Food Sharing Service

“Share and Enjoy Fresh Ingredients with Your Community”

Korea University, DAB 니꺼내꺼팀

이영빈 2019120004
박혜빈 2019130866
박진태 2019120059
손영진 2020120083
윤서현 2020250479

Oct 21, 2024

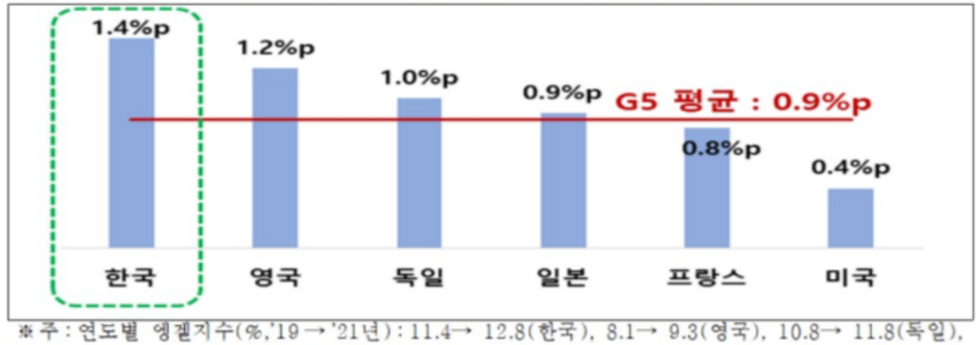
Contents

문제 상황 정의

최근 농산물의 높은 물가 상승률로 식재료 구매에 대한 부담이 발생하고 있으며,
현재 20대 1인가구가 늘어나고 있는 동시에 2030 청년의 건강 악화 문제 심각해지고 있음

/ 농산물의 높은 물가 상승률로 인한 식생활 부담 증가 /

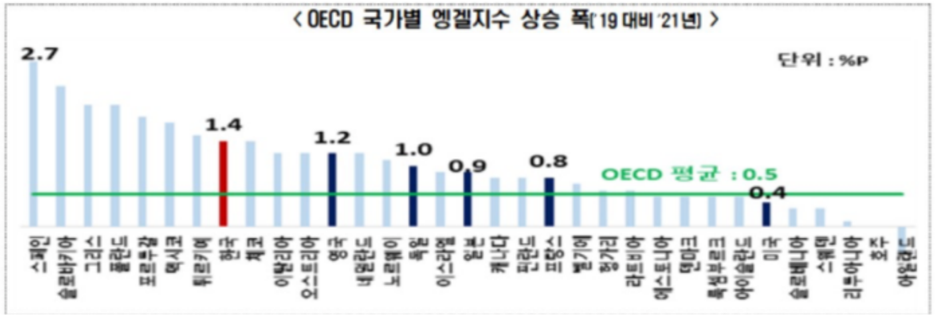
<주요국(G5+한국) 엥겔지수 상승 폭('19 대비 '21년)>



※주 : 연도별 엥겔지수(%,'19→'21년) : 11.4→ 12.8(한국), 8.1→ 9.3(영국), 10.8→ 11.8(독일), 15.4→ 16.3(일본), 13.1→ 13.9(프랑스), 6.3→ 6.7(미국)

※자료 : OECD 국민계정 DB(단, 일본은 '21년 자료가 부재하여 日내각부 통계 사용)

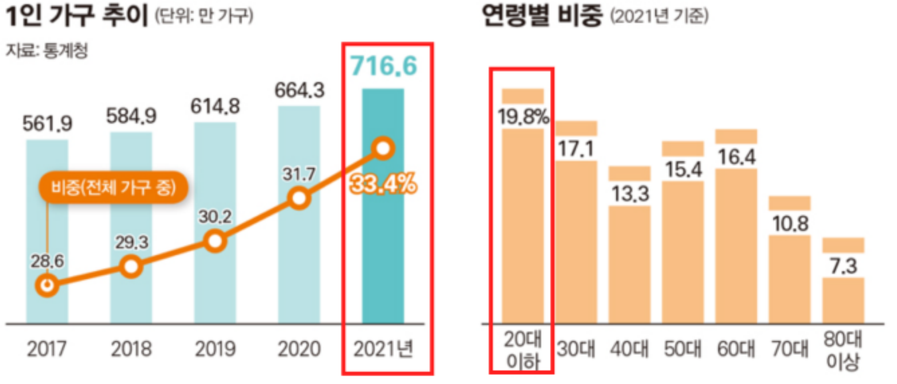
<OECD 국가별 엥겔지수 상승 폭('19 대비 '21년)>



<주요국 엥겔지수 상승폭(출처: 한경연)>

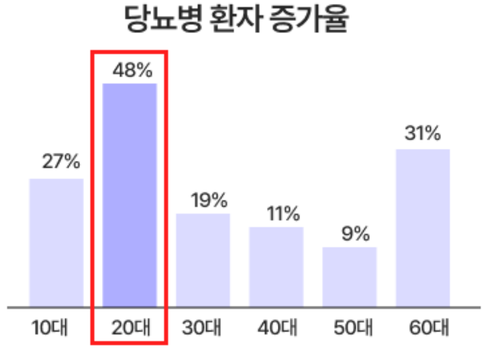
엥겔지수 : 총가계 지출액 중에서 식료품비가 차지하는 비율

/ 20대 1인가구의 증가와 2030 건강 악화 /



< 통계청 2021년 인구주택총조사 결과 >

전체 가구의 약 33%가
1인 가구이며,
1인 가구 연령대별 비중에서
20대 비율이 가장 높음



< 건강보험심사평가원 2022 발표 결과 >

< 배달 대체 밀키트에 대한 높은 수요 >

20대 1인 가구 외식 비율 ↑,
2030 당뇨병 유병률 증가 추세
→ 건강한 식단에 대한 수요 ↑

식재료의 수요와 공급의 미스매치, "Food Loss Problem"

공급단계

- 수확전 농산물:
  부적절한 취급으로 인한 공급사슬 내 부패 농산물
- 가격안정을 목적으로 폐기되는 농산물(매몰)
- 시장기준을 충족하지 못해 폐기되는 농산물
- 식품 제조·가공 시 모양이나 조리 관행으로
  폐기되는 식재료의 가식부(可食部)

↓

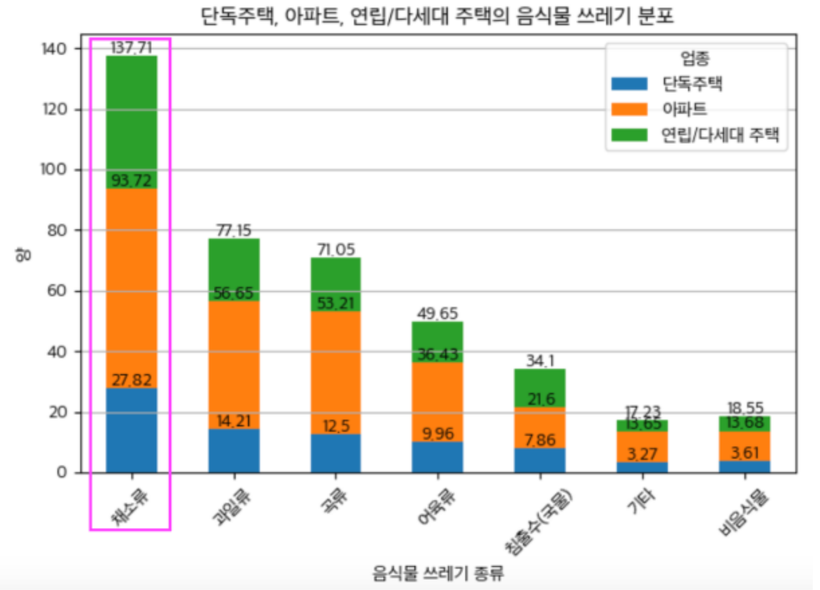
유통단계

- 소비기한, 품질유지기한 경과(임박) 식품
- 유통 중 품질저하식품(보관, 유통, 전시 등)

↓

소비단계

- 소비자 공급(배식) 전 음식
- 가정 내 음식
- 가정·외식·급식산업 공정단계의 가식부 폐기



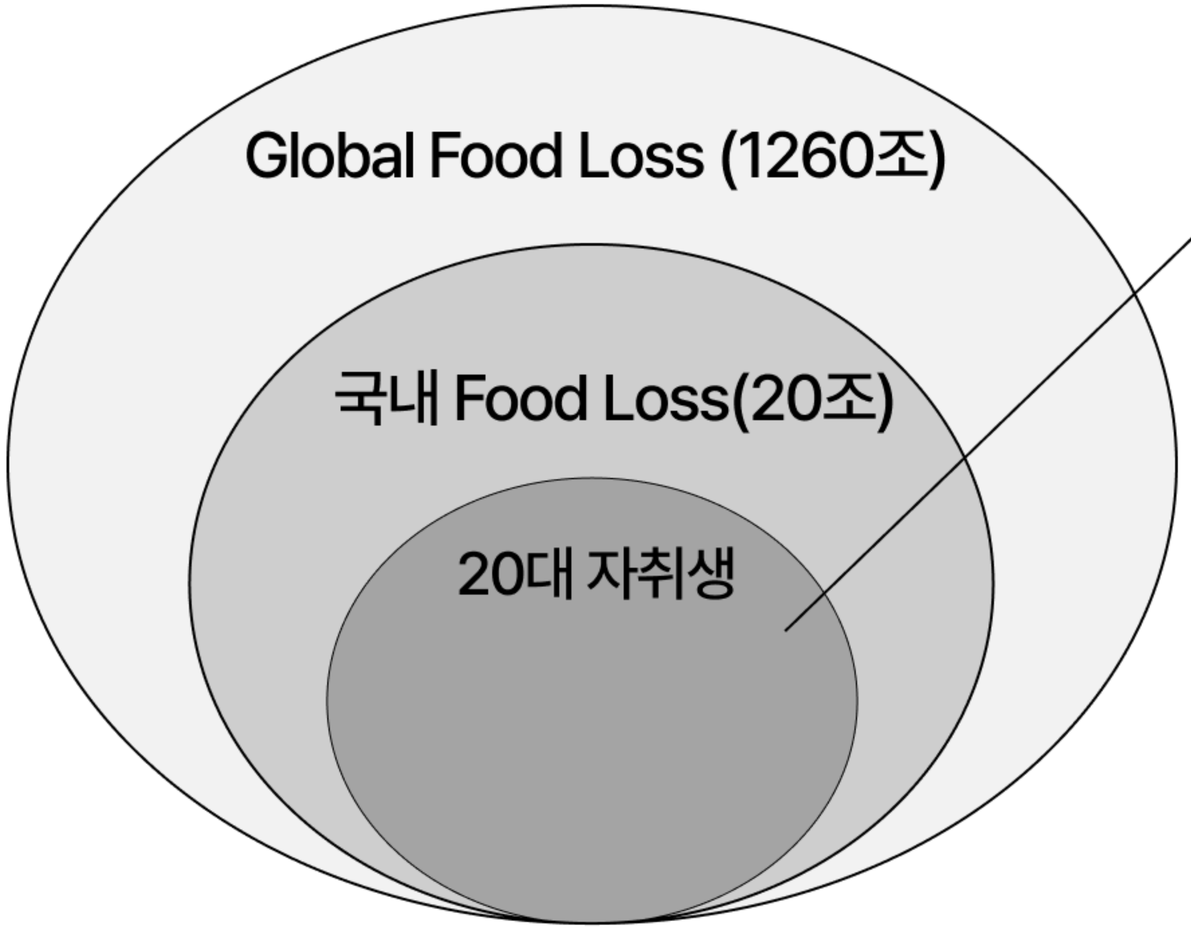
한국의 경우 국내 공급 농식품 가운데 약 14%가 폐기되고,
이로 의한 경제적 비용은 **20조원**(2018년 기준)에 달하는 것으로 추정됨

a. 유통·조리 과정에서 발생하는 음식물 : 57%
b. 먹고 남긴 음식물 : 30%
c. 보관하다 폐기하는 음식물이 9%

* 2019년 기준 농식품 폐기량은 약 500만t에 달함

목표 시장 선정

실현가능성 + 하이퍼로컬에 집중하여 목표 시장 선정



왜 20대 자취생이 타겟인가?

자취생에게 집중한 이유

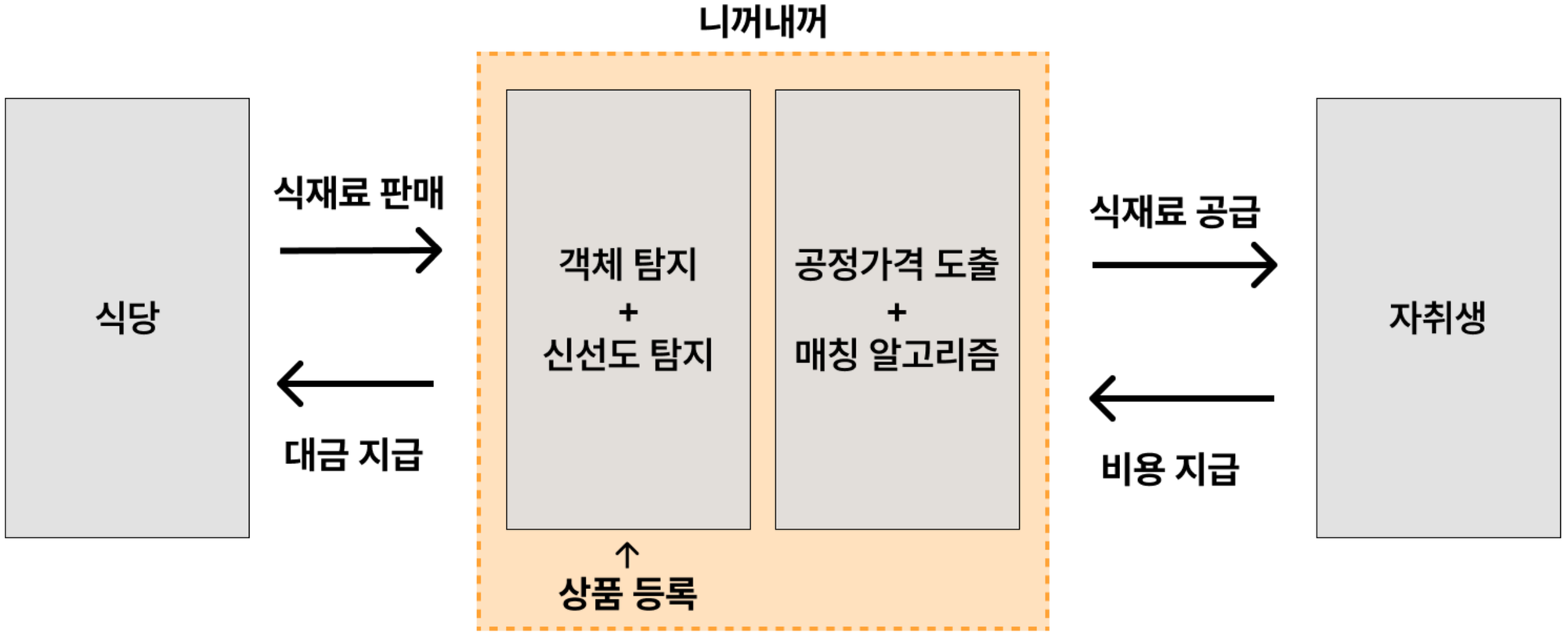
- 잦은 외식으로 인해 높아진 20대 건강 문제
- 건강한 식단에 대한 수요↑

하이퍼로컬인 이유

- 지역 신뢰를 기반으로 신선도 리스크 관리
- 남은 유통기한과 비례하는 유통 거리 문제

니꺼내꺼 비즈니스 모델

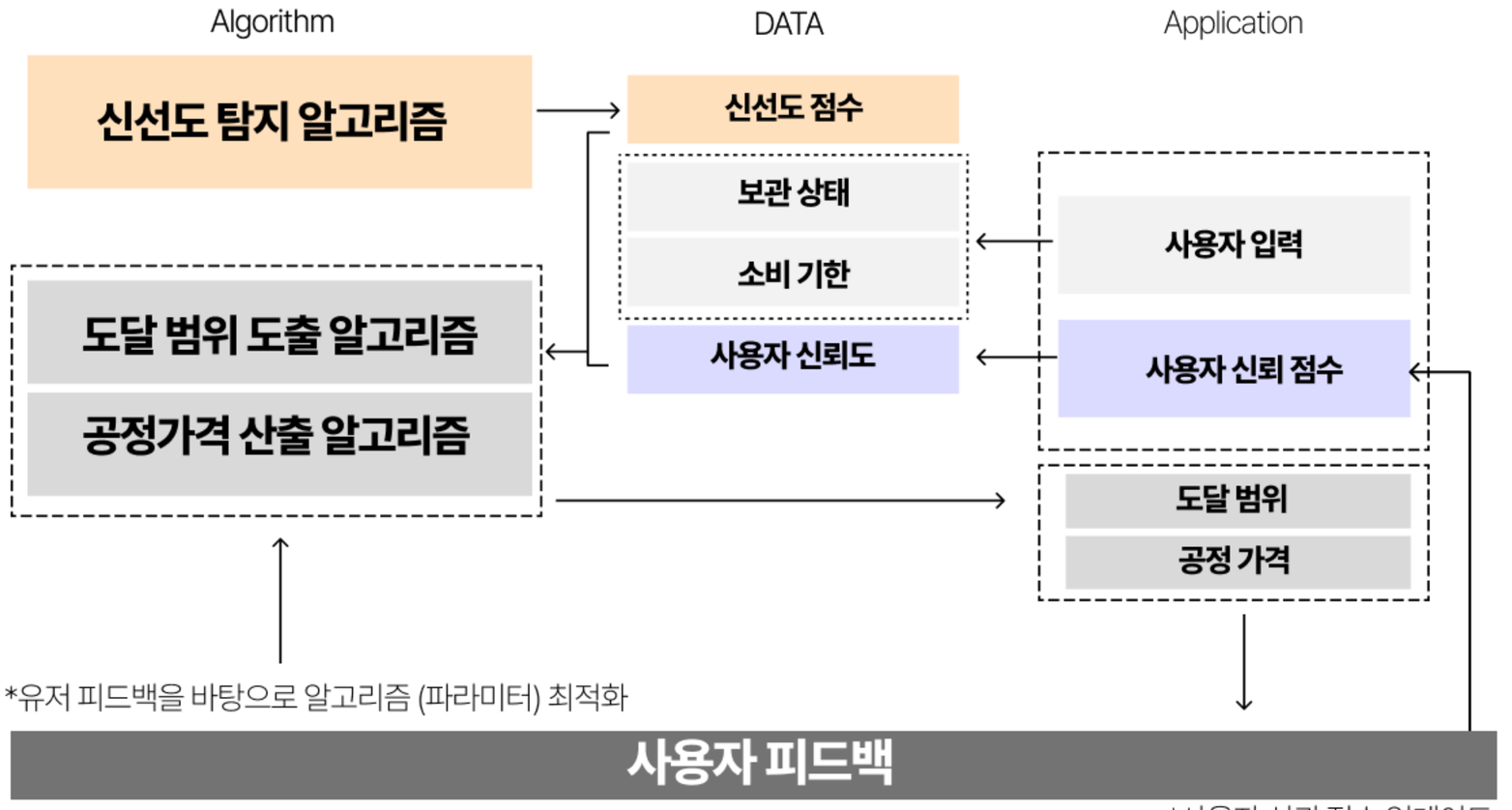
"유통/소비단계에서의 Food Loss를 줄이면서 20대 자취생들의 건강과 돈을 지켜주자"



Cold Chain 문제와 Hyperlocal 매칭 알고리즘

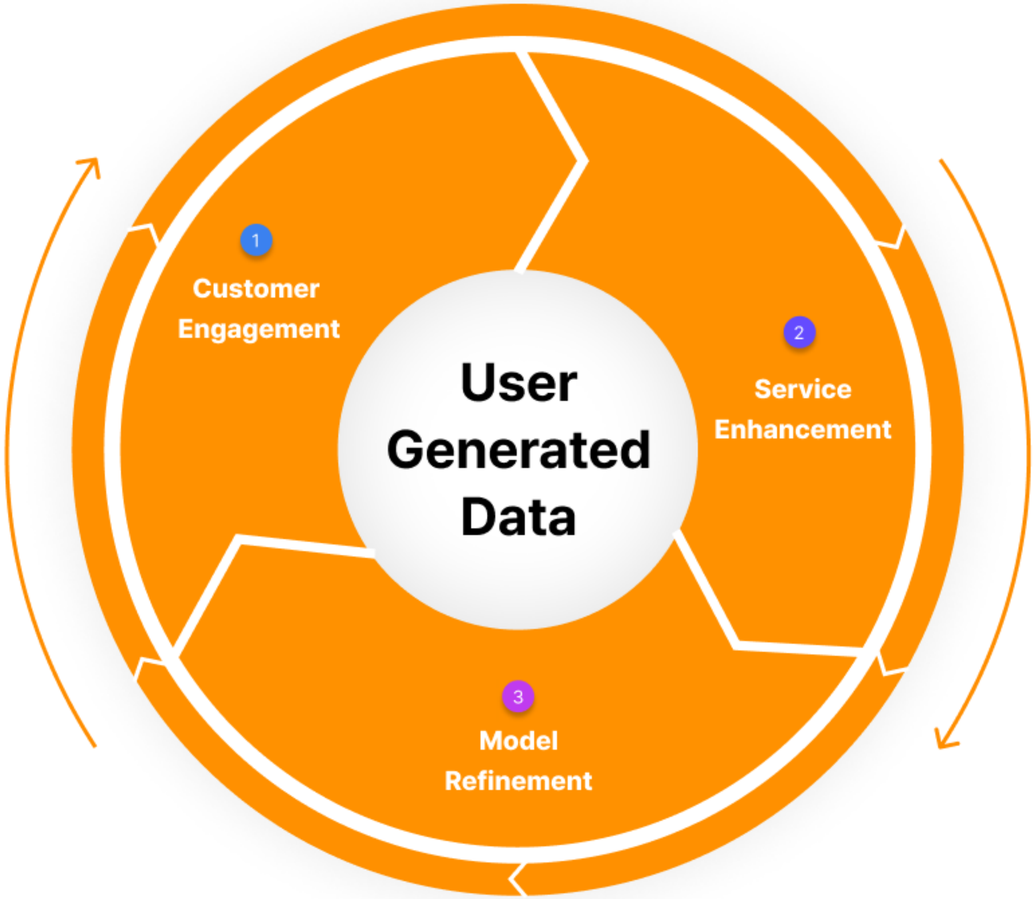
Cold Chain 문제: 식재료가 운송 및 보관 과정에서 신선도가 저하되고 품질이 악화되는 상황

신선도 기반 '도달 가능 범위' 를 도출하는 알고리즘 활용



Flywheel Structure

플랫폼은 신선도 탐지 모델을 기반으로 사용할수록 더욱 정확한 정보를 제공하며,
이를 통해 공동체에 저렴하고 신선한 식재료를 소비할 기회를 제공



1 Customer Engagement

사용자들이 플랫폼을 이용하면서 사진을 업로드하고, 식재료에 대한 정보를 제공하며, 후기를 남기는 등의 총체적인 데이터가 축적됨에 따라, 플랫폼의 전체적인 기능이 최적화됩니다. 이러한 데이터는 신선도 탐지 모델을 더욱 정교하게 만들어, 신선 식재료의 상태를 정확하게 판단하는 데 중요한 역할을 합니다.

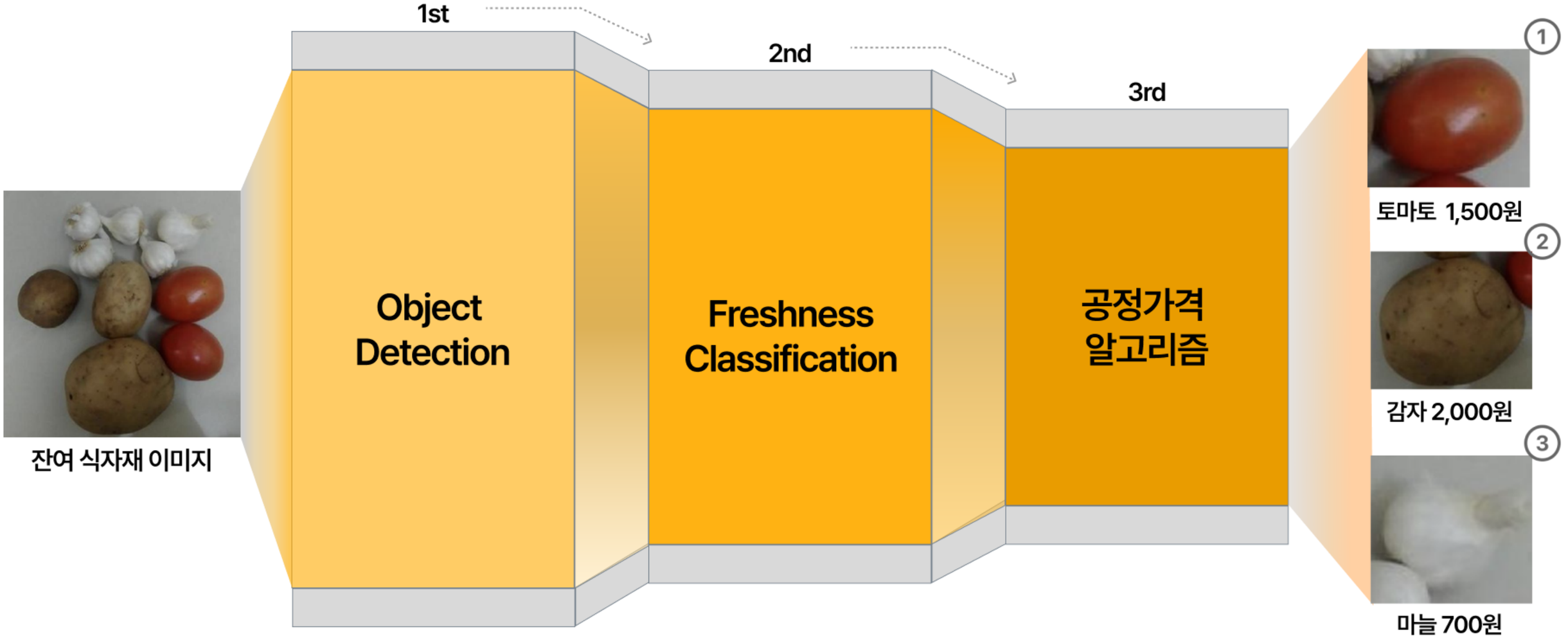
2 Service Enhancement

플랫폼은 지속적으로 사용자 피드백과 데이터를 분석하여, 더욱 개선된 사용자 경험을 제공합니다. 이 과정에서 유통기한이 임박한 신선 식재료를 더 효율적으로 공유할 수 있는 새로운 기능과 빠르고 정확한 매칭 시스템을 도입하여, 사용자들의 편리성을 높입니다.

3 Model Refinement

신선도 탐지 모델은 사용자가 플랫폼을 사용할수록 더 많은 데이터를 수집하고 학습함으로써, 시간이 지남에 따라 더욱 정교해집니다. 현실 속 이용자들이 계속해서 사용해서 제공하는 이미지 데이터를 통해 더 많은 종류의 식재료를 더욱 정교하게 탐지할 수 있고, 그들이 입력하는 부가 정보의 신뢰도, 유통 과정에서의 최대도달범위를 더욱 정교하고 정확하게 제공 가능해집니다.

최종 파이프라인


1st
2nd
3rd
Object Detection
Freshness Classification
공정가격 알고리즘
잔여 식자재 이미지
1
토마토 1,500원
2
감자 2,000원
3
마늘 700원

Object Detection

기존 연구 한계

•적은 식자재 카테고리
•적은 데이터셋 (약 10,000개 이하)
•확장성 & 실용성 높지 않음

해당 모델 데이터

roboflow

•단일 식자재 이미지 69,000개 수집
•복합 식자재 이미지 12,000개 수집

↓

일상에서 자주 접하는 40여개 식자재 라벨 정리
(사과, 바나나, 오이, 당근, 감자, 돼지고기 등)

Object Detection

최종 Object Detection 모델

69,000여개의 단일 식자재 이미지셋으로
YOLO_V8 Fine-Tuning

적은 학습만으로도 92% 이상의 정확도 달성

Fine-Tuned YOLO를 복합 식자재 이미지셋에 대해 추가적으로 Fine-Tuning

인간이 작은 지식부터 큰 지식으로 순차적으로 학습하는 방식과 유사하게, Deep Neural Network 또한 단계적인 Fine Tuning을 거칠 때, 더욱 정교해질 수 있을 것이라는 가설

소고기, 버섯, 소시지 등 몇 개의 Label을 제하고는 대부분 90% 정확도 달성

```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
45/50      4.38G     0.5844     0.4285      1.052         26        640: 100%|██████████| 406/406 [00:48<00:00,  8.31it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 6/6 [00:01<00:00,  4.61it/s]
                      all        341        839      0.896      0.879      0.922      0.753

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
46/50      4.38G     0.5753     0.4153      1.045         24        640: 100%|██████████| 406/406 [00:48<00:00,  8.32it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 6/6 [00:01<00:00,  4.87it/s]
                      all        341        839      0.897      0.874       0.92      0.752
```

Object Detection

최종 Object Detection 모델

69,000여개의 단일 식자재 이미지셋으로
YOLO_V8 Fine-Tuning

적은 학습만으로도 92% 이상의 정확도 달성

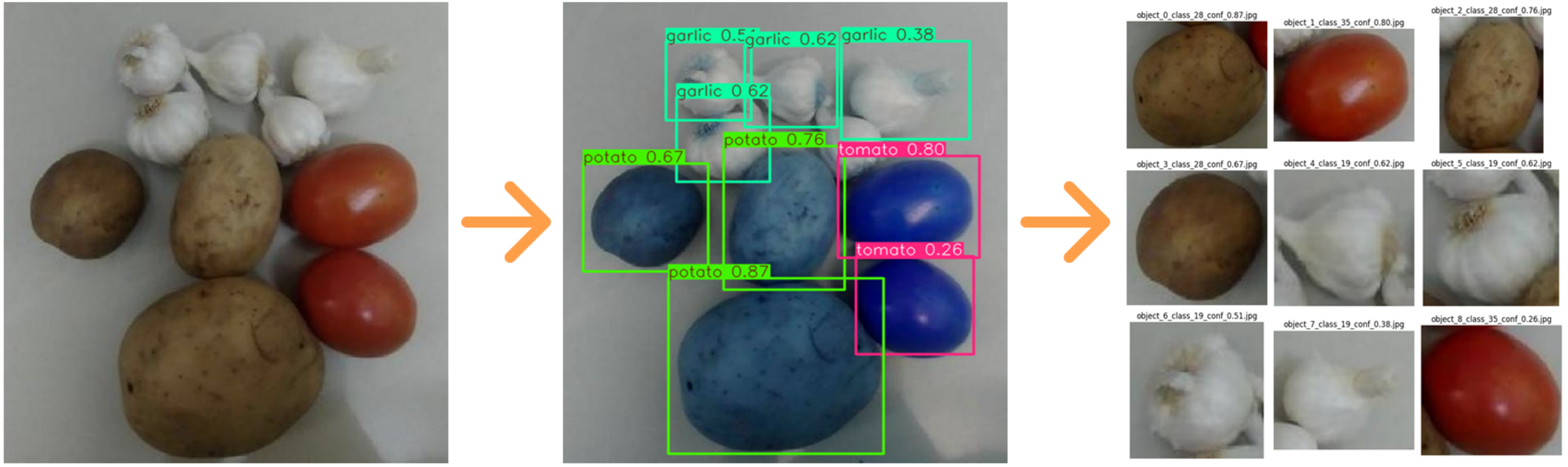
Fine-Tuned YOLO를 복합 식자재 이미지셋에 대해 추가적으로 Fine-Tuning

인간이 작은 지식부터 큰 지식으로 순차적으로 학습하는 방식과 유사하게, Deep Neural Network 또한 단계적인 Fine Tuning을 거칠 때, 더욱 정교해질 수 있을 것이라는 가설

소고기, 버섯, 소시지 등 몇 개의 Label을 제하고는 대부분 90% 정확도 달성

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	599	4761	0.789	0.662	0.719	0.531
apple	60	171	0.892	0.924	0.943	0.608
banana	51	120	0.875	0.867	0.894	0.551
beaf	5	5	0.507	0.2	0.448	0.38
bean_sprouts	1	1	0.6	1	0.995	0.697
bellpepper	164	176	0.938	0.932	0.945	0.75
brocoli	7	7	1	0.406	0.587	0.439
cabbage	4	4	0.671	0.75	0.787	0.651
carrot	150	188	0.882	0.856	0.898	0.642
chicken	152	473	0.963	0.915	0.957	0.774
chili	71	91	0.862	0.912	0.92	0.623
corn	2	2	0	0	0	0
cucumber	82	149	0.878	0.852	0.901	0.529
egg	6	10	0.582	0.5	0.628	0.511
eggplant	23	24	0.983	0.958	0.959	0.845
garlic	315	517	0.962	0.965	0.979	0.717
ginger	107	161	0.961	0.988	0.994	0.868
green_onion	16	22	0.608	0.682	0.756	0.421
kimchi	1	1	0.319	1	0.995	0.796
lettuce	2	2	1	0	0	0
mushroom	14	47	0.462	0.17	0.31	0.236
onion	372	665	0.91	0.938	0.957	0.742
orange	40	152	0.909	0.947	0.959	0.606
pork	207	782	0.867	0.895	0.927	0.627
potato	270	519	0.921	0.939	0.946	0.732
pumpkin	9	9	0.903	0.667	0.799	0.622
radish	2	2	1	0	0.526	0.476
sausage	2	6	1	0	0.03	0.012
shrimp	2	2	0	0	0	0
sweet_potato	2	2	1	0	0	0
tofu	1	1	0.923	1	0.995	0.597
tomato	132	449	0.95	0.93	0.962	0.727
tuna can	1	1	0.936	1	0.995	0.796

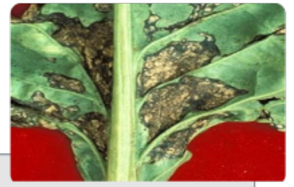
Object Detection



Object Detection 모델로 각 객체를 높은 정확도로 탐지한 후,
탐지된 객체별로 이미지를 Cropping하는 알고리즘까지 구축함

Freshness Classification

육류	
신선할 때	상했을 때
붉은색이 선명하고 지방이 흰색 / 크림색이고, 표면에 윤기남	색이 어두워지고 갈색 / 회색 / 녹색으로 변색되며 표면이 끈적해짐



채소류	
신선할 때	상했을 때
단단하고 색이 균일	표면이 마르거나 갈라짐, 변색 발생

어패류	
신선할 때	상했을 때
비늘이 반짝이고 살색이 선명, 눈이 투명, 살이 탄력적	비늘이 흐려지고 눈과 색이 탁해짐, 살이 무름

과일류	
신선할 때	상했을 때
밝고 생기 있는 색상, 표면이 윤기 있고 단단함	색이 흐려지고 갈변, 표면이 주름지고 물러짐

Comi, G. (2017). Spoilage of meat and fish. The Microbiological Quality of Food. / Choi, CH. et al. (2017). Prediction of beef freshness attributes using reflectance spectroscopy. Eng Agric Environ Food. / Yuan, Y., Chen, J., Polat, K., & Alhudhaif, A. (2024). An innovative approach to detecting the freshness of fruits and vegetables through the integration of convolutional neural networks and bidirectional long short-term memory network. Current Research in Food Science, 8, 100723. / Schifferstein, H. N. J. (2024). Changes in appearance during the spoilage process of fruits and vegetables: Implications for consumer use and disposal. Department of Human Centered Design, Delft University of Technology, Delft, the Netherlands

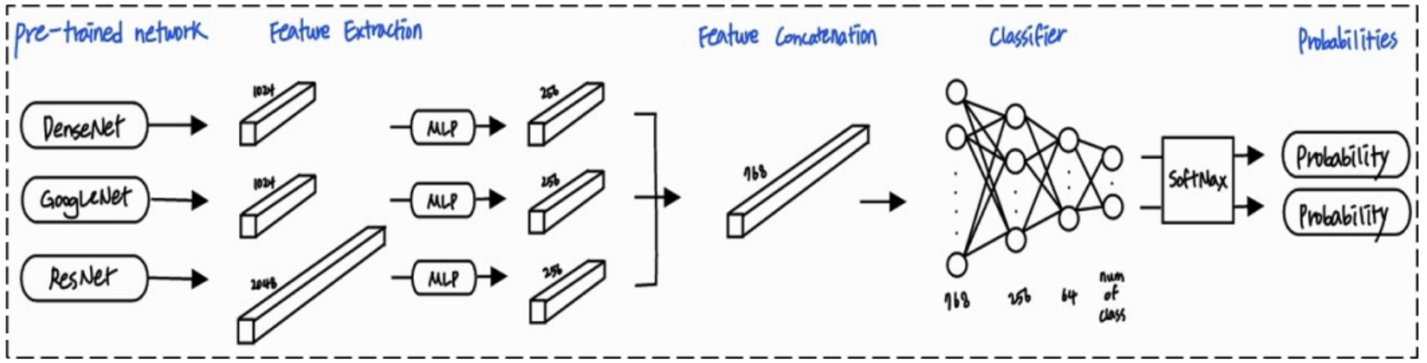
Freshness Classification

육류

채소류

과일류

동일 아키텍쳐



동일 아키텍쳐

Freshness Classification

/ 분류 /	/ 학습 이미지 /	/ 분류 모델 /
육류	• Fresh Meat **6,284 Images** • Spoiled Meat **5,324 Images**	**Binary Classification**
채소류	• Fresh_Potato, Fresh_Carrot, Fresh_Bellpepper, Fresh_Lettuce, Fresh_Onion, Fresh_Cucumber **4,111 Images** • Rotten_Potato, Rotten_Carrot, Rotten_Bellpepper, Rotten_Lettuce, Rotten_Onion, Rotten_Cucumber **4,335 Images**	**12-way Classification**
과일류	• Fresh Apple, Fresh Orange, Fresh Banana, Fresh Tomato **5,499 Images** • Rotten Apple, Rotten Orange, Rotten Banana, , Rotten Tomato **6,772 Images**	**8-way Classification**

Freshness Classification

기존 연구
대부분 6개 이하의 식자재에 대해 신선도 탐지

우리 모델
객체 탐지한 40여개의 Label 중, **13개 Label에 대해 신선도 탐지**

/ 확장성, 실용성 측면 개선 /

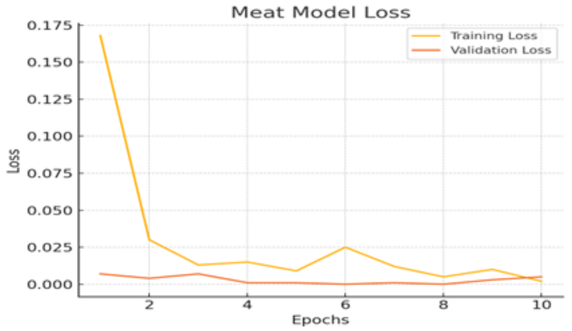
CNN에서 Downstream MLP로
이루어지는 단일 아키텍처에 대해
Softmax Score를 반환하는
출력 층의 노드 수만 늘리고,
새로운 식자재 Label을 추가한
데이터셋으로 모델을 파인튜닝하면,
언제든지 새로운 식자재에 대해서도
신선도를 탐지하도록 파이프라인 설계

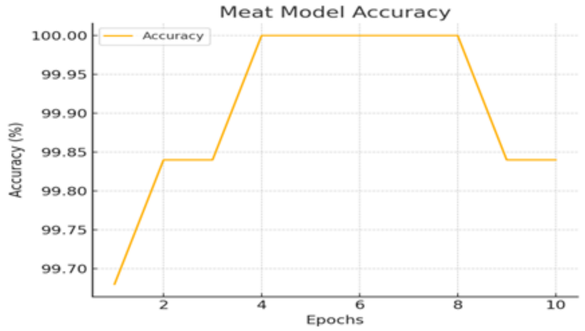
```python
'''모델 선언'''
class FreshnessDetectionModel(nn.Module):
    def __init__(self, num_classes, lr=0.001):
        super(FreshnessDetectionModel, self).__init__()

        # 사전 학습된 모델 불러오기
        self.densenet = models.densenet121(weights=DenseNet121_Weights.IMAGENET1K_V1)
        self.googlenet = models.googlenet(weights=GoogLeNet_Weights.IMAGENET1K_V1)
        self.resnext = models.resnext50_32x4d(weights=ResNeXt50_32X4D_Weights.IMAGENET1K_V1)

        # 분류기 부분 제거하여 특징 추출기로 사용
        self.densenet.classifier = nn.Identity()
        self.googlenet.fc = nn.Identity()
        self.resnext.fc = nn.Identity()

        # 사전 학습된 모델의 파라미터를 고정 (freeze)
        for param in self.densenet.parameters():
            param.requires_grad = False

        for param in self.googlenet.parameters():
            param.requires_grad = False

        for param in self.resnext.parameters():
            param.requires_grad = False
```

```python
        # 각 모델의 출력 특징을 처리하는 MLP
        self.mlp_densenet = nn.Sequential(
            nn.Linear(1024, 256),
            nn.ReLU()
        )

        self.mlp_googlenet = nn.Sequential(
            nn.Linear(1024, 256),
            nn.ReLU()
        )

        self.mlp_resnext = nn.Sequential(
            nn.Linear(2048, 1024),
            nn.ReLU(),
            nn.Linear(1024, 256),
            nn.ReLU()
        )

        # 최종 분류기
        self.final_classifier = nn.Sequential(
            nn.Linear(256 * 3, 256),
            nn.ReLU(),
            nn.Linear(256, 64),
            nn.ReLU(),
            nn.Linear(64, num_classes)  # Softmax는 적용하지 않음
        )
```

Freshness Classification

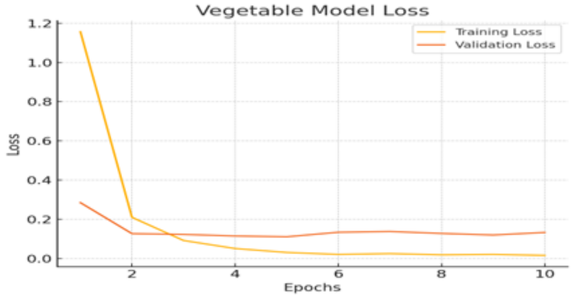
육류

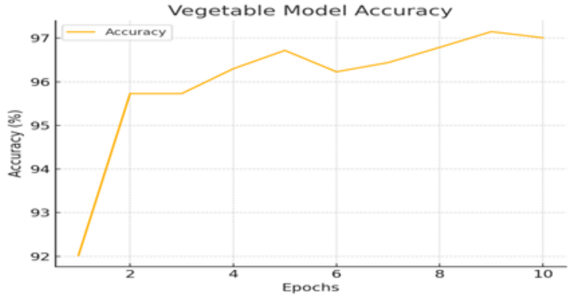




99.8%
Accuracy

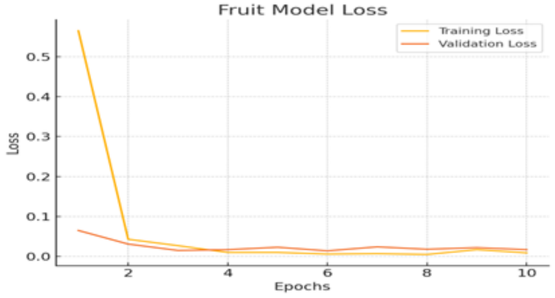
채소류

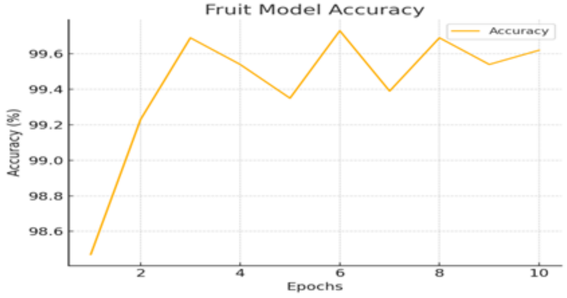




97%
Accuracy

과일류

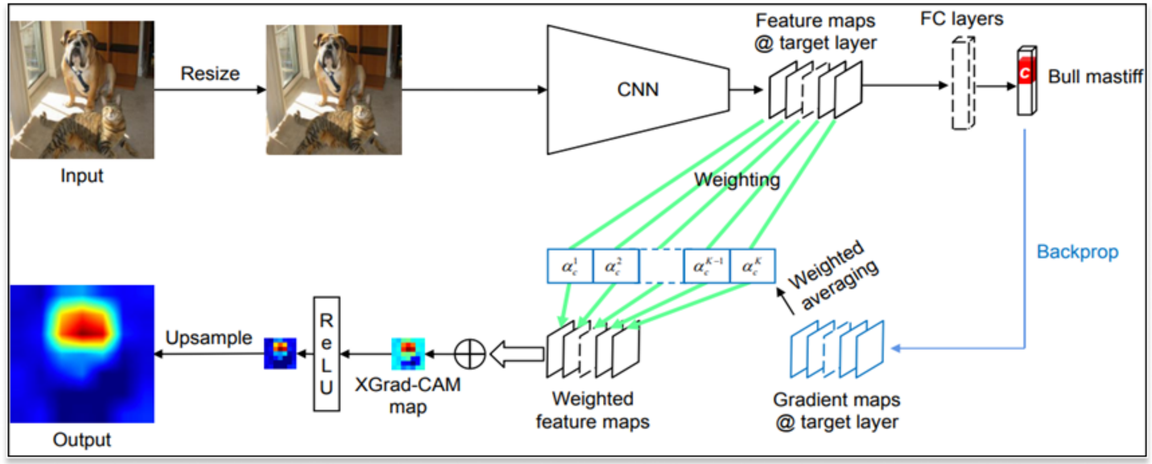




99.6%
Accuracy

Freshness Classification

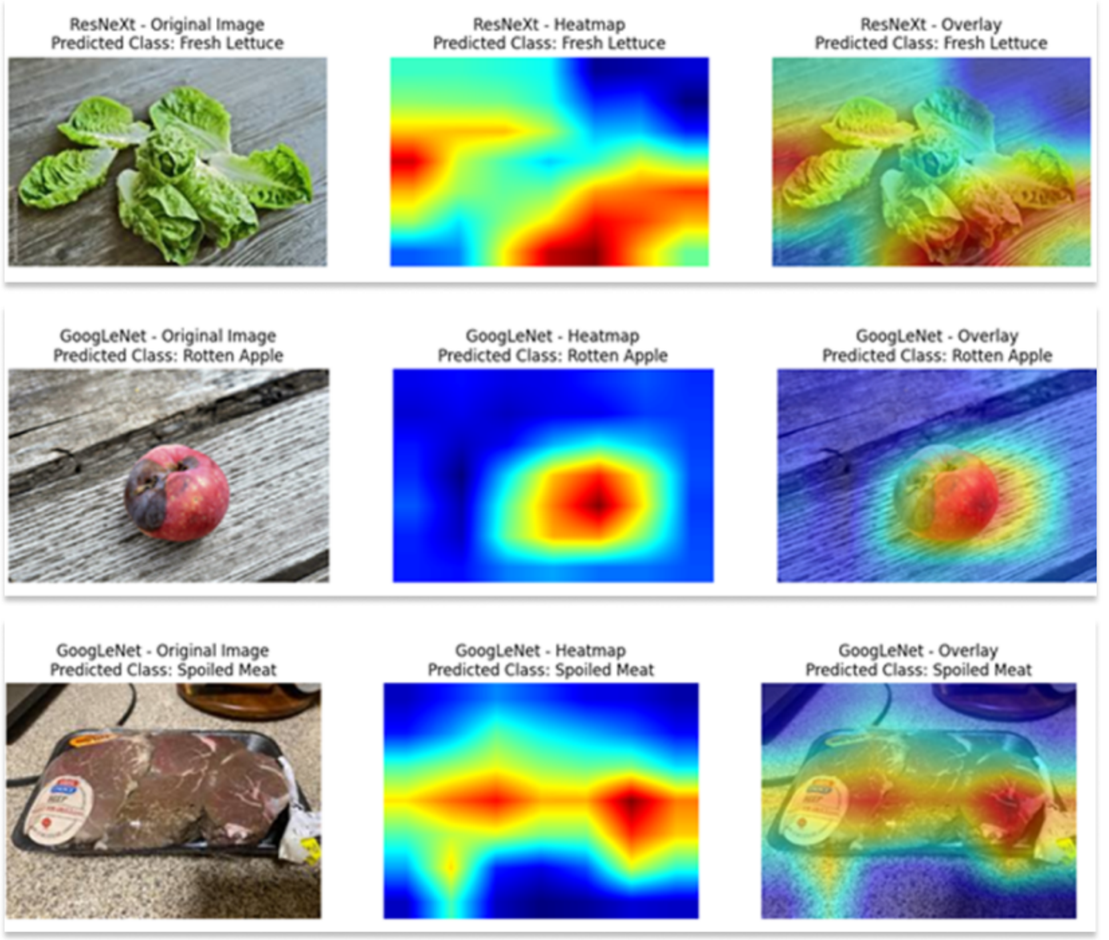
/ Gradient CAM 개념 /



“ Convolutional Neural Network가 이미지의
어떤 시각적 특징에 기반하여 판단을 내렸는지 시각화하는 기법 ”

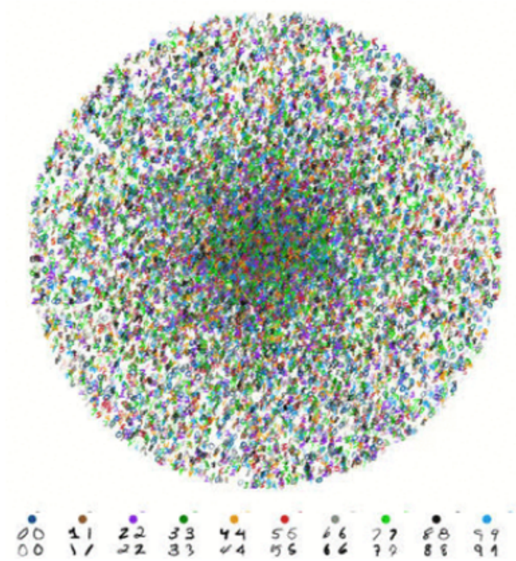
1. 최종 고차원 Feature Space에서 각 Feature의 중요도 계산
   : CNN의 마지막 Convolution layer에서 얻은 고차원 Feature map과
   predicted linear class score에 대한 기울기 기반으로 계산
2. 고차원 Feature Map을 종합하여 H*W의 2차원 Class Activation HeatMap 계산
3. HeatMap을 원본 이미지의 크기에 맞추어 Interpolation 및 Overlay

/ Gradient CAM 적용 결과 /



Freshness Classification

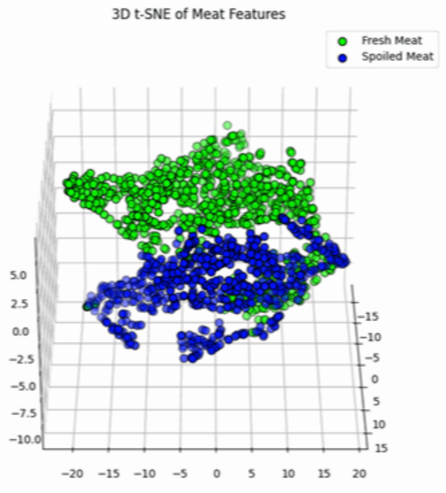
/ T-SNE 개념 /



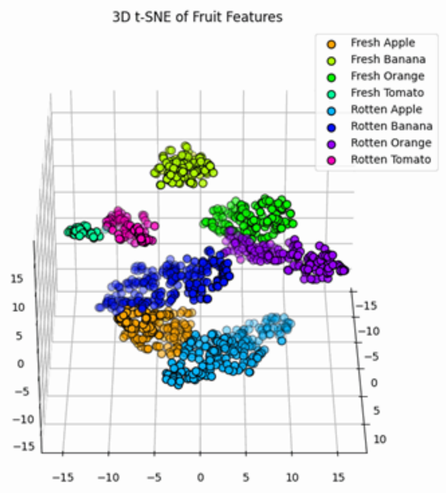
“ 고차원 데이터를 저차원 공간에 시각적으로 표현하기 위한 비선형 차원 축소 기법 ”

1. 데이터 포인트간의 유사도를 고차원에서는 가우시안 분포를, 저차원에서는 T-분포를 사용하여 계산
2. 가우시안 분포와 T-분포 간의 차이로서의 Kullback-Leibler Divergence를 최소화

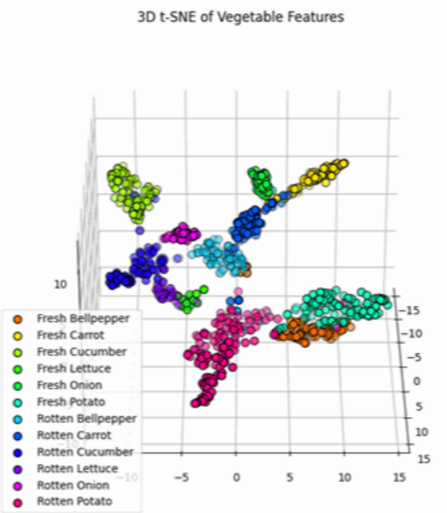
/ T-SNE 적용 결과 /



[육류]



[과일류]



[채소류]

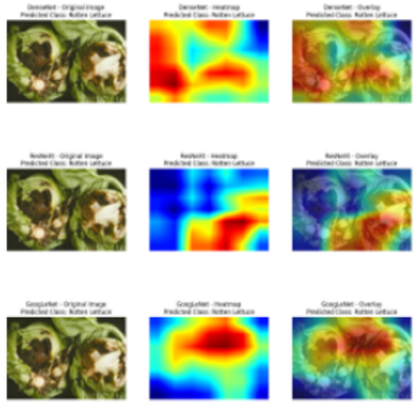
768 차원의 고차원 Feature Space에서도,
각 식자재의 신선도와 관련된 시각적 특징이 Cluster를 이루고 있음을 확인

즉, 전체 아키텍처가 식자재 이미지로부터
유의미한 시각적 특징을 학습하고 있음을 증명

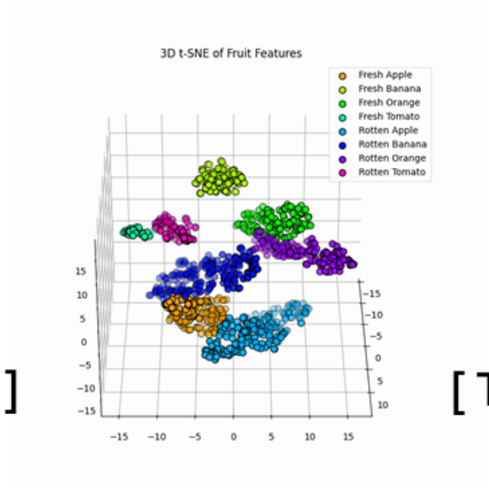
Freshness Classification

비즈니스 문제 해결과 후속연구 과정에서 모델의 실용성을 더욱 증진
→ 서비스 출시 / 연구 및 코드 공개 등을 통해 보다 폭넓은 문제에 적극적으로 기여

민감한 식료품 Domain에서의 Explainability를 달성
→ 신선한 식자재가 소비자에게 공정하게 전달되게끔 유도하여, 비즈니스 문제 해결



[G-CAM]



[T-SNE]

E.O.D

Korea University, DAB 니꺼내꺼팀

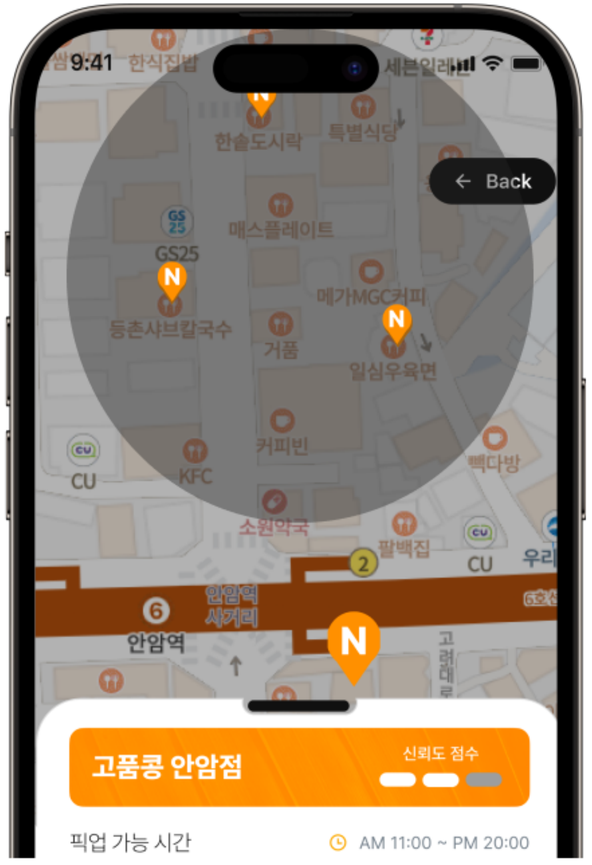
이영빈 2019120004
박혜빈 2019130866
박진태 2019120059
손영진 2020120083
윤서현 2020250479

Oct 21, 2024

콜드 체인 문제와 하이퍼로컬 매칭 알고리즘

콜드 체인 문제: 식재료가 운송 및 보관 과정에서 신선도가 저하되고 품질이 악화되는 상황

↓



신선도 기반 '도달 가능 범위'를 도출하는 알고리즘 활용

변수 설명

S_0 : 초기 신선도 점수 (AI 모델이 도출한 값)
$S_{\min}$: 최소 허용 신선도 점수
v : 평균 유통 속도 (km/h)
$t_{\max}$: 최대 유통 시간 (시간)
R : 유통 가능 범위 (km)

알고리즘

1. 최대 유통 시간 결정: 경험적 데이터를 기반으로 S_0와 $S_{\min}$의 관계를 설정하여 $t_{\max}$를 결정.
2. 유통 가능 범위 계산: 최대 유통 시간과 평균 유통 속도를 사용하여 도달 가능 범위 R를 계산

$$R = v \times t_{\max}$$

* 초기에는 아레니우스 방정식을 변형한 부패 모델을 이용해 최대 도달 시간을 설정하고, 나중에는 데이터를 축적해 기계학습을 통해 최적의 도달 시간을 도출, 일정한 온도에서 신선도가 시간에 따라 지수적으로 감소한다고 가정

$$k_{\text{부패속도}} = k_0 e^{-\frac{E_a}{RT}} \longrightarrow S(t) = S_0 e^{-k(T)t} \longrightarrow t_{\max} = \frac{1}{k(T)} \ln\left(\frac{S_0}{S_{\min}}\right)$$

콜드 체인 문제와 하이퍼로컬 매칭 알고리즘

$$P = \textcircled{1}((P_0 \times (1 - R_s \times R_{factor}) + (P_0 - \textcircled{1}(P_0 \times (1 - R_s \times R_{factor}))) \times \textcircled{2} F_s \times F_{factor}) \times \textcircled{3}(1 - \log(1 + k \times (1 - R_{st})))) \times \textcircled{4} S_e$$

1. Rotten Score에 따른 가격 감소:

목적: Rotten Score가 높을수록 가격을 감소시킵니다.
공식:

$$P_R = P_0 \times (1 - R_s \times R_{\text{factor}})$$

P_0: 초기 가격
R_s: Rotten Score (부패 점수)
R_{factor}: Rotten Score에 따른 가격 감소를 조정하는 계수

2. Fresh Score에 따른 가격 복원:

목적: Fresh Score가 높을수록 가격을 초기 가격에 가깝게 복원시킵니다.
공식:

$$P_F = P_R + (P_0 - P_R) \times F_s \times F_{\text{factor}}$$

P_F: Fresh Score에 따라 조정된 가격
F_s: Fresh Score (신선도 점수)
F_{factor}: Fresh Score에 따른 가격 복원을 조정하는 계수

3. 유통기한에 따른 가격 조정:

목적: 유통기한이 가까워질수록 가격을 감소하며, 유통기한이 지나면 판매를 금지합니다.
공식:

$$P_E = \begin{cases} 0, & \text{if } R_{dt} \leq 0 \\ P_F \times (1 - \log(1 + k \times (1 - R_{dt}))), & \text{if } R_{dt} > 0 \end{cases}$$

P_E: 유통기한에 따라 조정된 가격
R_{dt}: 잔여 유통기한 비율

$$R_{dt} = \frac{D_e - \text{Current Date}}{D_e - D_t}$$

D_e: 유통기한 (만료일)
D_t: 생산일
k: 유통기한에 따른 가격 감소를 조정하는 계수

4. 보관 상태에 따른 최종 조정:

목적: 보관 상태에 따라 최종 가격을 조정합니다.
공식:

$$P = P_E \times S_e$$

P: 최종 가격
S_e: 보관 상태에 따른 조정 계수

계산 예시:

예시 1: 유통기한이 많이 남은 경우
초기 가격 (P_0): 10,000원
Rotten Score (R_s): 0.2
Fresh Score (F_s): 0.9
잔여 유통기한 비율 (R_{dt}): 0.8
보관 상태 계수 (S_e): 냉장 보관 육류 (1.1)
$R_{\text{factor}} = 0.5, F_{\text{factor}} = 0.6, k = 5$

1. **Rotten Score 조정**:

$$P_R = 10,000 \times (1 - 0.2 \times 0.5) = 10,000 \times 0.9 = 9,000\text{원}$$

2. **Fresh Score 조정**:

$$P_F = 9,000 + (10,000 - 9,000) \times 0.9 \times 0.6 = 9,000 + 1,000 \times 0.54 = 9,000 + 540 = 9,540\text{원}$$

3. **유통기한 조정**:

$$P_E = 9,540 \times (1 - \log(1 + 5 \times (1 - 0.8)))$$

$$P_E = 9,540 \times (1 - \log(1 + 5 \times 0.2))$$

$$P_E = 9,540 \times (1 - \log(2))$$

$\log(2) \approx 0.693$, 따라서:

$$P_E = 9,540 \times (1 - 0.693) = 9,540 \times 0.307 = 2,929\text{원}$$

4. **보관 상태 조정**:

$$P = 2,929 \times 1.1 = 3,222\text{원}$$