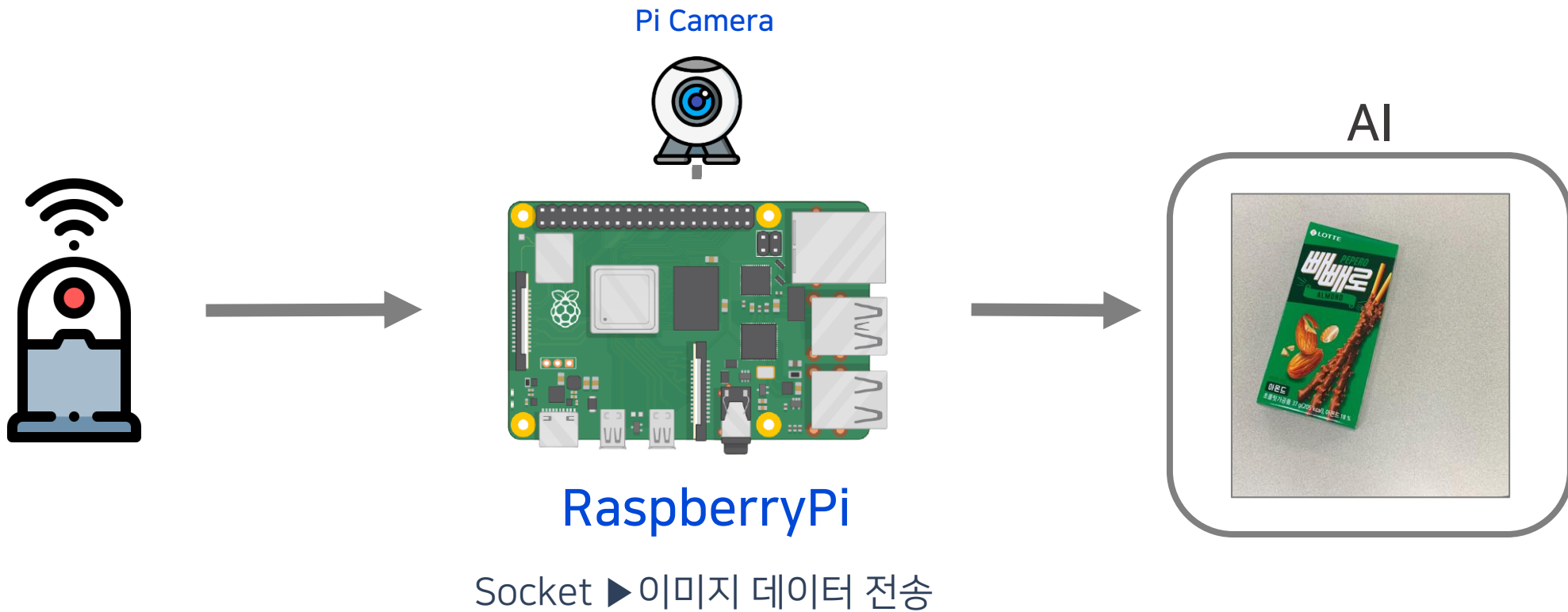
컨베이어 벨트 가동

- DC 모터 작동 → 물체 이송
- 라즈베리파이 DC모터 가동
(GPIO BCM모드)
- PWM으로 속도 조절

IoT 하드웨어 구성

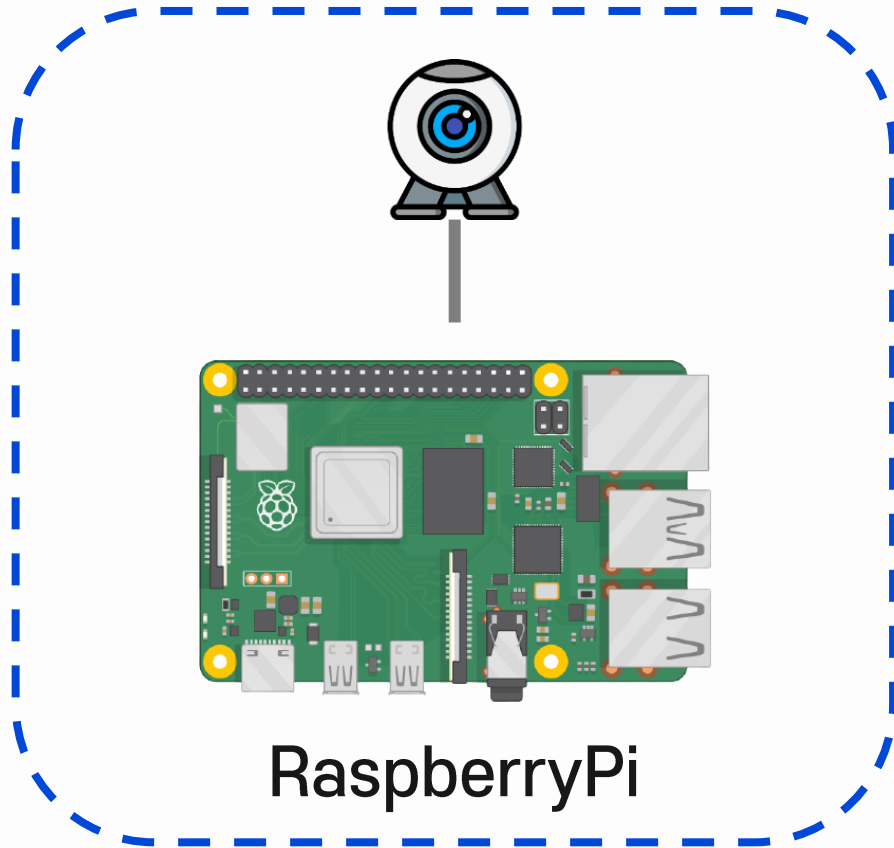


사진촬영 및 이미지 전송



IoT 하드웨어 구성

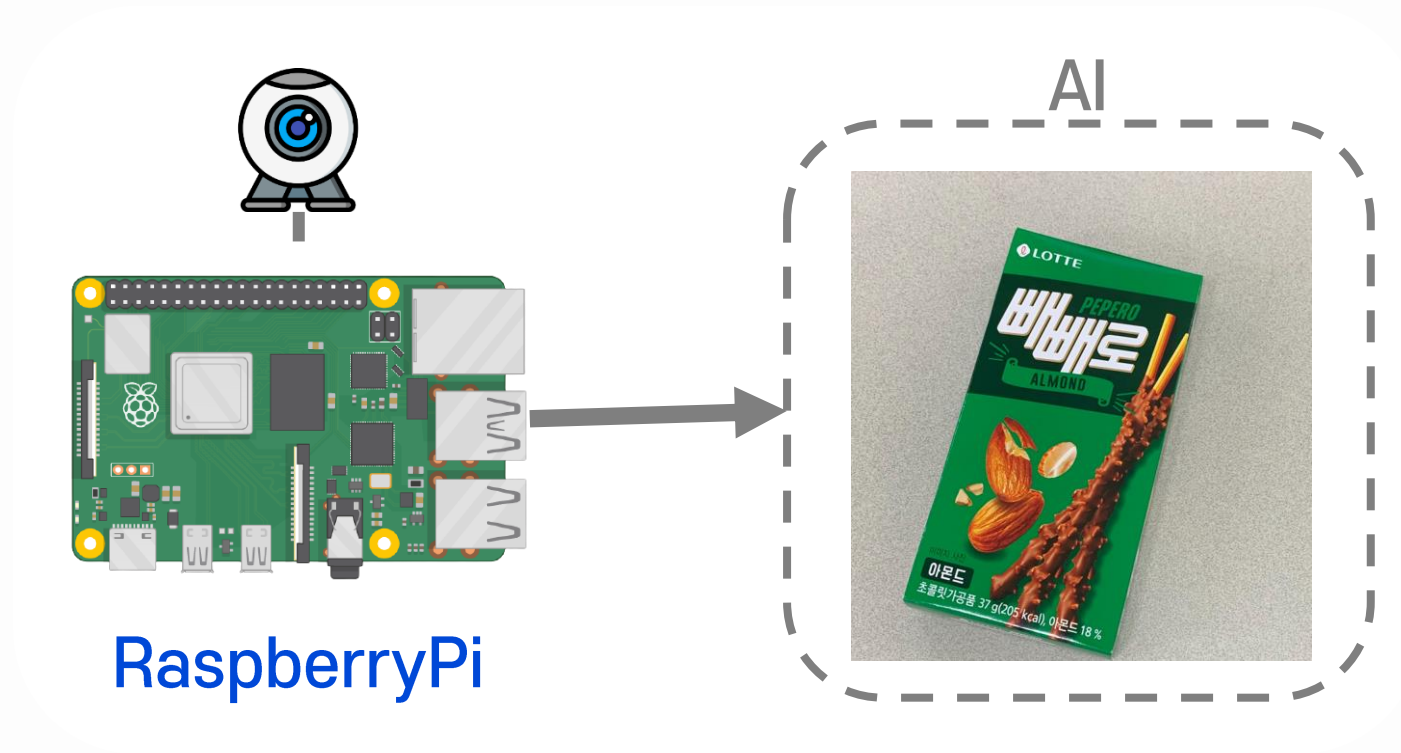
사진촬영 및 이미지 전송 



- 이미지 촬영 시점은 적외선 센서 이용
- OpenCV를 통해 이미지 촬영
- 사진 촬영 완료 시 신호(LED, 소리) 출력

IoT 하드웨어 구성

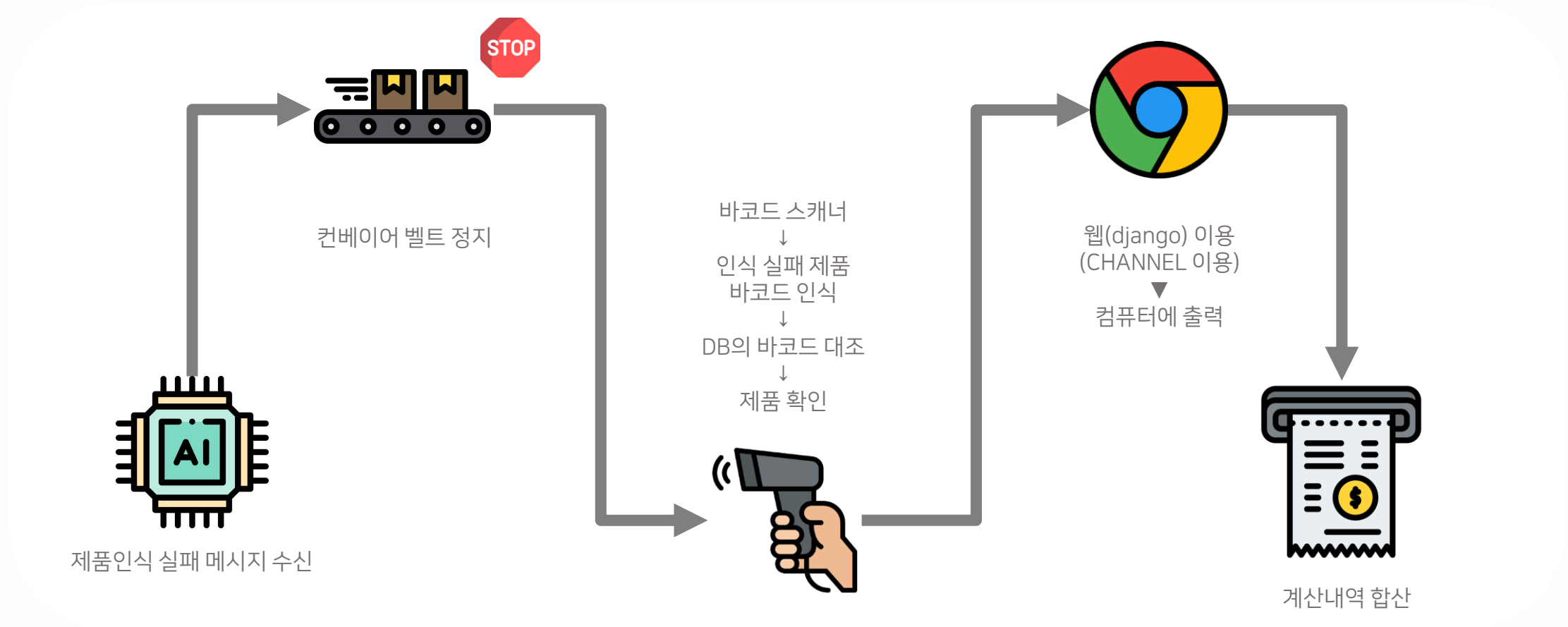
사진촬영 및 이미지 전송 




- Struct 라이브러리로 데이터 가공
- 가공된 바이너리 데이터 전송
- 전송 시 Socket 이용하여 AI EC2로 전송

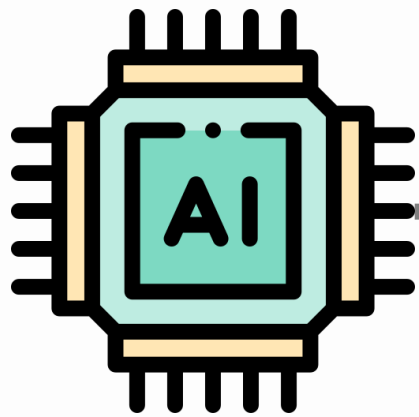
IoT 하드웨어 구성

인식오류 발생 대처 



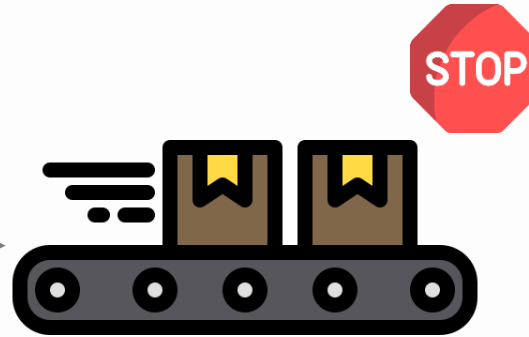
IoT 하드웨어 구성

인식오류 메시지 수신 



AI

제품인식 실패 메시지 수신




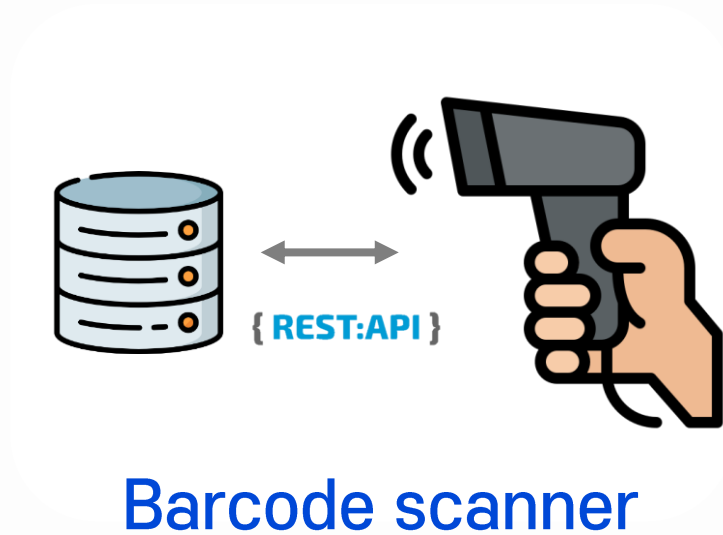
컨베이어 벨트 정지

Socket 구성

- 서버 : AI EC2
- 클라이언트 : Raspberry Pi
- AI 인식 실패
→ 중지 메시지 발신
- IoT 수신 → 모터 중지

IoT 하드웨어 구성

바코드 인식 제품확인 



- 스캔한 바코드 정보
→ 클라우드 DB 추출
- REST API 이용
- 통신 : Web socket
- 데이터: json
- 추출한 제품정보
Django 서버로 전송

7조 스마트 무인 계산대

AI

분류 및 객체 인식 모델링



ABOUT PROJECT

- 분류 모델링
- 객체 인식 모델링

분류 및 객체 인식 모델링



20 classes (20 items)



1,807 images

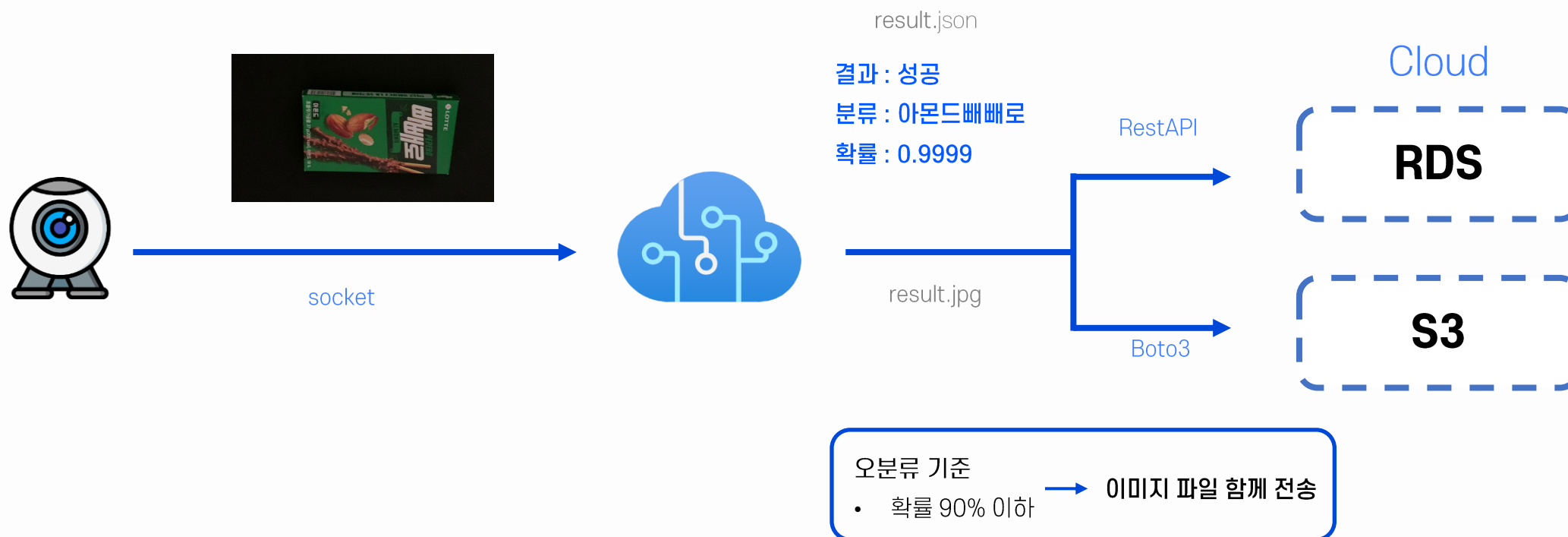
...



1,818 images

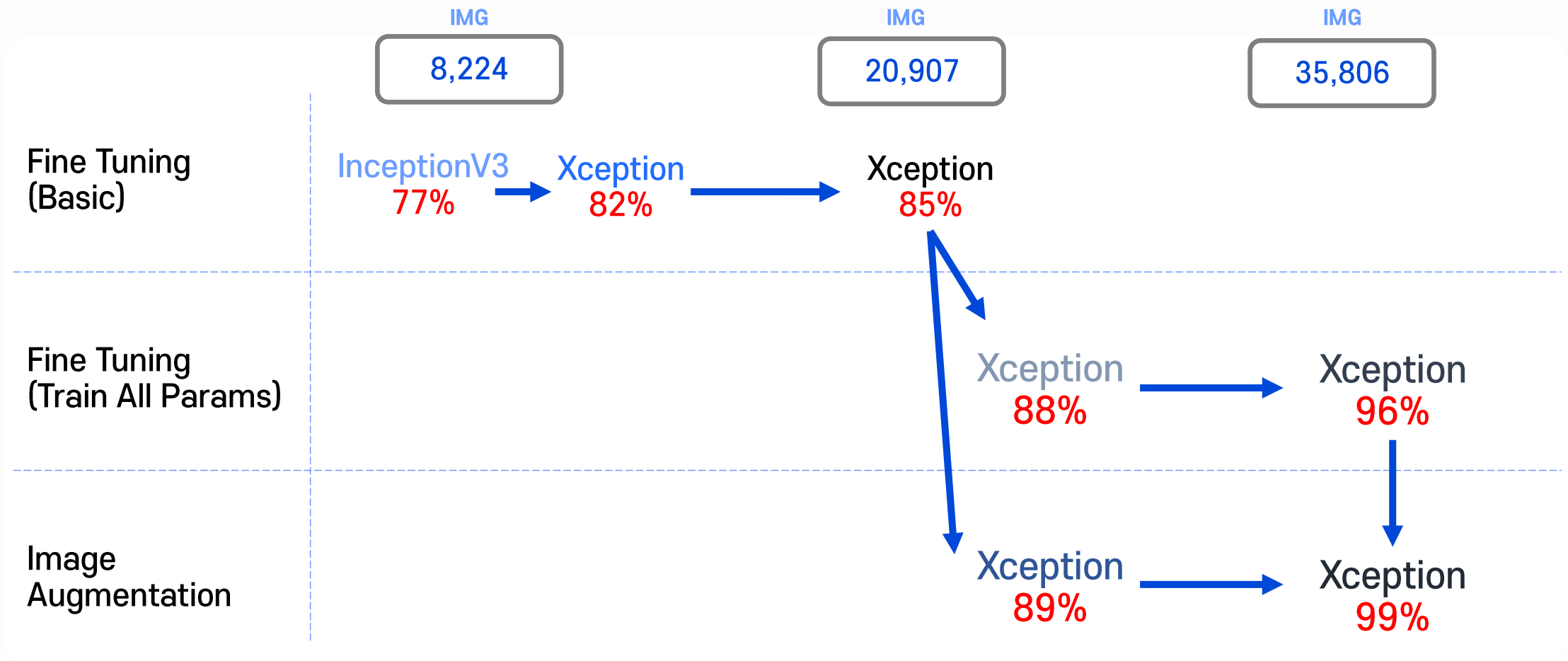
35,806 images 학습/검증데이터 7:3

분류 및 객체 인식 모델링



분류 및 객체 인식 모델링

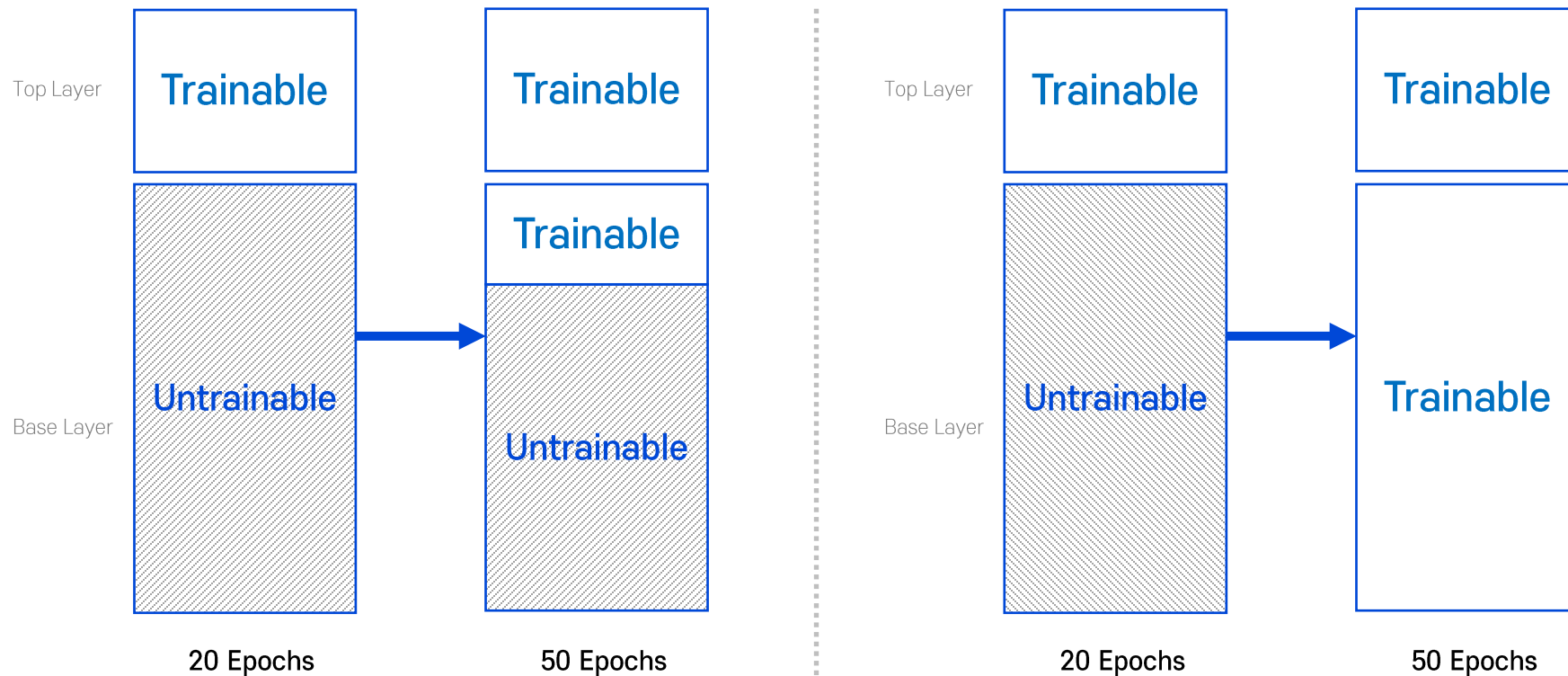
Modeling History 



분류 및 객체 인식 모델링

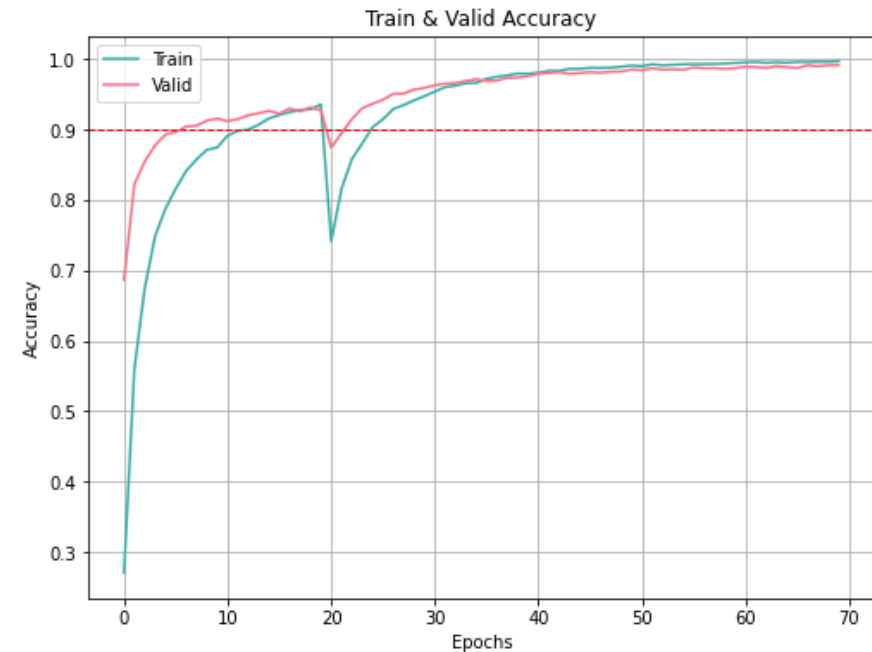
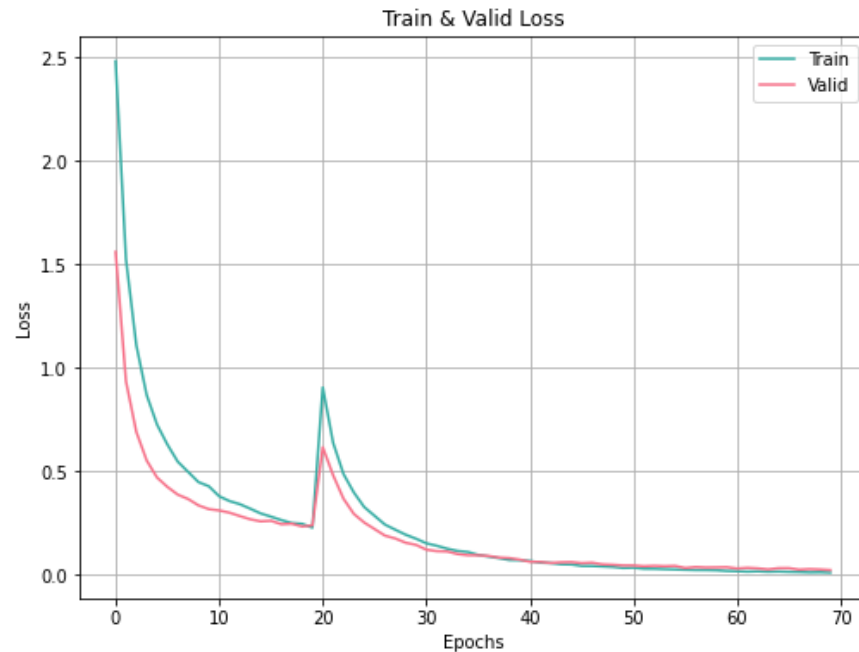
Modeling History 

Fine Tuning(Basic) 85% → 89% Fine Tuning(Train All Params)
색깔 구분 불가



분류 및 객체 인식 모델링

Performances 



- Train

Loss 0.0157

Accuracy 0.9961

- Validation

Loss 0.0265

Accuracy **0.9912**

실제 사용 이미지

- Test

Loss

Accuracy **0.96**

분류 및 객체 인식 모델링



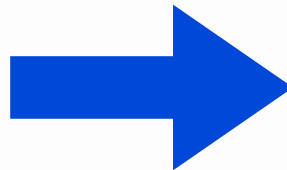
Object Detection Model

Classification Model Issue

- 한 번(한 사진)에 한 개의 물품만 분류 가능

▶ 시간 오래 걸림

- 사진 배경에 따른 영향 多



객체 인식 모델 (YOLO v4 모델)

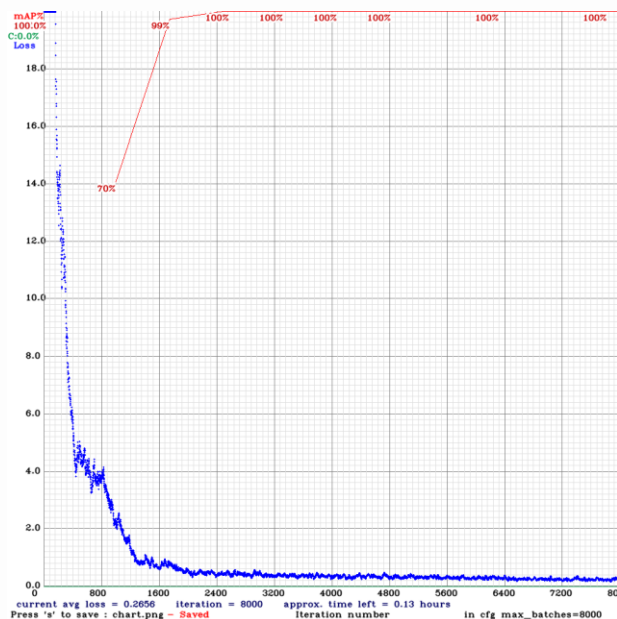


- 한 번(한 사진)에 여러 개의 물품 분류 가능

분류 및 객체 인식 모델링



Model Performance



mAP : 99.97%
학습 소요 시간 : 8시간

Test Performance

Test Dataset

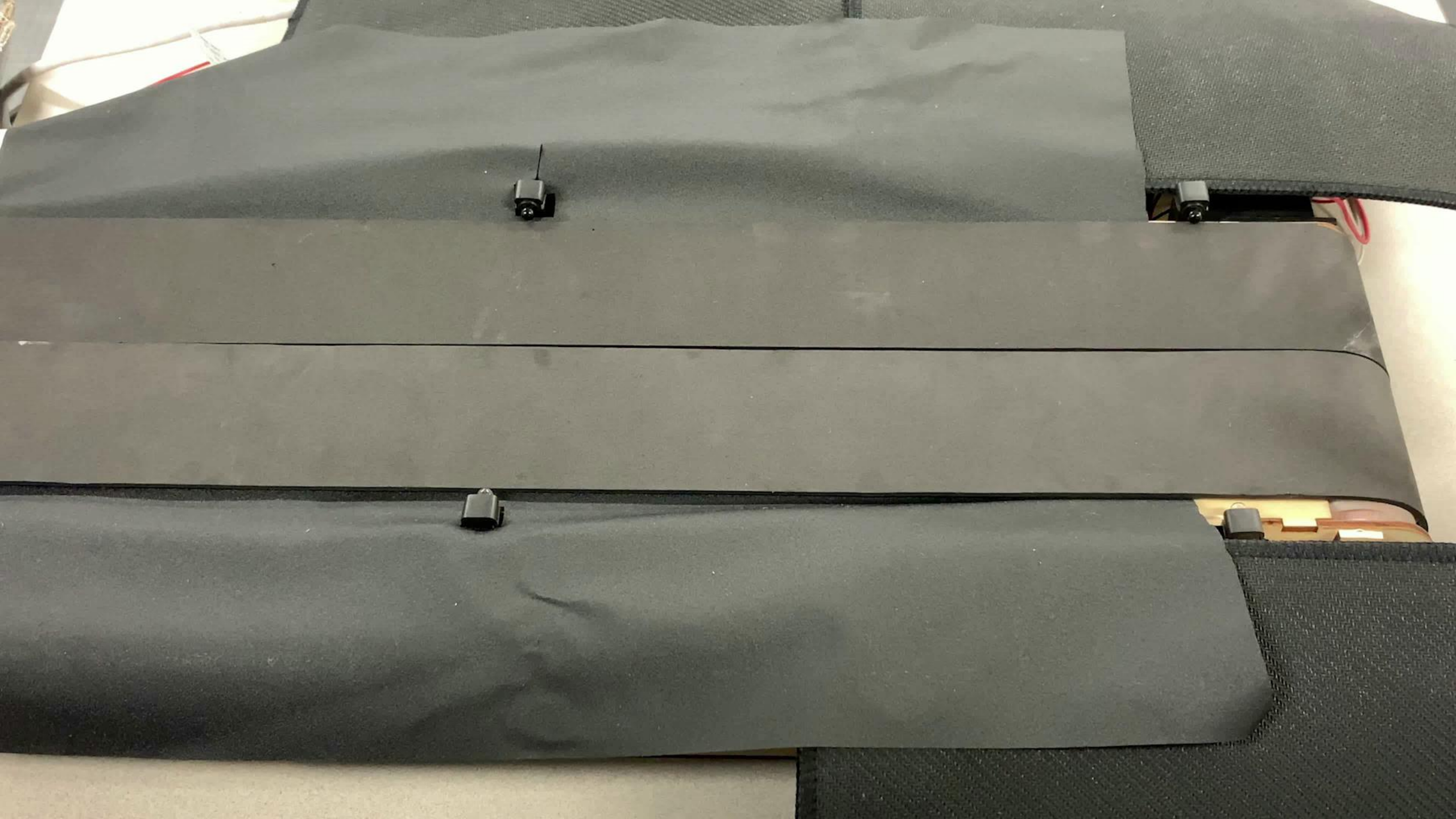


검출율 : 99%
오검출율 : 1%

> 이미지 100장

> 이미지당 아이템 2 ~ 5개

> 겹치지 않도록 가지런히



7조 스마트 무인 계산대

BIG DATA

데이터와 연관분석



ABOUT PROJECT









- 제품 정보 수집 및 전처리
- 이미지 데이터 수집 및 전처리
- 연관분석 활용한 마케팅

데이터와 연관분석

제품 정보 크롤링 및 DB 삽입



- CU 홈페이지 상품소개 페이지 크롤링
- 전처리 후 Item_info 테이블에 삽입

PB상품		상품소개	HEYROO 캐릭터
간편식사	즉석조리	과자류	아이스크림
식품	음료	상할용품	
 롯데) 화이트키위쿠키1500 1,500원 2+1	 롯데) 아몬드빼빼로1500 1,500원 2+1	 롯데) 초코빼빼로1500 1,500원 2+1	 해태) 알기웨하스900 900원
 해태) 크림웨하스900 900원	 해태) 맛동산90g 2,000원 2+1	 해태) 자유시간아몬드36g 1,000원	 삼립) 연유오팅 1,300원

Unnamed#multicampus#item_info# - HeidiSQL 11.2.0.6213

파일 편집 검색 쿼리 도구 이동 도움말

데이터베이스 필터 테이블 필터

multicampus.Item_info: 3,308 행 (총) (대략적), 제한 수: 1,000

Item_id	Product_name	Price	Qty	Category
11,720	오리온_닥터유에너지바	1,200	100	snack
11,721	롯데_화이트키위빼빼로1500	1,500	100	snack
11,722	롯데_아몬드빼빼로1500	1,500	100	snack
11,723	롯데_초코빼빼로1500	1,500	100	snack
11,724	해태_알기웨하스900	900	100	snack
11,725	해태_크림웨하스900	900	100	snack
11,726	해태_맛동산90g	2,000	100	snack
11,727	해태_자유시간아몬드36g	1,000	100	snack
11,728	삼립_연유오팅	1,300	100	snack
11,729	HEYROO쌀떡볶이1200	1,200	100	snack
11,730	HEYROO신나는뉴장1200	1,200	100	snack
11,731	HEYROO달콤한대롱과자12...	1,200	100	snack
11,732	HEYROO교룡이라면스낵12...	1,200	100	snack
11,733	HEYROO뽕이요스낵1200	1,200	100	snack
11,734	일신_푸쉬팝트리플캔디	3,000	100	snack
11,735	HEYROO간장치킨맛팝콘12...	1,200	100	snack
11,736	농심_오징어다리달달구이	1,500	100	snack
11,737	HEYROO카라멜맛팝콘1200	1,200	100	snack
11,738	HEYROO고소한치즈팝콘12...	1,200	100	snack
11,739	HEYROO콘소메맛팝콘1200	1,200	100	snack
11,740	오리온_후라보노콜민트	800	100	snack
11,741	농심_콘푸로스트230g	3,000	100	snack
11,742	롯데_쌀로벌오리지널	1,500	100	snack

데이터와 연관분석



1. 마이썬딸기
2. 다이제초코
3. 아몬드빠빠로
4. 초코빠빠로
5. 칠성사이다캔
6. 스니커즈땅콩
7. 씨리얼오트컵
8. 자일리톨용기껌
9. 지시후레쉬스틱팩검
10. 츄파춥스
11. 드림카카오72%
12. 칸초컵
13. 코카콜라캔
14. 코카콜라제로캔
15. 크런키더블크런치바
16. 페레로로쉐5구
17. 프링글스_오리지날
18. 허쉬밀크초콜릿
19. 허쉬큐앤크초콜릿
20. 홀스레몬맛



데이터와 연관분석

- 각 제품당 1800장의 이미지 데이터
- 윤곽선 검출 -> trimming -> YOLO labeling



Open CV

이미지 데이터 전처리



```
for k in range(len(path)):
    for img in glob.glob(path[k]+'/*.jpg'):

        image = cv2.imread(img)
        image_gray = cv2.imread(img, cv2.IMREAD_GRAYSCALE)

        b,g,r = cv2.split(image)
        image2 = cv2.merge([r,g,b])

        # blur = cv2.GaussianBlur(image_gray, ksize=(5,5), sigmaX=0)
        # blur = cv2.medianBlur(image_gray, 7)
        # blur = cv2.blur(image_gray, (5,5))
        blur = cv2.bilateralFilter(image_gray,9,75,75)

        ret, thresh1 = cv2.threshold(blur, 127, 255, cv2.THRESH_BINARY)
        edged = cv2.Canny(blur, 10, 250)
        kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (7,7))
        closed = cv2.morphologyEx(edged, cv2.MORPH_CLOSE, kernel)
        contours, _ = cv2.findContours(closed.copy(),cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        total = 0
        contours_image = cv2.drawContours(image, contours, -1, (0,255,0), 3)
        contours_xy = np.array(contours)

        # x의 min과 max 찾기
        x_min, x_max = 0,0
        value = list()
        for i in range(len(contours_xy)):
            for j in range(len(contours_xy[i])):
                value.append(contours_xy[i][j][0][0]) #네번째 괄호가 0일때 x의 값
                x_min = min(value)
                x_max = max(value)

        # y의 min과 max 찾기
        y_min, y_max = 0,0
        value = list()
        for i in range(len(contours_xy)):
            for j in range(len(contours_xy[i])):
                value.append(contours_xy[i][j][0][1]) #네번째 괄호가 0일때 y의 값
                y_min = min(value)
                y_max = max(value)

        # image trim 하기
        x = x_min
        y = y_min
        w = x_max-x_min
        h = y_max-y_min
```


데이터와 연관분석

- **Kaggle** : Grocery product 데이터 활용
- Sales_history 테이블에 데이터 삽입

	A	B	C	D	E	F	G
1	pork	sandwich bags	lunch meat	all- purpose	flour	soda	butter
2	shampoo	hand soap	waffles	vegetables	cheeses	mixes	milk
3	pork	soap	ice cream	toilet paper	dinner rolls	hand soap	spaghetti sauce
4	juice	lunch meat	soda	toilet paper	all- purpose		
5	pasta	tortillas	mixes	hand soap	toilet paper	vegetables	vegetables
6	toilet paper	eggs	toilet paper	vegetables	bagels	dishwashing liqu	cereals
7	paper towels	tortillas	vegetables	milk	ice cream	juice	dishwashing liqu
8	juice	paper towels	waffles	soda	cheeses	poultry	toilet paper
9	poultry	coffee/tea	coffee/tea	dishwashing liquid/detergent			
10	coffee/tea	toilet paper	pork	flour	milk	soda	dishwashing liqu
11	ice cream	soda	bagels	dishwashing liqu	eggs	sugar	waffles
12	tortillas	pork	shampoo	lunch meat	pasta	juice	bagels
13	fruits	all- purpose	aluminum foil	laundry detergent	individual meals	flour	pork
14	dinner rolls	individual meals	shampoo	ketchup	cereals	sandwich bags	laundry detergent
15	ice cream	cereals	paper towels	bagels	mixes	lunch meat	juice
16	sandwich bags	flour	juice	milk	paper towels	cereals	sandwich bags
17	hand soap	pasta	individual meals	spaghetti sauce	cereals	sandwich loaves	hand soap
18	toilet paper	bagels	shampoo	coffee/tea			
19	laundry detergent	coffee/tea	eggs	aluminum foil	beef	juice	flour
20	dishwashing liqu	yogurt	juice	sugar	soap	sandwich loaves	butter

연관분석 데이터 전처리



Sales_id	Us...	DATE	History
10,827	010-...	2021-...	{ "item": [{ "id": 13129, "Qty": 2}, { "id": 12711, "Qty": 2}, { "id": 1116
10,828	010-...	2021-...	{ "item": [{ "id": 12279, "Qty": 2}, { "id": 11893, "Qty": 1}], "phonenum
10,829	010-...	2021-...	{ "item": [{ "id": 12121, "Qty": 2}, { "id": 13473, "Qty": 2}, { "id": 1266
10,830	010-...	2021-...	{ "item": [{ "id": 13557, "Qty": 1}], "phonenum": "010-9848-6984"}
10,831	010-...	2021-...	{ "item": [{ "id": 10882, "Qty": 2}], "phonenum": "010-9848-6984"}
10,832	010-...	2021-...	{ "item": [{ "id": 11053, "Qty": 1}, { "id": 10888, "Qty": 2}, { "id": 1189
10,833	010-...	2021-...	{ "item": [{ "id": 13129, "Qty": 2}, { "id": 13319, "Qty": 1}], "phonenum
10,834	010-...	2021-...	{ "item": [{ "id": 13129, "Qty": 2}, { "id": 12156, "Qty": 2}, { "id": 1212
10,835	010-...	2021-...	{ "item": [{ "id": 12279, "Qty": 2}, { "id": 12766, "Qty": 2}, { "id": 1088
10,836	010-...	2021-...	{ "item": [{ "id": 12205, "Qty": 2}], "phonenum": "010-9848-6984"}
10,837	010-...	2021-...	{ "item": [{ "id": 12279, "Qty": 1}, { "id": 11745, "Qty": 2}], "phonenum
10,838	010-...	2021-...	{ "item": [{ "id": 10888, "Qty": 2}, { "id": 13473, "Qty": 2}, { "id": 1109
10,839	010-...	2021-...	{ "item": [{ "id": 12389, "Qty": 1}, { "id": 10888, "Qty": 1}, { "id": 1088
10,840	010-...	2021-...	{ "item": [{ "id": 11496, "Qty": 2}, { "id": 13212, "Qty": 2}, { "id": 1159
10,841	010-...	2021-...	{ "item": [{ "id": 12766, "Qty": 1}, { "id": 10888, "Qty": 1}, { "id": 1364
10,842	010-...	2021-...	{ "item": [{ "id": 11091, "Qty": 2}, { "id": 12938, "Qty": 1}], "phonenum
10,843	010-...	2021-...	{ "item": [{ "id": 11132, "Qty": 2}], "phonenum": "010-9848-6984"}
10,844	010-...	2021-...	{ "item": [{ "id": 13473, "Qty": 1}], "phonenum": "010-9848-6984"}
10,845	010-...	2021-...	{ "item": [{ "id": 13473, "Qty": 1}], "phonenum": "010-9848-6984"}
10,846	010-...	2021-...	{ "item": [{ "id": 10888, "Qty": 1}], "phonenum": "010-9848-6984"}
10,847	010-...	2021-...	{ "item": [{ "id": 10888, "Qty": 2}, { "id": 11499, "Qty": 2}, { "id": 1159
10,848	010-...	2021-...	{ "item": [{ "id": 12389, "Qty": 2}, { "id": 11053, "Qty": 1}, { "id": 1149
10,849	010-...	2021-...	{ "item": [], "phonenum": "010-9848-6984"}
10,850	010-...	2021-...	{ "item": [{ "id": 13643, "Qty": 1}], "phonenum": "010-9848-6984"}
10,851	010-...	2021-...	{ "item": [{ "id": 13212, "Qty": 1}, { "id": 13473, "Qty": 2}], "phonenum
10,852	010-...	2021-...	{ "item": [{ "id": 13473, "Qty": 1}, { "id": 11745, "Qty": 2}], "phonenum

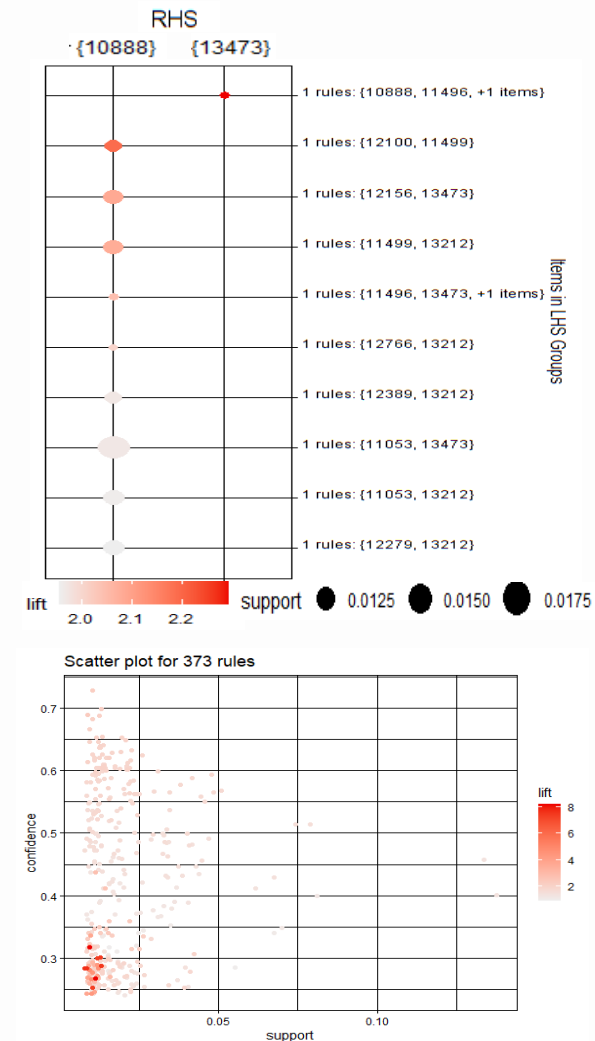
데이터와 연관분석

Grouped Matrix



제품 X	제품 Y	지지도	신뢰도	향상도
요거트	샐러드	0.080275	0.512965	1.542842
치즈	샐러드	0.072953	0.505501	1.520393
우유	샐러드	0.135762	0.449474	1.351881
샐러드	우유	0.135762	0.408331	1.351881
모닝롤	샐러드	0.079040	0.393500	1.183528
모닝롤	우유	0.070219	0.349583	1.157380
샐러드	치즈	0.080275	0.241443	1.542842
샐러드	모닝롤	0.079040	0.237729	1.183528
우유	모닝롤	0.070219	0.232477	1.157380
샐러드	치즈	0.072953	0.219422	1.520393

연관규칙 분석

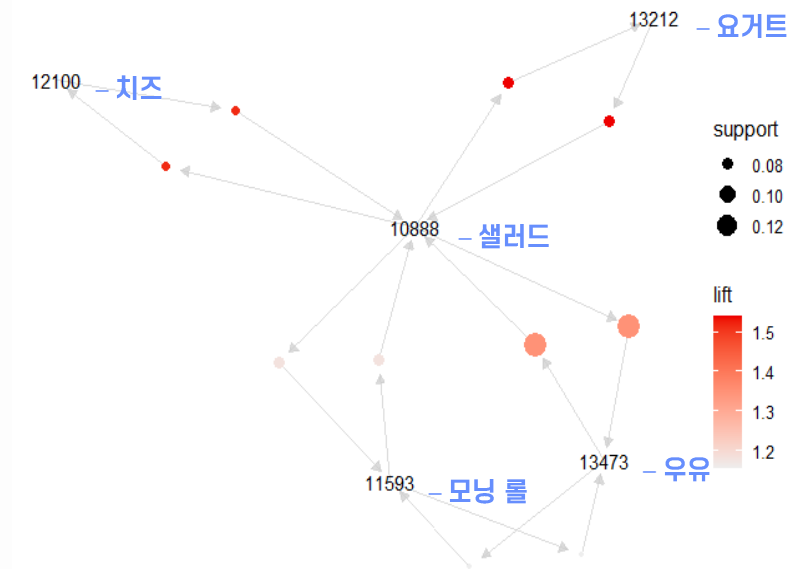


데이터와 연관분석

연관진열 마케팅 제안



Network graph



7조 스마트 무인 계산대

CLOUD

서버 생성 및 웹페이지 제작



ABOUT PROJECT

- 장고를 이용한 계산 페이지 구축
- 플라스크를 이용한 관리자 페이지 구축
- 서버리스를 위한 다양한 api 구축

서버 생성 및 웹페이지 제작



```
File Edit Selection View Go Run Terminal Help settings.py - Visual Studio Code
C:\Users\MIN> MIN > OneDrive > 바탕 화면 > multicampus_ai_casher > Cloud > Web > chatting > mysite > mysite > settings.py > ...
28 ALLOWED_HOSTS = ["*"]
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'channels',
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41     'chat',
42 ]
43
44 # Channels
45 ASGI_APPLICATION = 'mysite.routing.application'
46 CHANNEL_LAYERS = {
47     'default': {
48         'BACKEND': 'channels_redis.core.RedisChannelLayer',
49         'CONFIG': {
50             "hosts": [('127.0.0.1', 6379)],
51         },
52     },
53 }
54
55 MIDDLEWARE = [
56     'django.middleware.security.SecurityMiddleware',
57     'django.contrib.sessions.middleware.SessionMiddleware',
58     'django.middleware.common.CommonMiddleware',
59     'django.middleware.csrf.CsrfViewMiddleware',
60     'django.contrib.auth.middleware.AuthenticationMiddleware',
61     'django.contrib.messages.middleware.MessageMiddleware',
62     'django.middleware.clickjacking.XFrameOptionsMiddleware',
63 ]
64
65 ROOT_URLCONF = 'mysite.urls'
66
67 TEMPLATES = [
68     {
69         'BACKEND': 'django.template.backends.django',
70         'DIRS': [],
71         'APP_DIRS': True,
72         'OPTIONS': {
73             'context_processors': [
74                 'django.template.context_processors.debug',
75                 'django.template.context_processors.request',
76                 'django.contrib.auth.context_processors.auth',
77                 'django.contrib.messages.context_processors.messages',
78             ],
79         },
80     },
81 ]
```

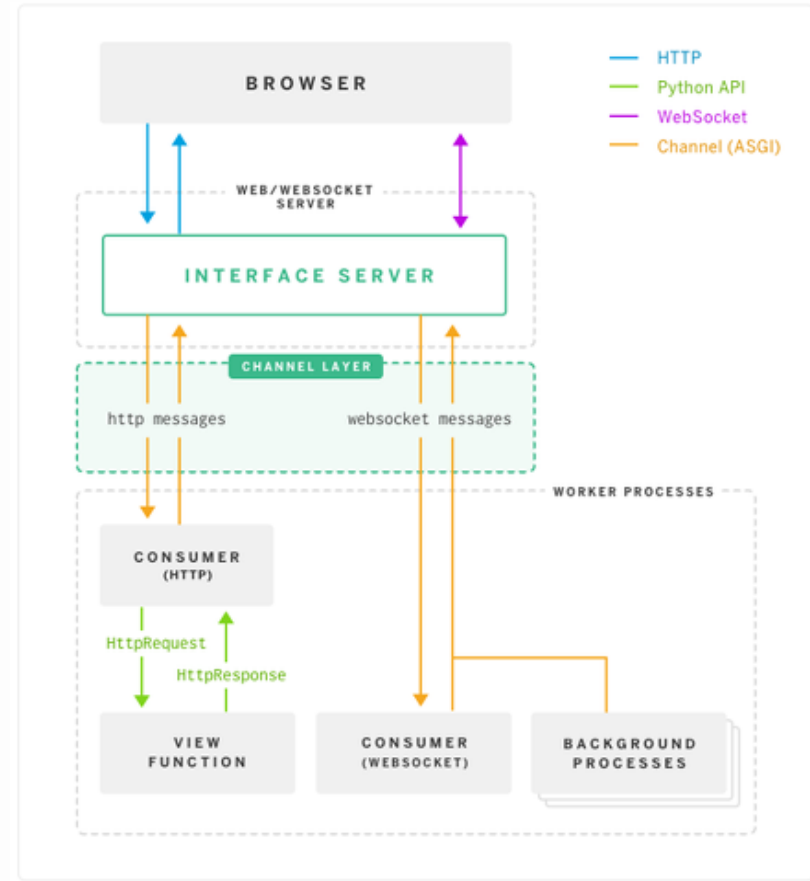
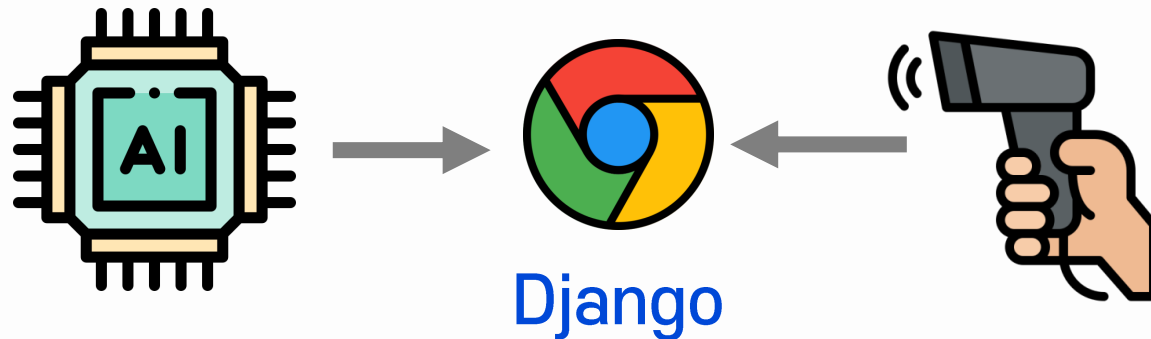
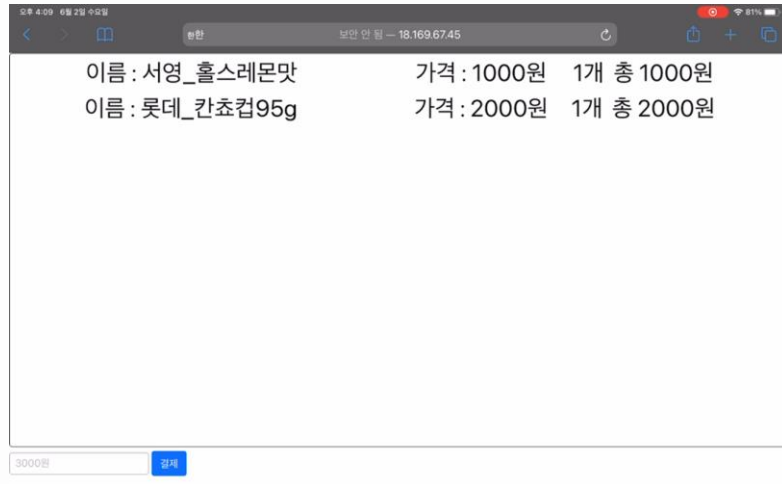
```
# Channels
ASGI_APPLICATION = 'mysite.routing.application'
CHANNEL_LAYERS = {
    'default': {
        'BACKEND': 'channels_redis.core.RedisChannelLayer',
        'CONFIG': {
            "hosts": [('127.0.0.1', 6379)],
        },
    },
}

Watching for file changes with
Performing system checks...

System check identified no issues (0 silenced).
June 03, 2021 - 20:50:35
Django version 3.2.3, using settings 'mysite.settings'
Starting ASGI/Channels version 3.0.1 development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
HTTP GET /chat/1234/ 200 [0.01, 175.196.19.85:62237]
WebSocket HANDSHAKING /ws/chat/1234/ [175.196.19.85:62239]
WebSocket CONNECT /ws/chat/1234/ [175.196.19.85:62239]
```

서버 생성 및 웹페이지 제작

계산 페이지



서버 생성 및 웹페이지 제작



관리자 페이지

484		["item": [{"id": "11751", "prob": 35, "outco		실시간 판매 현황 페이지	
		휴대폰 번호	날짜	영수증 정보	
		010-8462-8846	2021-05-01 00:00:00	12389번 1개 12279번 2개 12665번 1개 11745번 2개 13090번 2개 10888번 2개 11053번 1개 11593번 2개	
		010-8462-8846	2021-05-01 00:04:03	11132번 1개 10888번 1개 12100번 2개 13473번 2개 11132번 1개 10882번 1개 10888번 2개	
485		["item": [{"id": "11967", "prob": 0.598008,			
		010-8462-8846	2021-05-01 00:08:07	12389번 1개 13357번 2개 11593번 1개 12675번 1개 13473번 1개 10991번 1개 12539번 1개 13357번 2개	
아이				판매 관리	
1086					
1086				instant	
10866		햄_백햄롱롱버거	2900원	27개	instant
10867		주_백햄데리마요밥바	1900원	81개	instant

서버 생성 및 웹페이지 제작



- Api gateway

The screenshot displays the AWS API Gateway console interface. On the left, a sidebar shows the 'Resources' tree with a dropdown menu open, listing paths: '/', '/barcode', '/judgement', '/receipt', and '/result'. Each path is associated with an HTTP method (GET, POST, OPTIONS). The main panel is titled 'process_result' and shows a 'Function overview' section. It contains a diagram with two components: 'process_result' (represented by an orange lambda icon) and 'API Gateway' (represented by a purple icon). Below the diagram are buttons for '+ Add trigger' and '+ Add destination'. The 'process_result' component has a 'Layers' section with '(0)' layers listed.

7조 스마트 무인 계산대

최종 결과

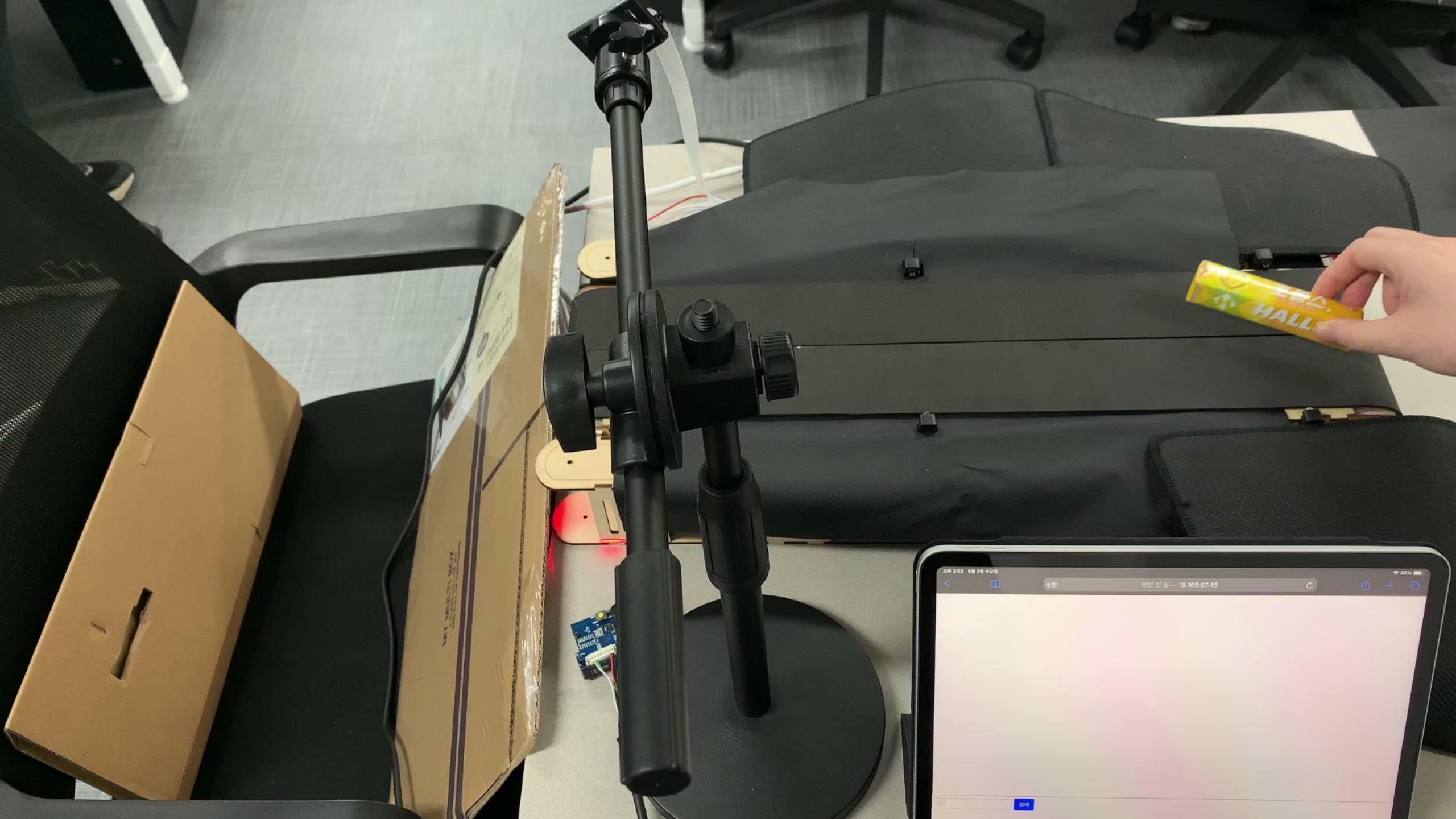
ABOUT PROJECT

- 시연 영상
- 발전 방향
- 애로사항
- 소감



프로젝트 수행 결과





프로젝트 수행 결과



애로사항

Ai

- 물건이 늘어날 때마다 학습시키기
→ 유지보수가 번거로움(시스템 문제)
- 배경에 영향 多
- 조명 반사에 영향 多

빅데이터

- 물건이 늘어날 때
→ 제품 정보 DB에 삽입 + 이미지 데이터 수집, 전처리로 인한 번거로움
- 이미지 데이터에서 제품의 윤곽선 검출
→ 배경과 제품의 색상/명도 차이 영향을 많이 받음

IoT

- 컨베이어 벨트 속도가 빠르면, 이미지 번짐 현상
- 카메라의 조리개 값의 영향을 많이 높은 스펙의 카메라가 요구됨
- 외부 서버와의 데이터 통신에 어려움
- 바코드 번호를 통해 websocket으로 제품의 정보를 클라우드 측으로 전송하는 데 어려움이 있었음

클라우드

- 타 전공들 간의 통신 문제 해결에 어려움
- 웹소켓을 이용한 장고 서버 구축에 어려움
- EC2 에서 외부로 신호 전송 시 CORS 설정에 대한 번거로움 및 어려움

프로젝트 수행 결과



발전방향

- 판매내역 데이터 베이스에 실제 편의점 구매데이터가 쌓일수록 한층 더 유의미한 연관분석 결과가 도출될 것
=> 효과적인 마케팅 기대
- 하드웨어 성능 개선 시 결제까지 소요되는 시간 단축
- 프론트 UI를 사용자 친화적으로 개선

프로젝트 수행 결과



소감

다양한 전공과 함께 프로젝트를 할 수 있어서 좋았습니다.
다른 분야와 연동 시 다양한 통신 방법을 사용해 볼 수 있어서 좋았고, 배웠던 내용들을 실제로 적용해 줄 수 있어서 뜻깊은 경험이었습니다.

모든 분야를 통합하는 과정이 생각보다 어려가 많이 발생해서 시간이 오래 걸렸습니다.

프로젝트를 진행하면서 배운 것들을 실제로 활용해볼 수 있어서 좋았고 각 분야를 융복합하는 과정에서 몰랐던 것들을 많이 배울수 있어서 너무 좋은 경험이었습니다

혼자서는 할 수 없는 프로젝트를 팀원들과 함께 해서 멋지게 마무리 할 수 있었습니다

타 분야와의 통합과정에서 어떠한 통신방법이 있고, 무엇을 고려해야 하는지 깨닫게 되는 뜻깊은 시간이었습니다.

아는 것이 서로 다르다보니 새로운 것을 알게 되었고 융합으로 진행하다보니 각 파트별 힘든 점을 이해할 수 있는 기회였습니다

처음에 막막해 보였던 프로젝트가, 각 분야의 팀원들의 힘이 모여 완성된 것을 보며, 참 뿌듯하고 기쁩습니다.



감사합니다

[K-Digital] 멀티캠퍼스 7조
우리 똑똑하조