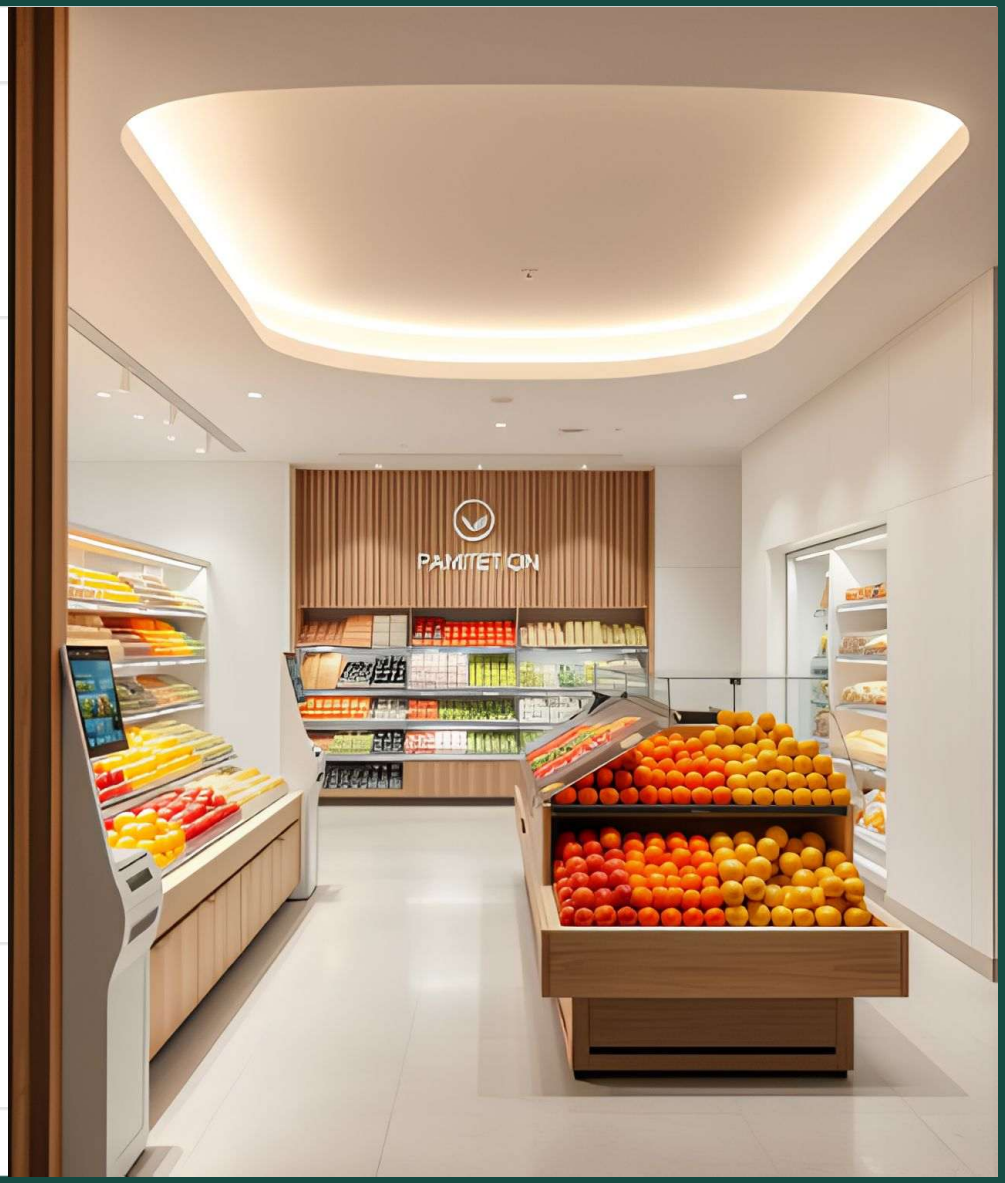
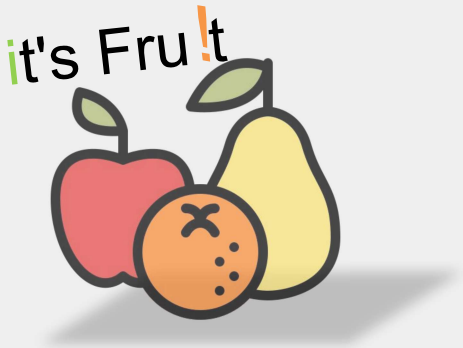
it's Frut!

Self-service fruit store



Team name	Monster
Leader	Lee Do-hyung
Member	Lee Seo-hee, Lee Sung-jae, Ha Seung-joo



Project Planning 01

1. Background
2. Goals
3. Roles
4. Operation Plan
5. Expected effect

Vision Model 02

1. Model Overview
2. Fruit detection model
3. Damage detection model
4. Strengths and Weaknesses

Price prediction model 03

1. Modeling
2. Problems

Web Service 04

1. Service Application
2. Improvements
3. Demonstration video

Results 05

1. Expected effect
2. Impressions
3. QnA

01

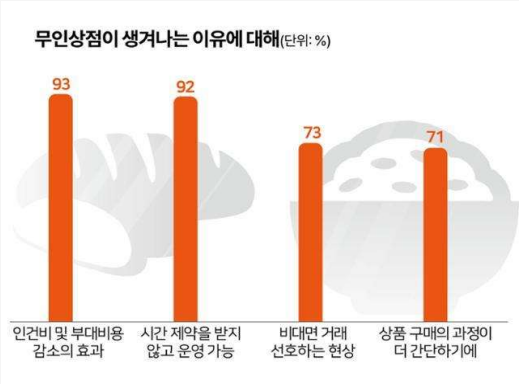
Planning

1. Background
2. Goals
3. Roles
4. operation plan
5. expected effects

Proposal - unmanned stores

01 Increase in unmanned stores

02 Reasons for increase



03 Purpose of using

Proposal - Fruits

01 Fruit prices soar

02 Purchasing Considerations

03 Dissatisfaction factor

Difficulties in purchasing for single-person households

- Amount of packaging
- Inconvenient storage of leftover fruit
- Inconvenient to handle the peel

01. Background

Business Model Benchmarks

Unmanned fruit store ‘Orot’

Orot ; Complete without any shortcomings

- **Main customer base is women in their 20s to 40s**
- **High rate of repeat visits**
- **Less problems such as theft**

- **Price reflection due to fluctuations in fruit prices**
- **Disposal of inventory of products with reduced freshness**

2024.07. **Store Tour**

Problems and solutions

Problems	• **Increased need for small purchases by single-person households due to rising fruit prices.** • **Decrease in fruit quality during distribution and in-store storage.** • **Appropriate pricing is needed to address consumer resistance and inventory issues caused by rising prices.**
Solutions	• **Allows consumers to purchase products individually.** • **Additional discounts are offered to reflect issues such as display periods and quality declines.** • **A series of processes can be performed unmanned using an object recognition model.**

01.Background

Goals

Fruit classification through object detection

Classify fruit items using photos taken by consumers

Quality Measurement and Quality-Based Discounts

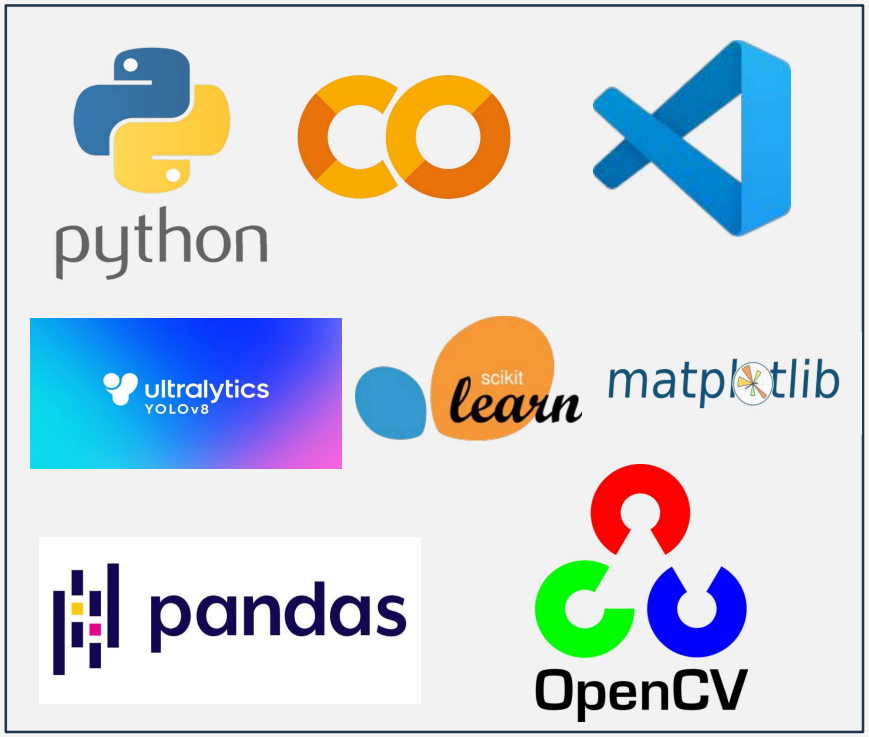
Check the damage of the fruit at the time of purchase and provide a discount based on quality.

Fruit market price prediction

Market price forecasts to help sellers determine selling and discount prices

Development Environment

Programming Languages and Libraries



Web server

BACK END

FastAPI

FRONT END

React JS

NEXT.JS

Deployment Environment

aws

Amazon
EC2

AWS RDS

Roles

Kim Eun-jong (Mentor)

- Project planning and management advice
- Model comparison, selection, and tuning advice
- Advice on how to deploy a service
- Advice on how to document your project

Lee Do-hyung (Leader)

- Data cleaning
- Price prediction model
- Output Management

Lee Seo-hee

- Damage detection model
- Web service

Lee Sung-jae

- Data collection/cleaning
- Price prediction model
- Dataset Management
- Created a PowerPoint presentation

Ha seung-joo

- Data cleaning
- Fruit detect model
- Presentation

01.Background

Schedule Management

Phase	Outputs	7.2~7.8	7.9~7.15	7.16~7.22	7.23~7.29	7.30~8.5	8.6~8.13	8.14~8.20	8.21~8.27	8.28~8.29
Understanding Artificial Intelligence	Requirements Definition Document									
Data Definition	Collection Data Definition									
Data Preparation	Module for extracting data from COCO dataset									
	Image preprocessing module									
Modeling	Product classification deep learning model									
Evaluation	Technical Verification Comparison Selection Analysis Report									
	Test Case									
Deployment	Web service									
	Story board									
Presentation	PPT									
	Source Code									
	Github									

Application Methods and Expected Benefits

Application Methods

1. Identify products selected by users and provide price information through object detection
2. Using object detection results to suggest discounts based on the current condition of the product
3. Providing services such as selecting and paying for fruits to be purchased using a camera through a mobile webpage

Expected Benefits

1. Satisfying the needs of consumers who want to buy individual fruits at a low price
2. Reduce inventory by increasing sales through quality-based discounts
3. Simplify the purchasing process by identifying products, quality, measurement, and pricing through object recognition.

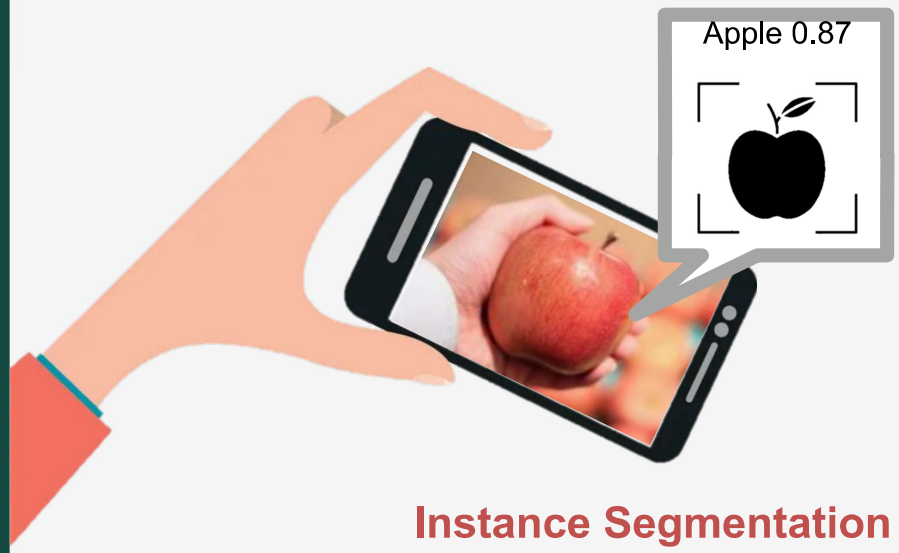
02

Vision Model

1. Model Overview
2. Fruit detection model
3. Damage detection model
4. Achievements and Limitation

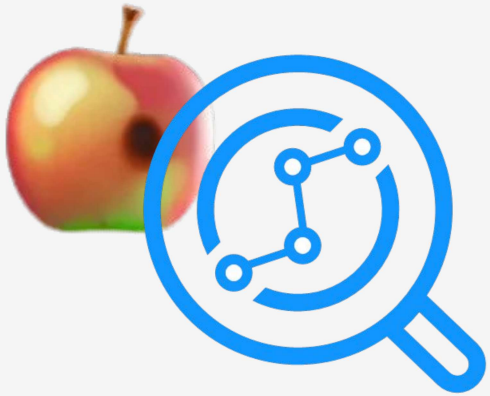
Vision Model Flow

1. Fruit Detection Model



Instance Segmentation

2. Damage Detection Model



Convolution Auto Encoder
+ K-means Clustering

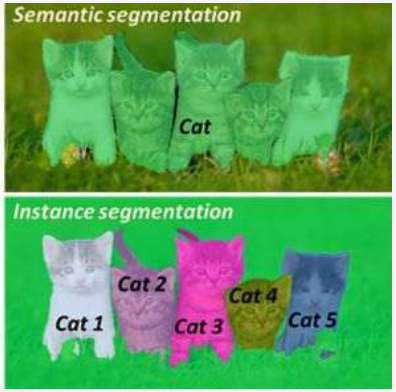
Technologies Utilized

01 Fruit Detection Model

- Instance Segmentation
- **Instance Segmentation** is a technique that detects objects in a video and segments the image at the pixel level.

  It is used to individually recognize each fruit and accurately determine the quantity.

  Unlike **Semantic Segmentation**, it distinguishes different objects within the same class.



02 YOLO Framework

- Excellent real-time inference performance
- Ease of using the framework and post-processing
- Scalability through different versions and sizes

Model comparison and selection

1. Review COCO Benchmark Results

- What is COCO Benchmark?

  Standard criteria for evaluating the performance of computer vision models using the MS COCO dataset
- We want to compare COCO benchmark results of object detection models and segmentation models to select a suitable model.

Model	mAP (0.5:0.95)	mAP@0.5	mAP@0.75	Params (M)	FPS (on GPU)	Year	Reference
Faster R-CNN	37.4	58.4	39.4	41	~6-7	2015	논문: Faster R-CNN
Mask R-CNN	38.2	60.3	40.6	44	~5	2017	논문: Mask R-CNN
SSD	25.1	43.1	25.8	34	~46	2016	논문: SSD
YOLOv5	50.4	69.7	55.2	7.1	~140	2020	GitHub: YOLOv5 GitHub
YOLOv8	52.3	70.6	56.3	11.2	~150	2023	GitHub: YOLOv8 GitHub

High performance, low complexity

Performance Evaluation Metrics

- mAP : Average Precision at various IoU thresholds
- IoU (Intersection over Union): A metric that measures the overlap between the predicted bounding box and the ground truth bounding box in object detection models.
- Params (M): Number of model parameters (in millions), meaning the model's complexity and memory requirements
- FPS (on GPU): Frames Per Second (FPS) on GPU

Model Comparison and Selection

2. YOLOv8-seg vs. Mask R-CNN

YOLOv8-seg

- A single-stage approach
- Both object detection and segmentation are performed simultaneously, offering fast processing speed and strong real-time performance.
- Applied in fields like autonomous driving and medical imaging.

Mask R-CNN

- Uses a two-stage approach
- The first stage generates Region Proposals.
- The second stage classifies objects and predicts masks within the proposed regions.
- Provides high accuracy and is advantageous for detailed object handling.

For this project, the **YOLOv8-seg** model was chosen to enable real-time inference on photos taken by customers.

YOLOv8 architecture

Overview

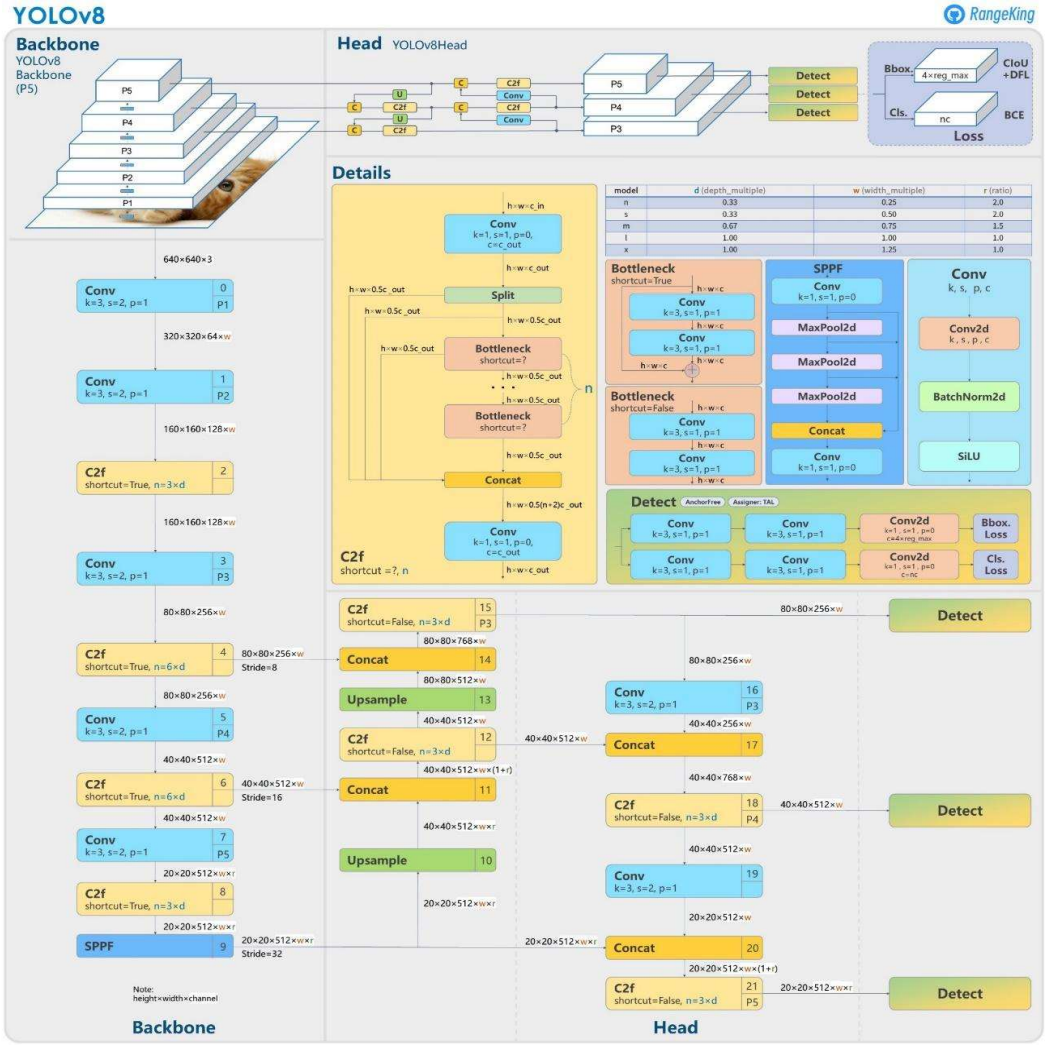
- Backbone

  The common CNN-based feature extraction structure of the YOLOv8 model combines features extracted from multiple layers in a pyramid structure.

- Head

  Designed differently for each task (Detection, Segmentation, Pose Estimation):

  - **Detection**: Head specialized for object detection.
  - **Segmentation**: Head specialized for image segmentation.
  - **Pose Estimation**: Head specialized for human pose estimation.



[Brief summary of YOLOv8 model structure · Issue #189 · ultralytics/ultralytics (github.com)]

YOLOv8 architecture

Overview

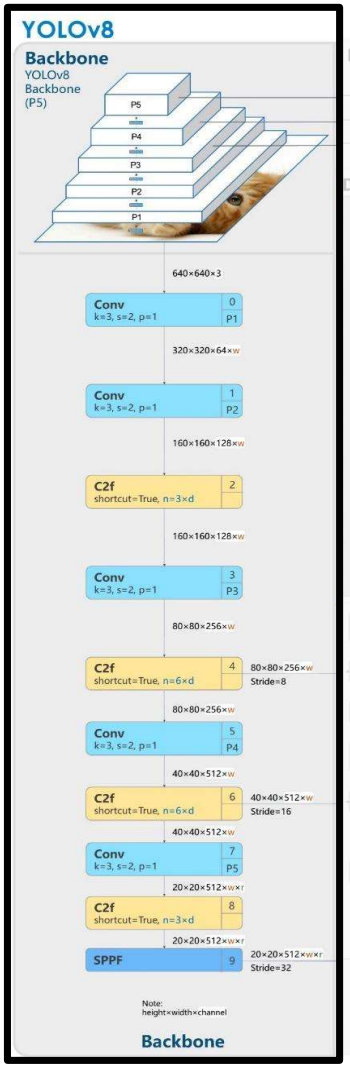
- Backbone

  The common CNN-based feature extraction structure of the YOLOv8 model combines features extracted from multiple layers in a pyramid structure.

- Head

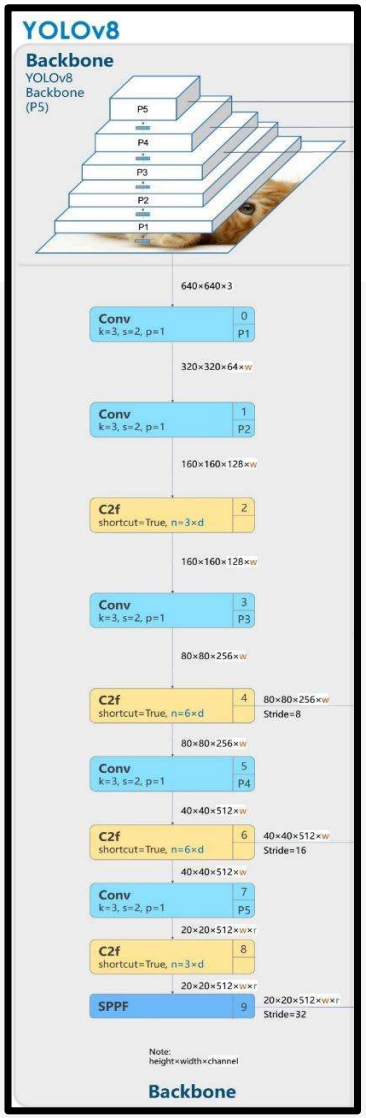
  Designed differently for each task (Detection, Segmentation, Pose Estimation):

  - **Detection**: Head specialized for object detection.
  - **Segmentation**: Head specialized for image segmentation.
  - **Pose Estimation**: Head specialized for human pose estimation.



[Brief summary of YOLOv8 model structure · Issue #189 · ultralytics/ultralytics (github.com)]

YOLOv8 architecture



Backbone

CNN-based Feature Extraction Structure

- Gradually extracts image features using various layers and blocks.
- **Pyramid Structure**: Combines FPN (Feature Pyramid Network) and PAN (Path Aggregation Network) to merge features extracted from multiple layers, collecting information from different scales and enabling recognition of objects of various sizes.

Components:

- **Conv**: Extracts low-level features of the image.
- **C2f (Cross Stage Partial with Focus)**: Combines intermediate features to propagate information efficiently.
- **SPPF (Spatial Pyramid Pooling - Fast)**: Combines features of various sizes to extract global features.

YOLOv8 architecture

Head

1. **Detect**: A head specialized for object detection

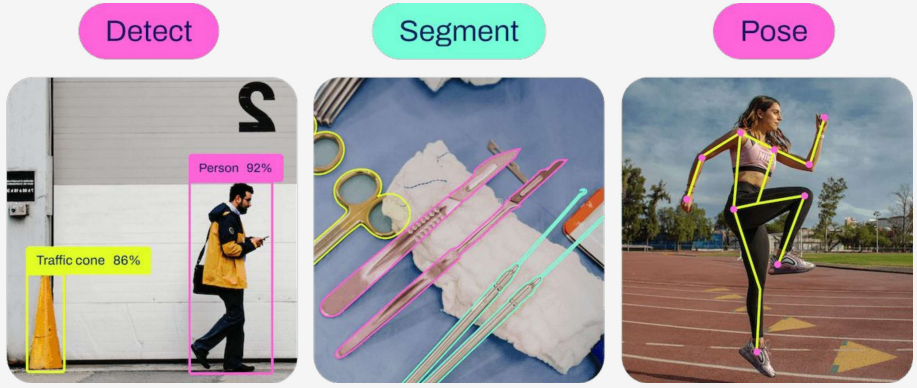
   Predicts the object location (bounding boxes) and class (type) from the image, performing fast and accurate object detection.

2. **Segmentation**: A head specialized for image segmentation

   Predicts which object each pixel belongs to, accurately segmenting the shape of the object. Instance segmentation generates a mask for each object.

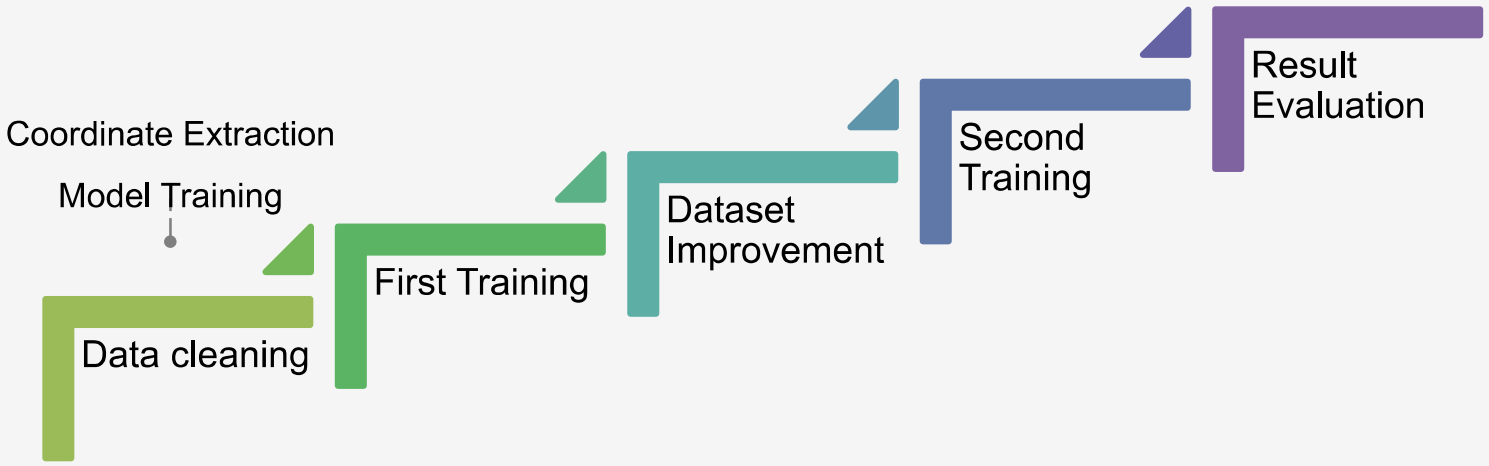
3. **Pose Estimation**: A head specialized for human pose estimation

   Predicts the location of human joints to estimate the pose.



Ultralytics YOLOv8 Tasks - Ultralytics YOLO Docs

Modeling Workflow



Dataset

01 Agricultural Product Quality QC Image Data

Providing Organizations: AIHUB, Kyungpook National University
Data Volume: 300,000 images (222GB)
Agricultural Product Types: 10 types of agricultural products with 3 or more grade classifications for each type
Data Cleaning Plan:

- Use only 4 types of fruits out of the 10 agricultural product types.
- **Class Reconstruction**
- Annotation in YOLOv8-seg label format (txt)
- Image resizing and random background composition.

02 2D, 3D Hand Movement Data from an Ego-Vision Perspective

Providing Organization: AIHUB
Data Volume: 360,000 images
Classes:

- 2D Images (own hand, other's hand)
- 3D Images (interaction objects, actions)

Data Cleaning Plan:

- Extract 100 interaction object images from AIHUB.
- Collect 266 additional images through direct capture.
- Perform image cropping and resizing.
- Manually annotate using Roboflow tool, then augment the dataset by 3x.

Dataset Improvement

Additional Hand Data Collection

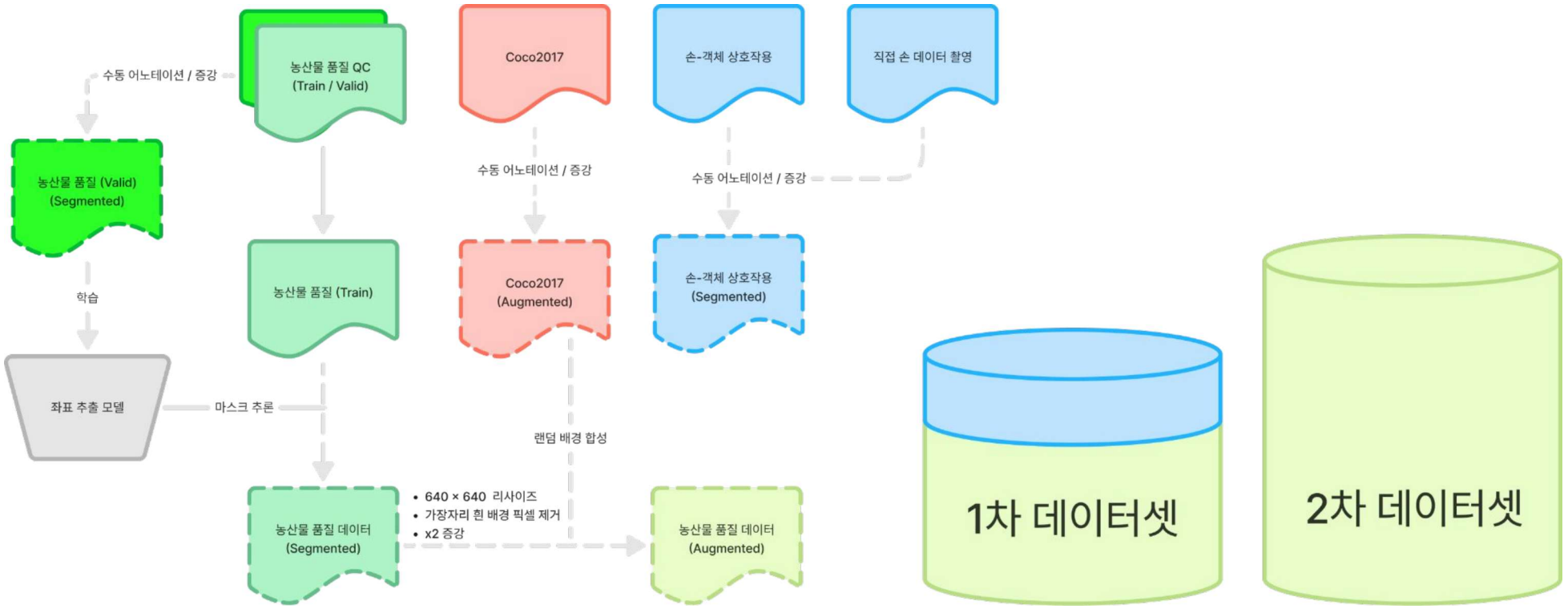
- **During the project, we noticed a lack of hand shape data for users holding fruits. To address this, we collected 266 additional images of hands holding objects like fruits, enhancing the dataset for more accurate hand movement analysis.**

Examples of interaction object images from AIHUB

Example of hand images captured directly

02. Fruits Detection Model

Data Preprocessing Flow



02. Fruits Detection Model

Raw Data

Data structure

- Using data from 4 types of fruits out of the 10 agricultural products

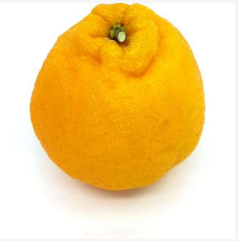
농산물 품목			이미지 산출	장수	
품목	품종				
사과	1	부사	3개 등급 x 5,000장	15000	apple
	2	양광	3개 등급 x 5,000장	15000	
배	1	신고	3개 등급 x 5,000장	15000	pear
	2	추황	3개 등급 x 5,000장	15000	
감귤	1	한라봉	3개 등급 x 5,000장	15000	Hallabong
	2	온주밀감	3개 등급 x 5,000장	15000	Onjuorange
감	1	반시	3개 등급 × 3334장	10002	Persimmon
	2	부유	3개 등급 × 3334장	10002	
	3	대봉	3개 등급 × 3334장	10002	

Class Reconstruction

Class Configuration

- Classes were organized based on common fruit classification.
- Tangerines were categorized by variety (e.g., Hallabong, Onjuorange).
- Quality grades were not considered and were integrated.

Onjuorange



Hallabong

Raw Data

Image

Format: PNG
Size: 1000x1000
Image Features:

- White background
- Various shooting angles

Annotation

Format: JSON

Annotation: Bounding box

AIHUB Agricultural Product Quality Dataset Image Examples

Raw Data

Image

Format: PNG
Size: 1000x1000
Image Features:

- White background
- Various shooting angles

Previous annotations are not usable

~~Annotation~~

~~**Format**: JSON~~

~~**Annotation**: Bounding box~~

※ New annotations are required for fine-tuning the YOLO-seg model

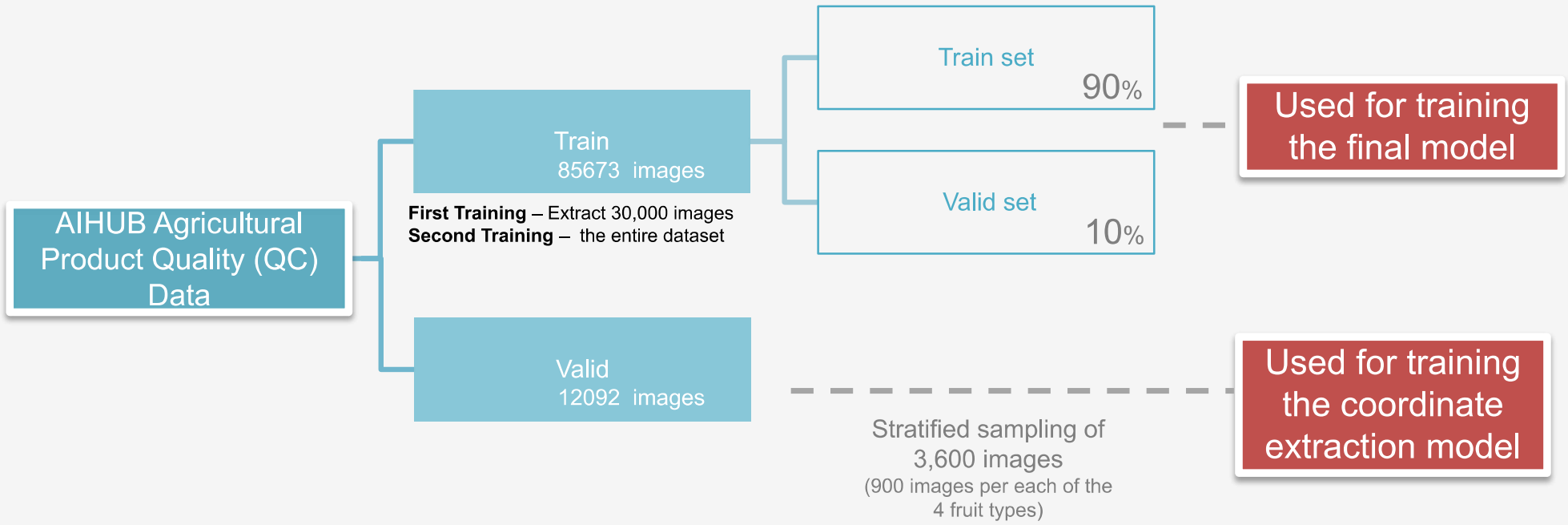
→ txt format

→ Mask coordinates

AIHUB Agricultural Product Quality Dataset Image Examples

02. Fruits Detection Model

Data Structure



Data Preprocessing

1. Annotation

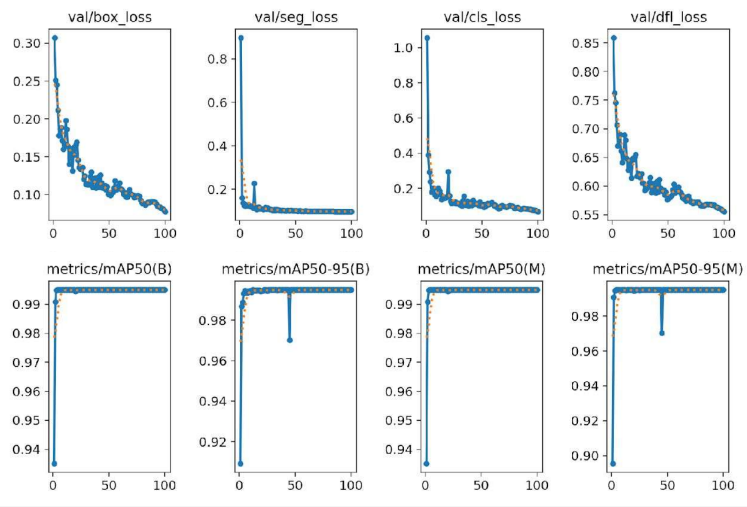
Steps	Details
 Training the coordinate extraction model	- **Base Model**: YOLOv8m-seg - **Data**: Stratified sampling of 3,600 images from the AIHUB dataset valid set - **Annotation**: Manual annotation using Roboflow, followed by 3x augmentation - **Goal**: Achieving superior mask extraction performance
 Extracting coordinates from training images	**Inference on Training Data Images using the Coordinate Extraction Model** • **Fruit Object Mask**: Reduce the mask edge by 5 pixels • **Extracted Coordinates**: Extract 80 coordinates from the reduced mask -
 "Generating annotation txt files	**Extracting Class ID from Image File Names** • Apple: 0, Hallabong: 1, Onjuorange: 2, Pear: 3, Persimmon: 4, Hand: 5 • Generate txt files in YOLO Segmentation label format

Data Preprocessing

1. Annotation

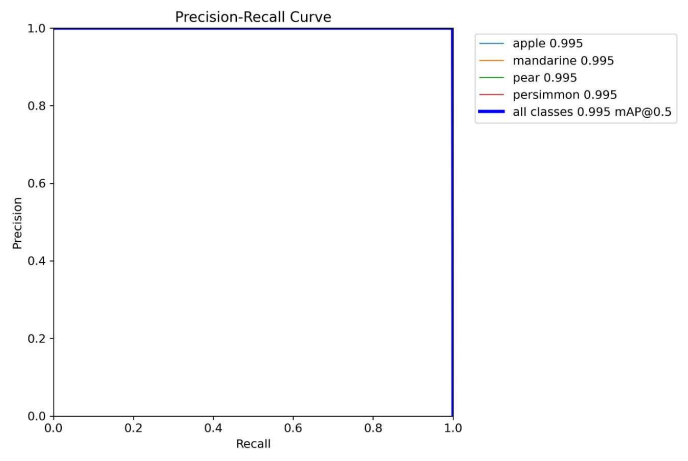
Ensure the reliability of the coordinate extraction model performance

- Validation segmentation loss is below 0.1
- Mask-related metrics are close to 1



- Mask Precision-Recall curve

Very high performance across all classes



- Mask Precision: The ratio of correctly predicted masks
- Mask Recall: The ratio of correctly predicted masks among actual masks

Data Preprocessing

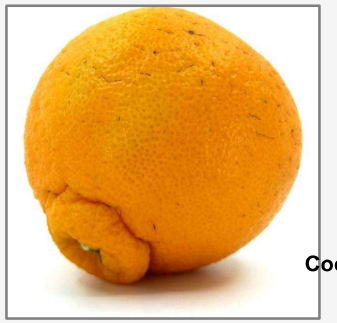
1. Annotation

Training image data

- Filename

mandarine_hallabong_S_9-71.png

- Raw image



1. Extracting class ID

- apple : 0
- hallabong : 1
- onjuorange : 2
- pear : 3
- perssimon : 4

2. Mask coordinate extraction

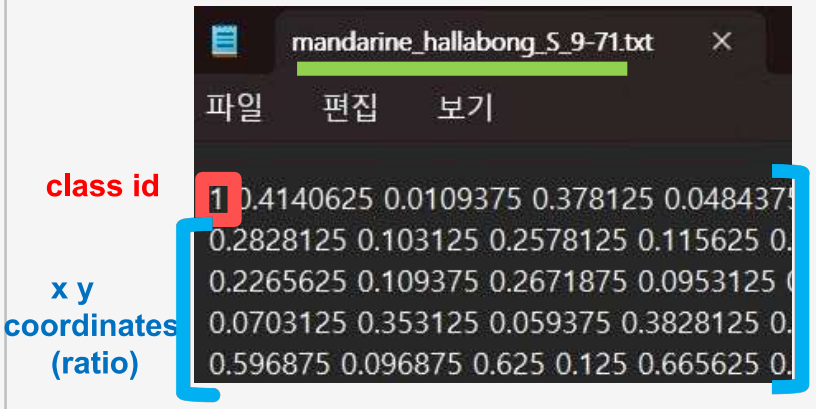
Coordinate Extraction Model

Inference Result Mask Post-processing:
Reduce the mask edge by 5 pixels

Coordinate Extraction: Extract 80 coordinates from the reduced mask

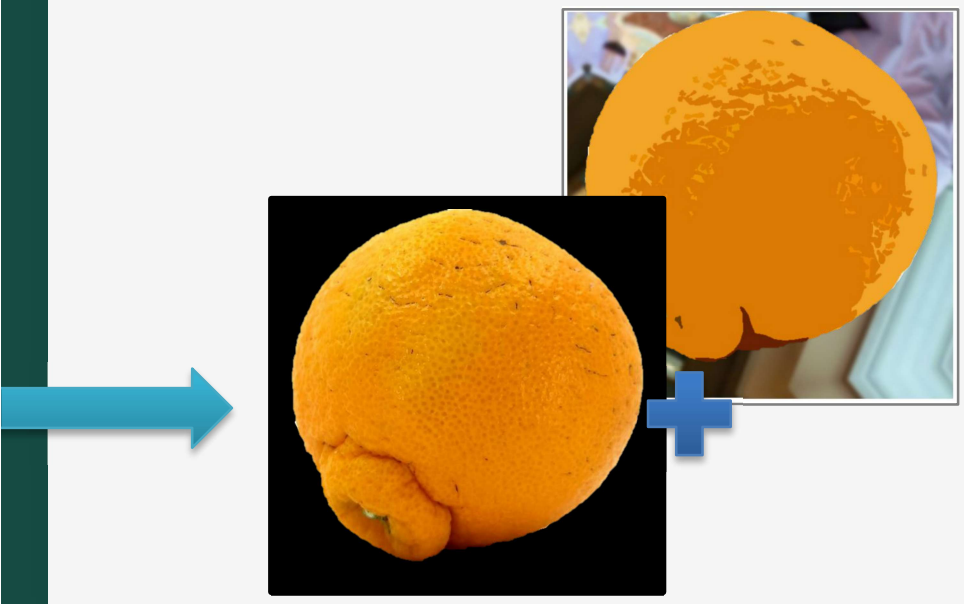
Generation png annotation file

1. File name: Same as image name
2. class id
3. Mask xy coordinates (ratio)



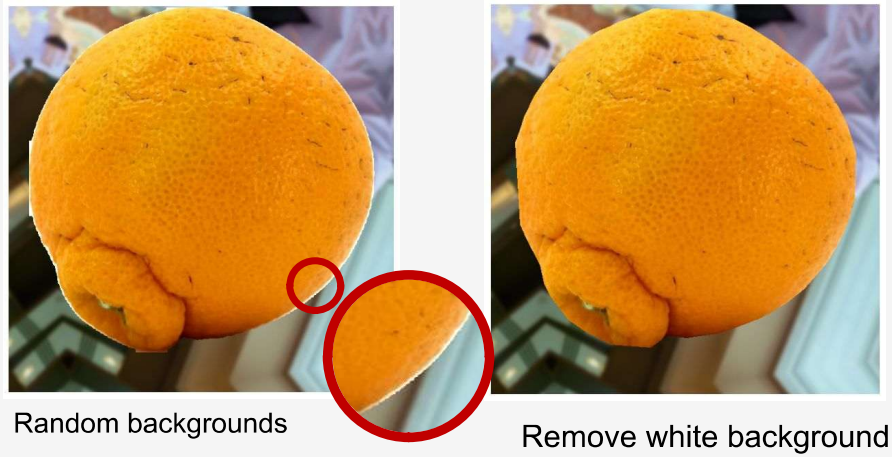
Data Preprocessing

2. Image Preprocessing



Process

1. Extract fruit masks using coordinates generated by the coordinate extraction model
2. Reflect actual store environment
   → Synthesize various random backgrounds on the fruit mask
3. Remove unnecessary background information
   → Remove white background pixels from the edges of the fruit mask



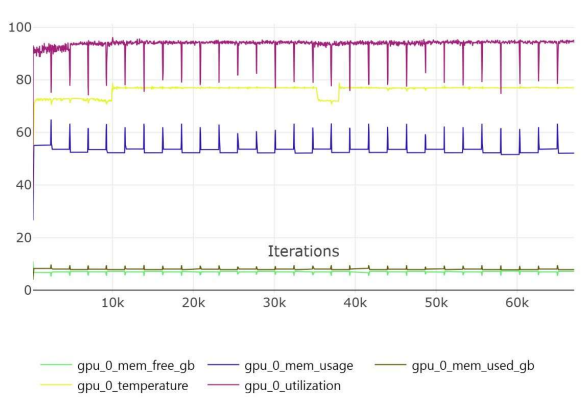
Random backgrounds

Remove white background pixels

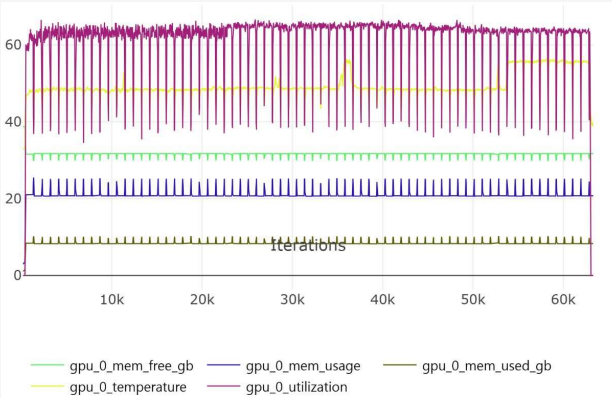
Model Training

Train Environment

	First Train	Second Train
IDE	Google Colab	
GPU 사용량	Tesla T4	NVIDIA A100-SXM4-40GB
GPU 메모리	15GB	40GB
GPU 활용도	about 94%	about 64%



First Train GPU Mornitor



Second Train GPU Mornitor

Settings

Hyperparameter

1. Explore the best hyperparameter combinations via YOLO model.tune
   - YOLOv8n-seg
   - 2 Epoch, 10 Iteration

2. **Set the Best hyperparameters** as initial parameters and final training
   - YOLOv8m-seg
   - 100 Epoch, 5 Patience

Settings

- Best Hyperparameters

Hyperparameters related to optimization and loss function		1stTrain	2nd Train
lr0	learning rate Initial	0.00987	0.01
lrf	learning rate Final	0.00997	0.01
momentum	Momentum	0.94167	0.937
weight_decay	Weight decay	0.0005	0.005
warmup_epochs	warmup_epochs	3.04295	3.0
warmup_momentum	momentum Initial	0.80059	0.8
box	Box loss weights	7.5056	7.5
cls	Class loss weights	0.50106	0.5
dfl	DFL loss weights	1.51839	1.5

Settings

Hyperparameters related to data augmentation		1st train	2nd train
hsv_h	Hue value	0.0	0.015
hsv_s	Saturation value	0.0	0.5
hsv_v	Value value	0.0	0.3
degrees	Rotation angle value	0.0	5.0
translate	Parallel translation value	0.0	0.1
scale	Image resize values	0.0	0.1
shear	Slope transformation value	0.0	2.0
perspective	Perspective transformation strength	0.0	0.001
flipud	Vertical Flip Probability	0.0	0.0
fliplr	Horizontal Flip Probability	0.0	0.5
bgr	BGR order	0.0	0.0
mosaic	Whether to use mosaic augmentation	1.0	1.0
mixup	Whether to use Mixup augmentation	0.0	0.0
Copy_paste	Whether to use copy-paste	0.0	0.0

▲ Since the image color, brightness, rotation, etc. were converted and augmented in advance through Roboflow during the first learning, the value was set to 0.

Evaluation Metrics

Box / Mask evaluation indicators

- Precision: The proportion of objects predicted that are actually correctly predicted (TP / (TP + FP)).
- Recall: The proportion of correctly predicted objects among actual objects (TP / (TP + FN)).
- mAP50: After calculating the average precision of each class with an IOU criterion of 0.5 or higher, the average value is calculated for all classes..
- mAP50-95: After calculating the average precision for each class from 0.5 to 0.95 (0.05 interval) based on IoU, the average value is calculated for all classes.
  - IoU(Intersection over Union) : A metric that measures the degree of overlap between the predicted and actual areas.
    Based on the bounding box or mask area

02. Fruits Detection Model

1st Train

Overview

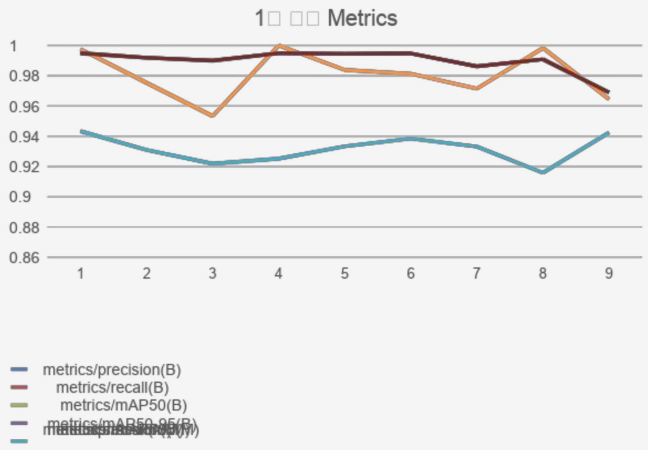
- **Goal: Fruit and hand object segmentation**
- Dataset
  - Train : 54,000 Fruits (2x increase to 27,000) + 1,098 Hands (3x increase to 366)
  - Valid : 3000 Fruits
    - Hand data is not sufficient, so it is only used in the Trainset (first learning).

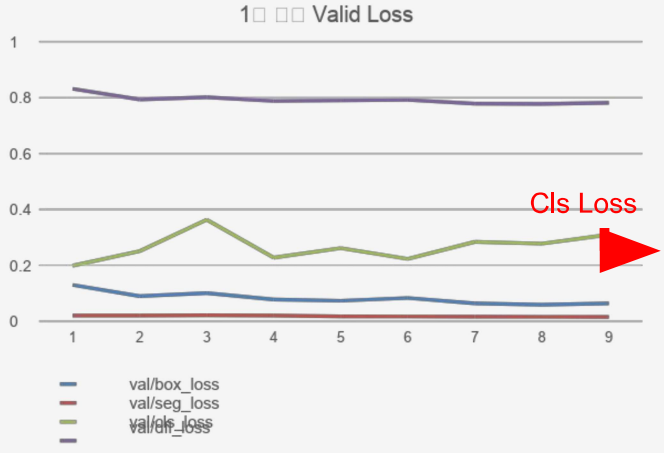
- Augmentation method
  - Apply color, brightness rotation, inversion, and translation-related augmentations using Roboflow tools
  - Physical augmentation of data volume
  - Applying mosaic augmentation techniques through parameters during learning

1st Train

1st Train Result

- High performance with metrics 0.9 or higher
- However, the problem is that the Validation Classification Loss is not improved below 0.2.
- Fruit classification performance needs to be improved.

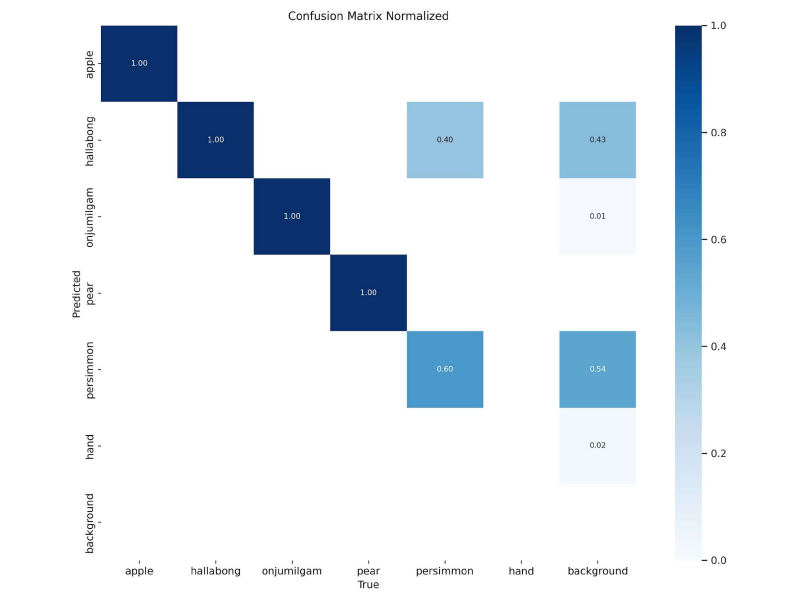




02. Fruits Detection Model

Improvements needed

- problem of 1st train results
  - Classification decrease in accuracy
    - Fruit: Double the dataset by augmenting the same data
    - Hands: Hands: Accuracy decreased due to insufficient data
  - To improve fruit classification performance, dataset enhancement and retraining will be performed.
- Dataset composition improvement
  - Excluding hand data
  - Add more fruit data and improve augmentation methods
    - **<u>Utilize approximately 80,000 images of different fruits</u>**
    - Apply augmentation techniques through training parameters"



- Misclassified Persimmon as Hallabong
- Hand data is excluded from the validation set due to insufficiency

2nd Train

Overview

Goal: Improve fruit classification and mask performance

Dataset Composition:

- **Train:** 70,482 fruit images
- **Valid:** 7,841 fruit images

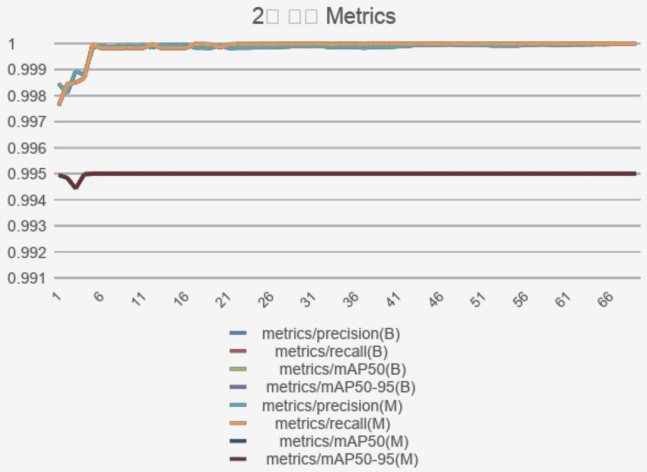
Augmentation Method:

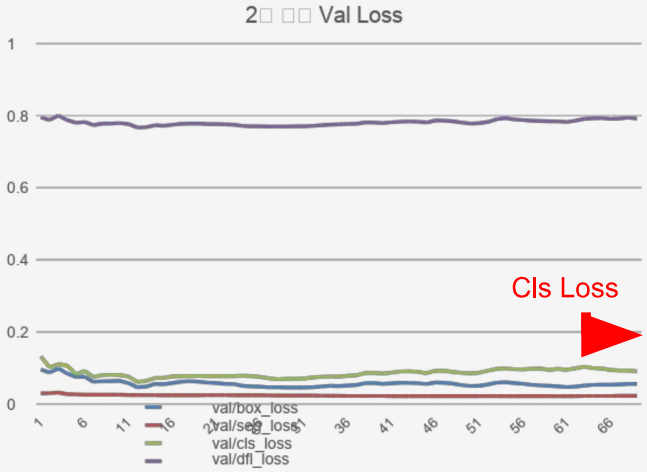
- Find the best augmentation-related hyperparameters through tuning and apply them during training
- No use of Roboflow augmentation, no physical increase in dataset size

2nd Train

2nd Train Results

- Metrics have improved to 0.995 or higher.
- Validation Classification Loss has improved to be below 0.1.





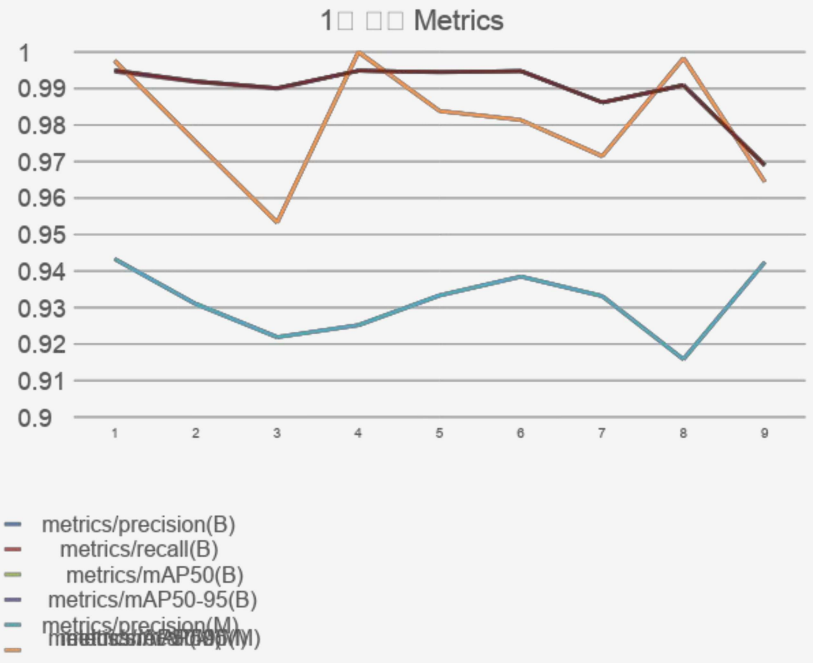
Results

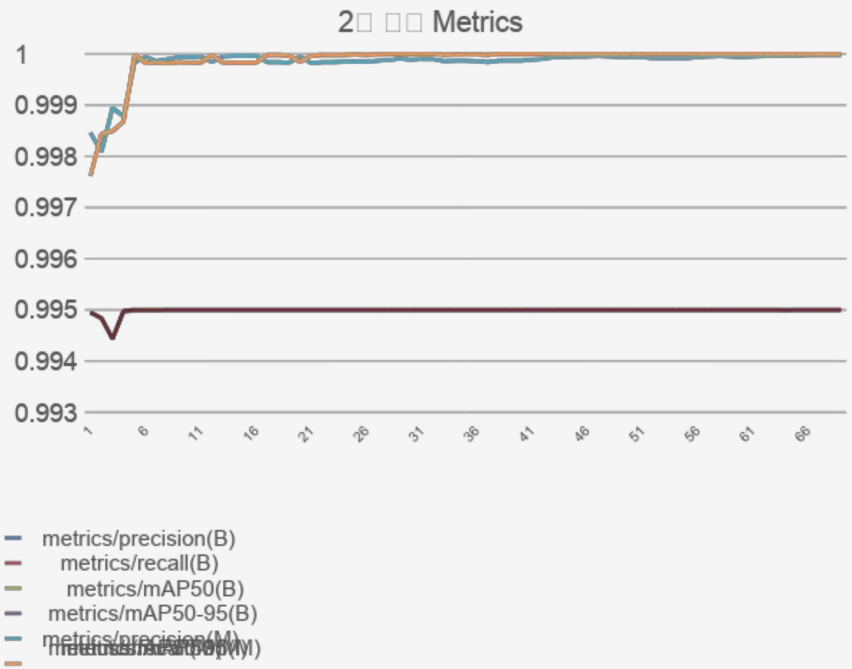
1. Comparison of Performance
2. Test Case

02. Fruits Detection Model

Comparison of performance

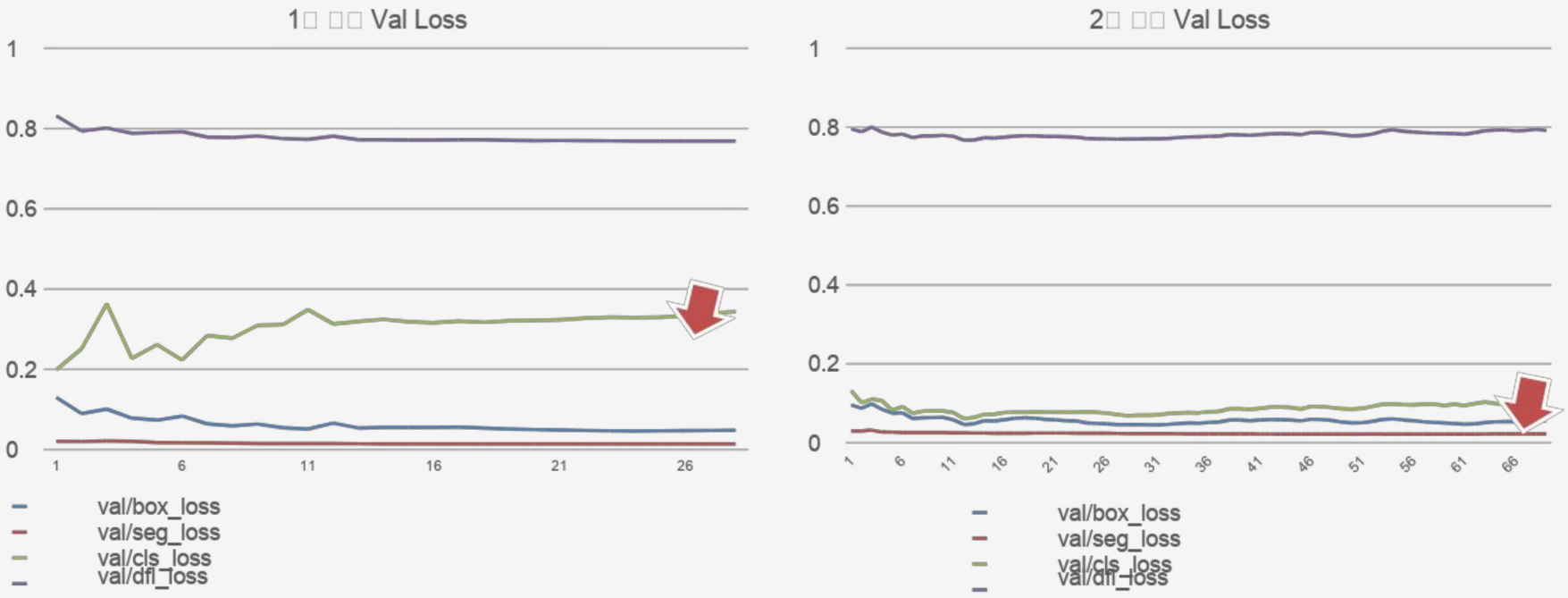
- As a result of the second training, Precision and Recall have improved close to 1
- mAP has improved to around 0.995





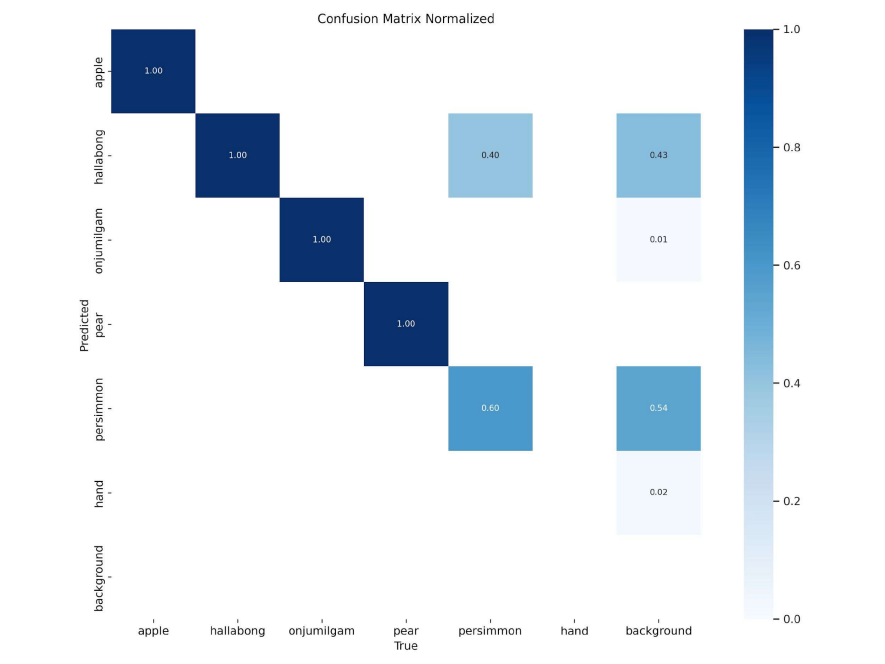
Comparison of performance

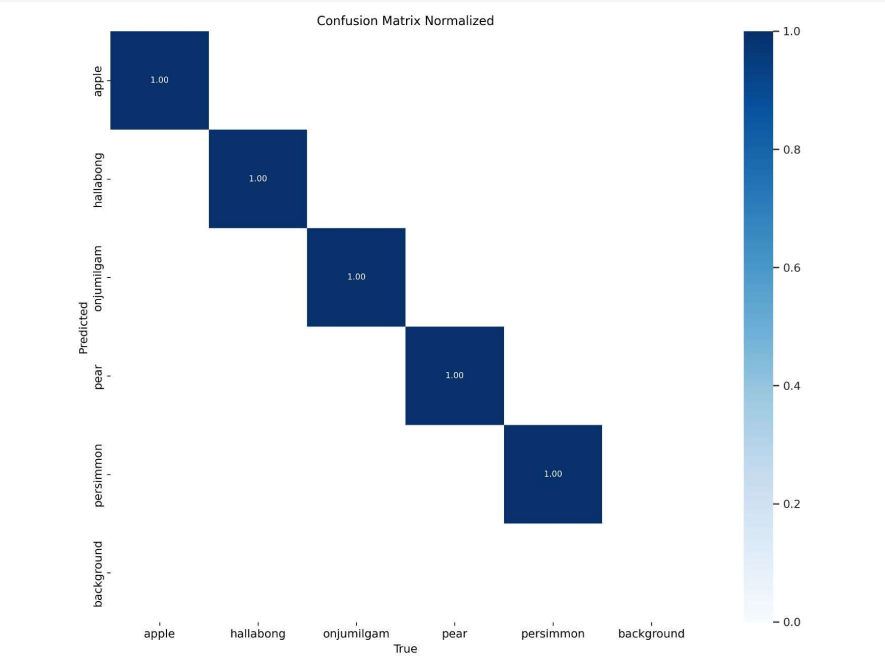
- 2nd training results, Improve the valid classification loss to be below 0.1



Comparison of performance

- 2nd training results, improved classification performance for all fruits





Test Case

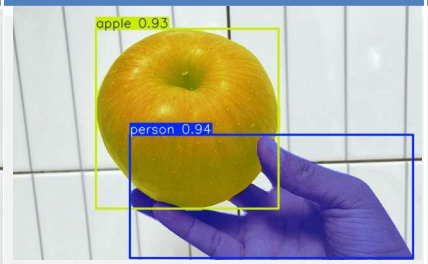
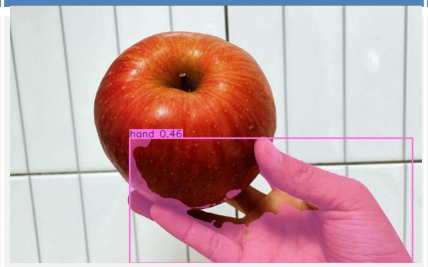
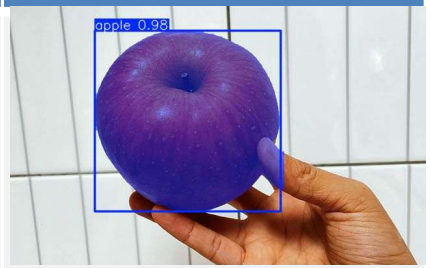
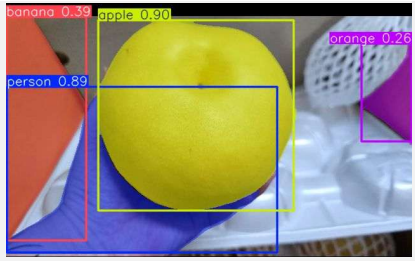
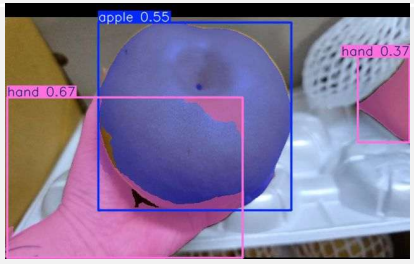
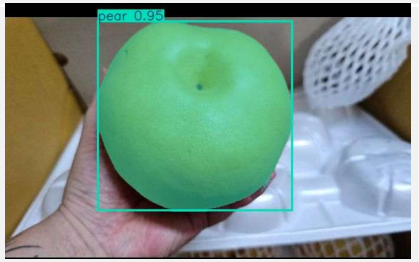
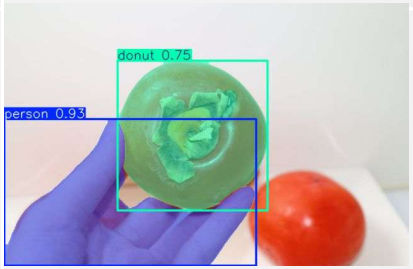
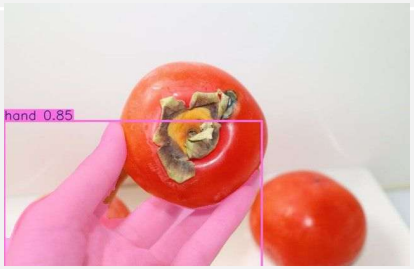
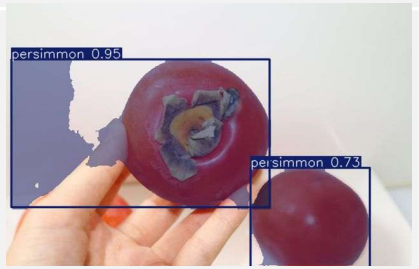
- Comparison of YOLOv8m-seg Base / 1st Train Model / 2nd Train Model

	YOLOv8-m seg	1st Train Model	2nd Train Model
Classification	• **Poor classification performance for new fruits (all fruits except apples)** • High performance in person classification	Low fruit detection and classification	High performance in fruit detection and classification
Mask	Hand, fruit mask high accuracy	Hand, fruit mask accuracy is relatively low	Fruit mask accuracy is relatively high

02. Fruits Detection Model

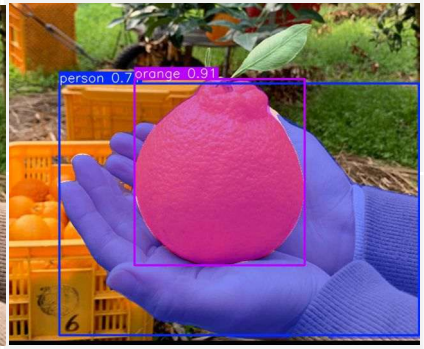
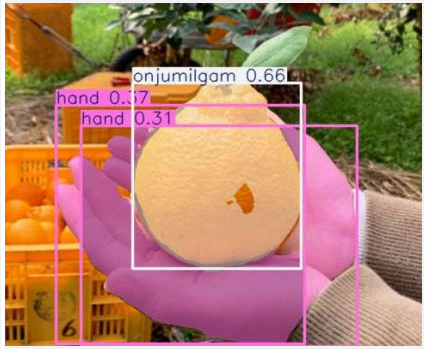
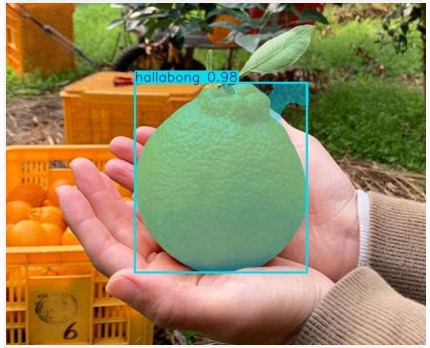
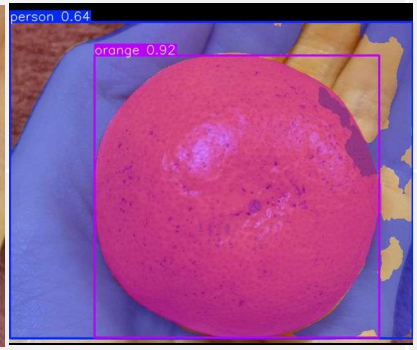
Test Case

Fruit detection and Classification
Improving Mask Performance

	Raw Images	YOLOv8-m seg	1st Train Model	2nd Train Model
apple				
pear				
persimmon				

02. Fruits Detection Model

Test Case

	Raw images	YOLOv8-m seg	1st Train Model	2nd Train Model
hallabong				
onjuorange				

02. Fruits Detection Model

Results

Comparison

1st Train

Evaluation

- Precision, Recall, mAP are high at 0.9 or higher
- Verification dataset classification loss is 0.2 or higher

Test case

- Fruits – Does not detect fruit objects or classifies them incorrectly
- Hands - Good classification performance, but low mask accuracy

2nd Train

Evaluation

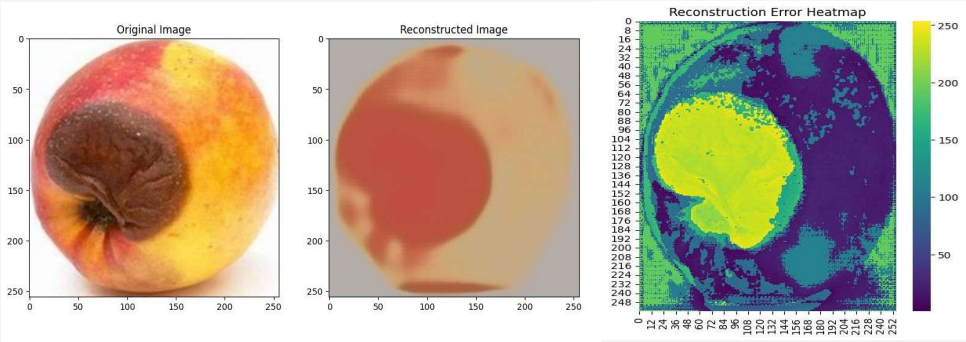
- Precision, Recall, mAP improved to 0.95 or higher
- Verification dataset classification loss reduced to 0.1 or lower

Test case

- Fruit – Improved object detecition and classification performance
- The mask accuracy is high, but the mask accuracy is low
- in some samples.

Technologies Utilized

Convolution Auto Encoder

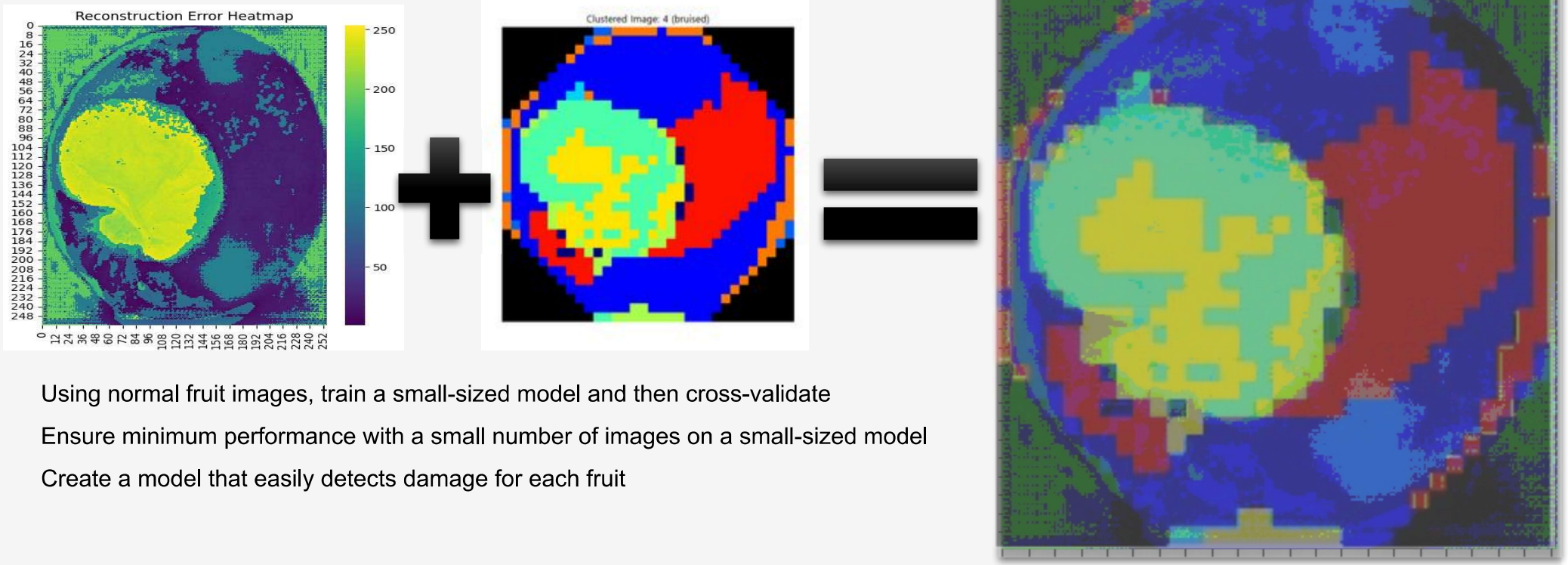


- Compression-restoration learning of normal images
- High restoration errors when restoring abnormal areas

K-means Clustering

- Split the fruit segmentation area
- Creating a segmented K-means clustering model
- TYPE distinction by small area of the image

Combined effect



- Using normal fruit images, train a small-sized model and then cross-validate
- Ensure minimum performance with a small number of images on a small-sized model
- Create a model that easily detects damage for each fruit

Achievements

1. Refined data preprocessing and cleaning
   - Fine-tuned the coordinate extraction model to generate high-accuracy annotation coordinates, improving the quality of the data.
   - Enhanced the data to better reflect real-world situations by processing image backgrounds and masks, aiming to improve the model's generalization ability.

2. Performance improvement through dataset enhancement.
   - After initial training, identified issues in the dataset and improved its composition to achieve the fruit classification goal.
   - After conducting the second training with the improved dataset, the model's performance has significantly improved.

3. Scalability and applicability
   - The model can be easily expanded by training it to adapt to various types of fruits and environments.
   - The model is applicable to various industries, including agriculture, distribution, and more.

Limitations and Improvement Plans

1. Utilizing a fruit dataset captured in a controlled environment.

- Complex preprocessing steps, such as background removal and mask processing, are required.
- The model's generalization performance may be limited in real-world scenarios.
- Improvement Plans
  - Supplementing with fruit open datasets
  - Review the need for additional experiments and retraining through data collection in real-world environments.

2. Insufficient hand data

- Although hand data was included to consider real-world usage environments, the data was insufficient, leading to the exclusion of the hand class after the first training and proceeding with the experiment.
- Improvement Plans
  - Collect hand data and object-hand interaction data from various environments.
  - Additional photos including hands and fruits were taken, considering real-world usage environments.
  - According to the testcase results, the YOLOv8-seg base model classified hands as "person," but the performance was good.

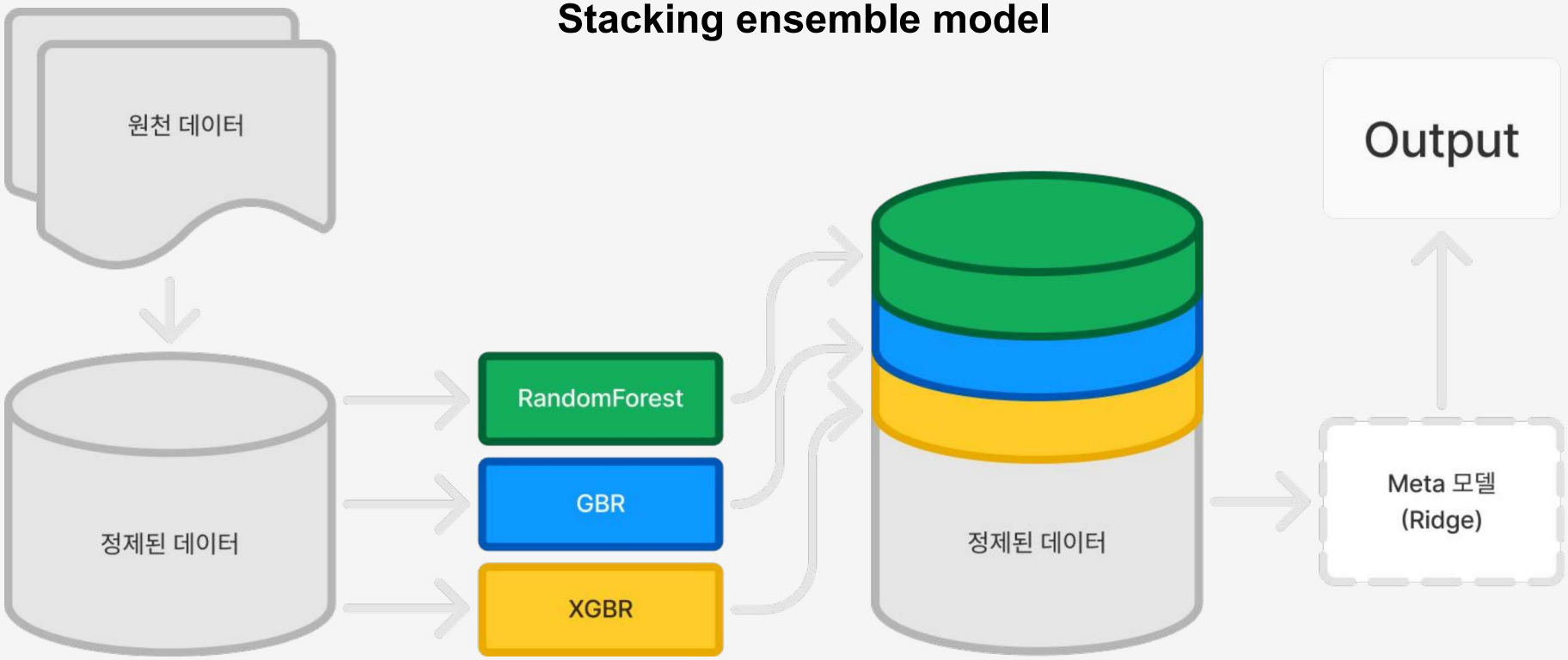
    → The "Person" class data from the COCO dataset used for the pretraining of the base model can be extracted and utilized.

03

Price Prediction Model

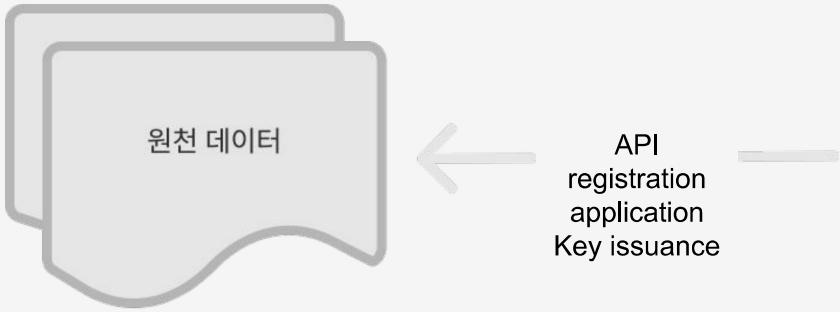
1. Modeling
2. Problem

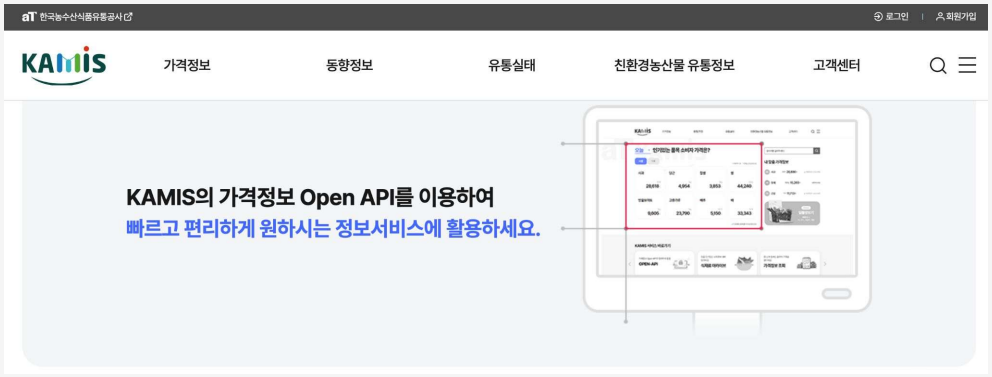
Modeling


Stacking ensemble model
원천 데이터
정제된 데이터
RandomForest
GBR
XGBR
정제된 데이터
Meta 모델
(Ridge)
Output

03. Price Prediction Model

Modeling

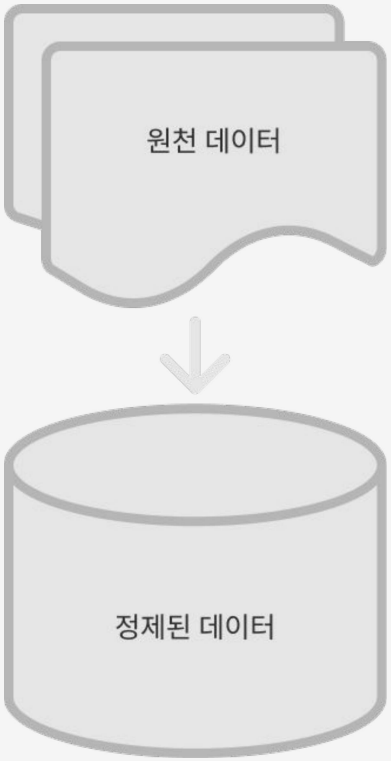




1. Korea Agro-Fisheries & Food Trade Corporation Agricultural Products Distribution Information
   a. Information on agricultural products from Garak Wholesale Market by item and grade
   b. Fuji apples 1 year retail price

2. Meteorological Administration Jongno-gu Weather Information
   a. Weather phenomenon observation data such as temperature, precipitation, and sunshine hours

Modeling

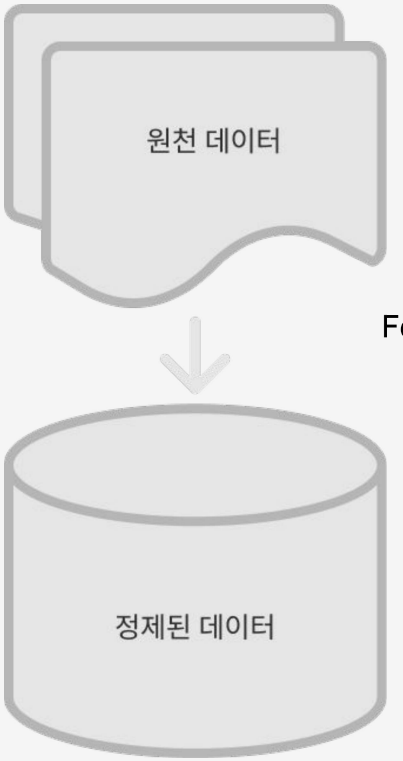


Data Cleaning

Generalizing the data cleaning process using sklearn.pipeline

```
# Pipeline
pipeline = Pipeline(steps=[
    ('convert_date', convert_date_transformer),
    ('date_features', date_features_transformer),
    ('drop_invalid_temperature', drop_invalid_temperature_transformer),
    ('bin_days', bin_day_transformer),
    ('drop_rows', drop_rows_transformer),
    ('drop_columns', drop_columns_transformer),
    ('one_hot_encode', one_hot_encode_transformer),
])
```

Modeling



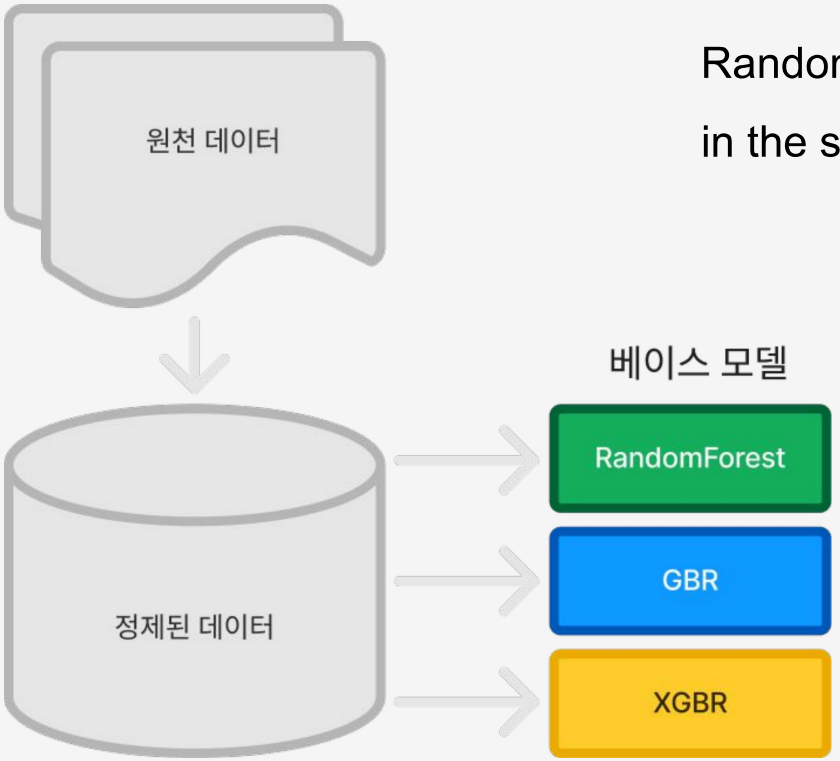
Feature Engineering

Generalizing the data cleaning process using sklearn.pipeline

```
# Pipeline
pipeline = Pipeline(steps=[
    ('convert_date', convert_date_transformer),
    ('date_features', date_features_transformer),
    ('drop_invalid_temperature', drop_invalid_temperature_transformer),
    ('bin_days', bin_day_transformer),
    ('drop_rows', drop_rows_transformer),
    ('drop_columns', drop_columns_transformer),
    ('one_hot_encode', one_hot_encode_transformer),
])
```

03. Price Prediction Model

Modeling



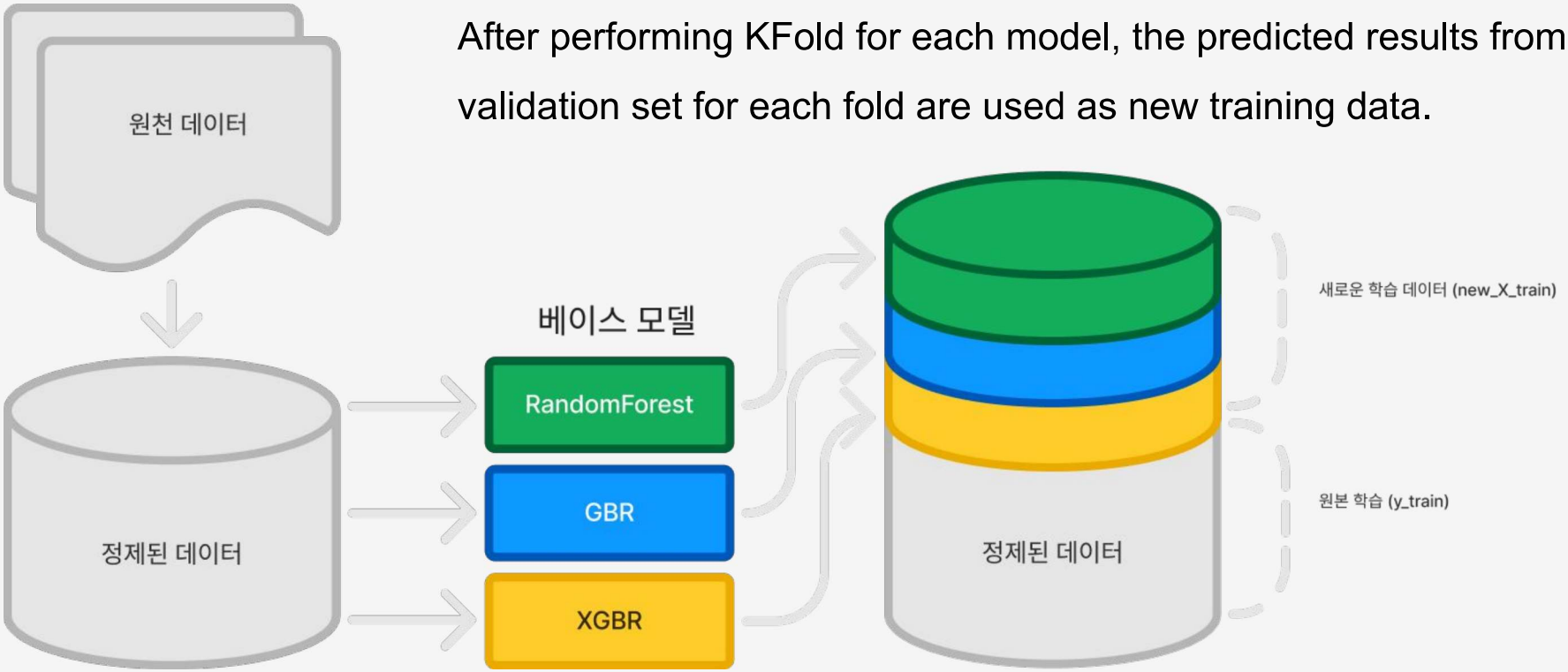
RandomizedSearch finds the optimal hyperparameters in the set search space of each model.

```
base_models = [
    ('rf', RandomForestRegressor(random_state=19),
     {"n_estimators": randint(50, 300),
      "max_depth": randint(3, 10)}),
    ('gb', GradientBoostingRegressor(random_state=19),
     {"n_estimators": randint(50, 300),
      "learning_rate": uniform(0.01, 0.1)}),
    ('xgb', XGBRegressor(random_state=19),
     {"n_estimators": randint(50, 300),
      "learning_rate": uniform(0.01, 0.1),
      "max_depth": randint(3, 10)})
     ]

optimized_base_models = []
for name, model, param_dist in base_models:
    search = RandomizedSearchCV(model, param_distributions=param_dist,
                                n_iter=50, cv=5, random_state=19, n_jobs=-1, verbose=1)
    search.fit(X_train, y_train)
    optimized_base_models.append((name, search.best_estimator_))
```

Modeling

After performing KFold for each model, the predicted results from the validation set for each fold are used as new training data.



Modeling

The result of inference in X_test

```
RMSE: 1591.035549502167
R²: 0.7315085158672294
```

Data forecast results after August 1, 2024

```
RMSE: 5104.619904969978
R²: -1.3897993399905264
```

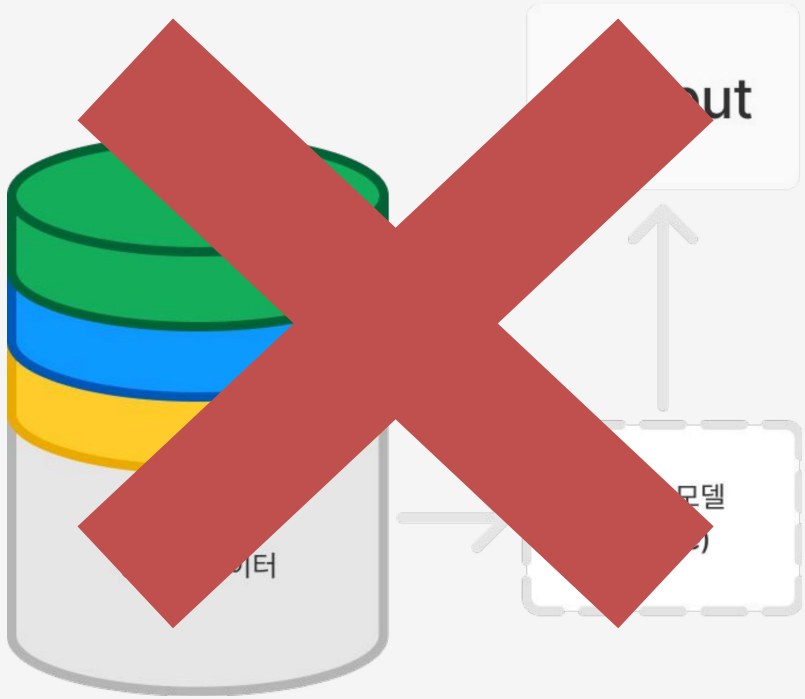
Category column one-hot encoding processing

```
RMSE: 1286.3850392665236
R²: -0.479852158968914
```

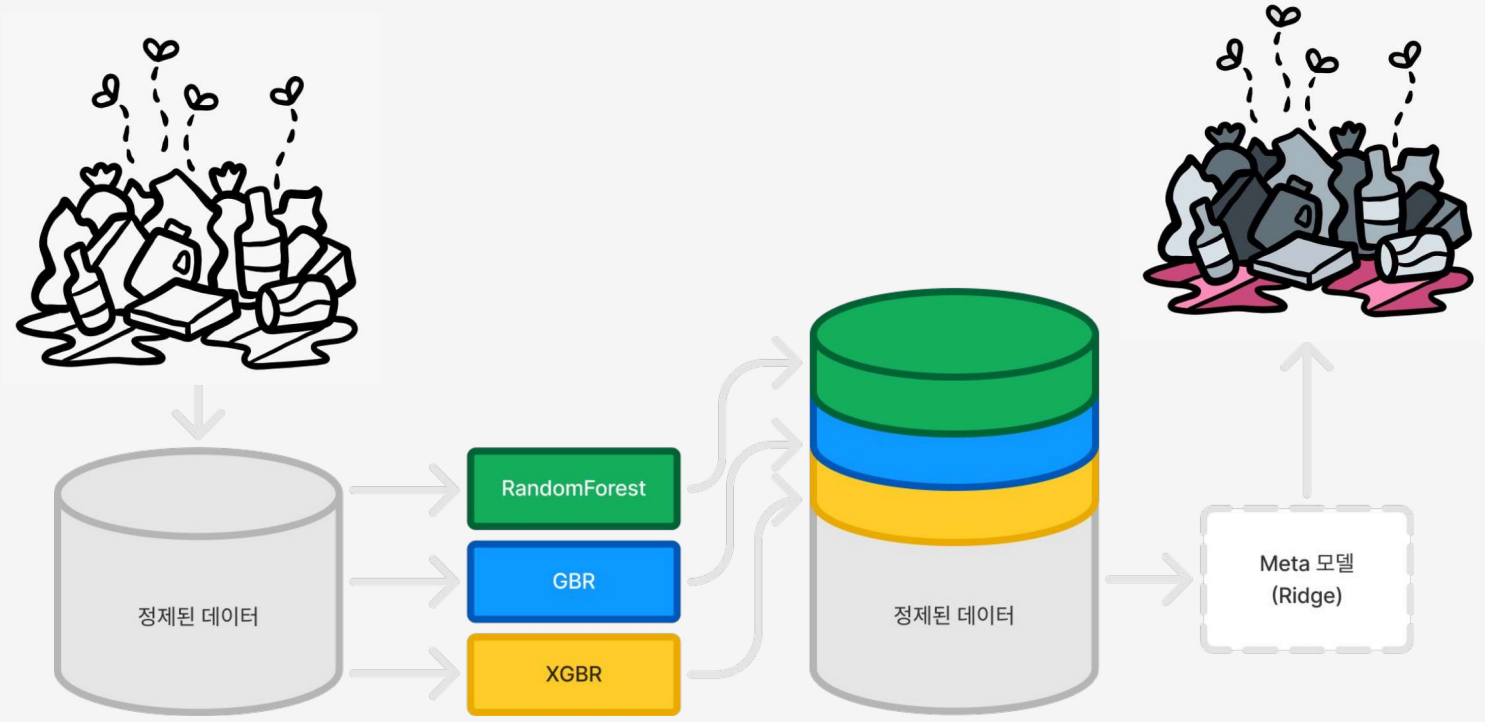
Change meta model

```
RMSE: 1383.5153745985 13
R²: 0.07199817374880757
```

Train the model using the final model (Ridge)



Problem - Dataset (Garbage in, Garbage out)



Problem - Algorithm (No Free Lunch)

No Free Lunch Theorems for Search

SFI-TR-95-02-010

David H. Wolpert (dhw@santafe.edu)
William G. Macready (wgm@santafe.edu)
The Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM, 87501

February 23, 1996

Abstract

We show that all algorithms that search for an extremum of a cost function perform exactly the same, according to any performance measure, when averaged over all possible cost functions. In particular, if algorithm A outperforms algorithm B on some cost functions, then loosely speaking there must exist exactly as many other functions where B outperforms A. Starting from this we analyze a number of the other a priori characteristics of the search problem, like its geometry and its information-theoretic aspects. This analysis allows us to derive mathematical benchmarks for assessing a particular search algorithm's performance. We also investigate minimax aspects of the search problem, the validity of using characteristics of a partial search over a cost function to predict future behavior of the search algorithm on that cost function, and time-varying cost functions. We conclude with some discussion of the justifiability of biologically-inspired search methods.

1 Introduction

Many problems can be cast as optimization over a "cost" or "fitness" function. In such a problem, we are given such a function, $f : \mathcal{X} \to \mathcal{Y}$ ($\mathcal{F}$ being the set of all such mappings). For that f we seek the set of $x^* \in \mathcal{X}$ which give rise to a particular $y^* \in \mathcal{Y}$. Most often, we seek the x^*'s which extremize f (this will often be implicitly assumed in this paper). Physical examples of such a problem include free energy minimization ($\mathcal{Y} = \Re$) over spin configurations ($\mathcal{X} = \{-1, +1\}^N$), or over bond angles ($\mathcal{X} = \{\Re \times \Re \times \Re\}^N$), etc. Examples also abound in combinatorial optimization, ranging from number partitioning to graph coloring to scheduling [4].

1

'No Free Lunch' - published in 1997

"if an algorithm performs well on a certain class of problems then it necessarily pays or that with degraded performance on the set of all remaining problems"

A specific type of optimized algorithm is not the same for all types of problems.

04

Web Service

1. Service Application
2. Improvements
3. Demonstration video

Service Application

Next.JS

- React Framework
- Benefits of design and SEO, SSR, etc. suitable for mobile environment

FastAPI

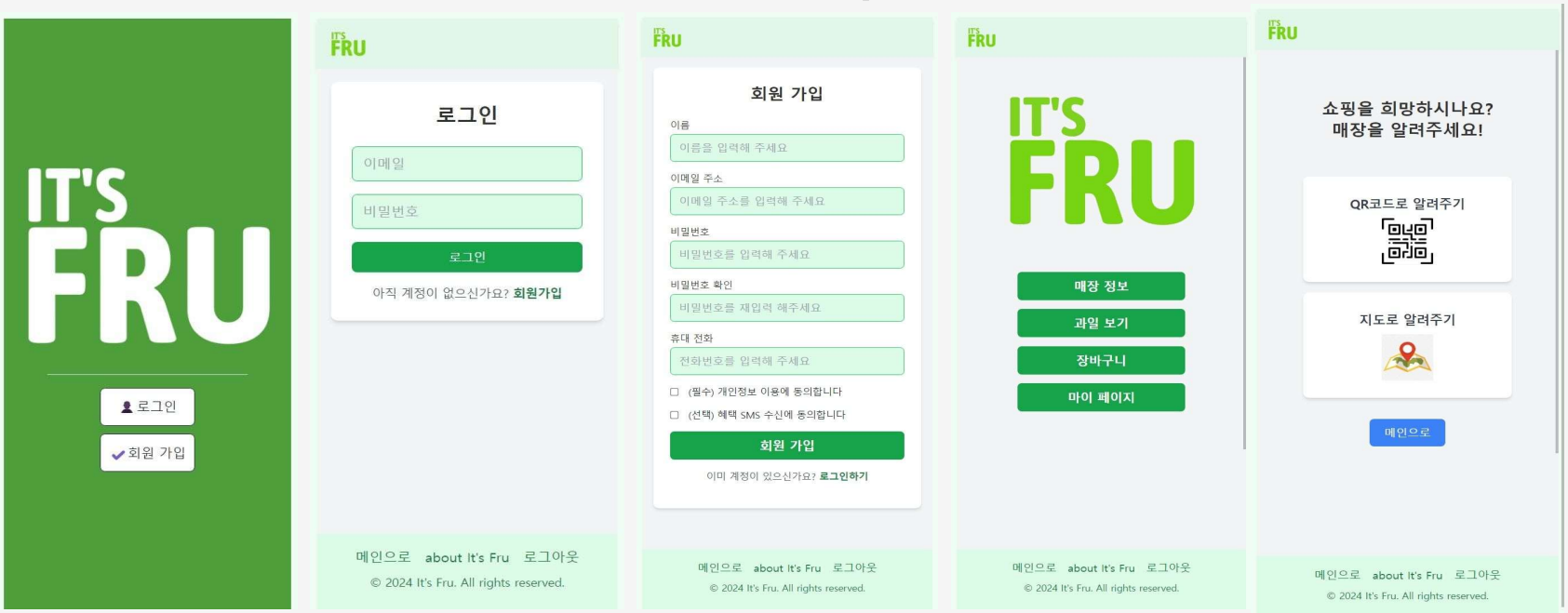
- Pytho Based on Python
- Easy integration with deep learning models

AWS

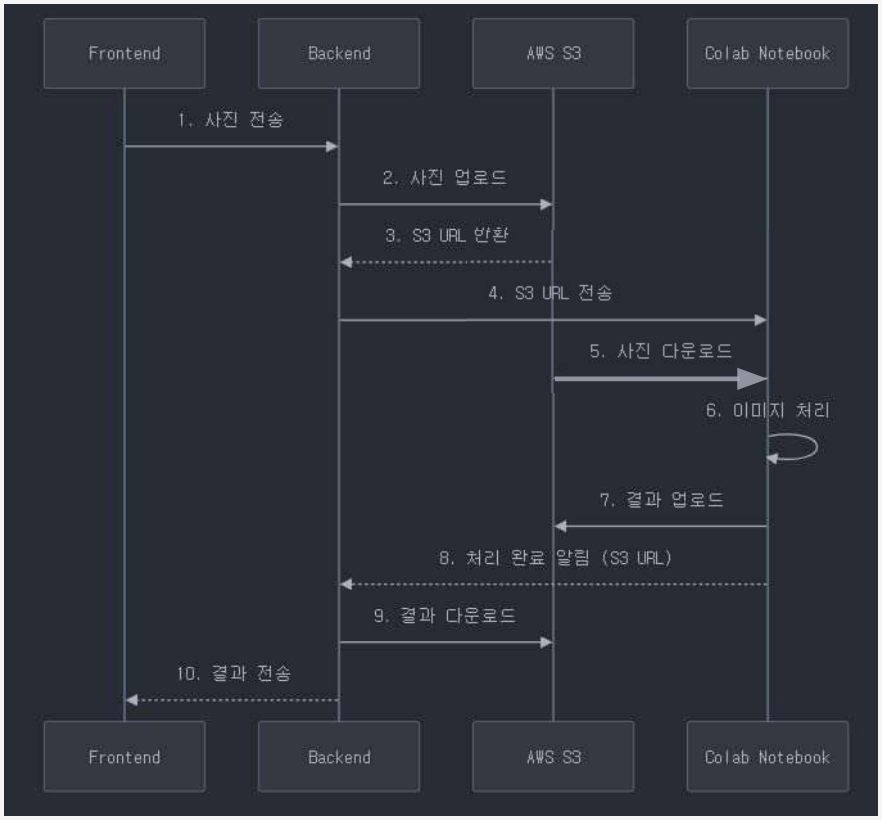
- S3, EC2, RDS
- Instances required for service deployment

Service Application

Screen example



Improvments



- Meet permissions and security settings in a real mobile environment
- Utilize authentication-related services and state management libraries
- Improved design for the mobile ecosystem
- Setting up servers and endpoints for image analysis
- Implementing QR code functionality and increasing accessibility using PWA

Demonstration video

05

Conclusion

1. Expected Benefits
2. Impressions
3. Q & A

Expected Benefits

1. **Satisfying the needs of consumers who want to buy individual fruits at a low price**
2. **Reduce inventory by increasing sales through quality-based discounts**
3. Simplify the purchasing process by identifying products, quality, measurement, and pricing through object recognition.
4. Promoting consumer inflow with flexible pricing policy
5. Collect consumer purchase data to expand discount rate optimization model