

AI Studio

# 사용자 매뉴얼

최초작성: 2021.05.28  
최종수정: 2021.08.06

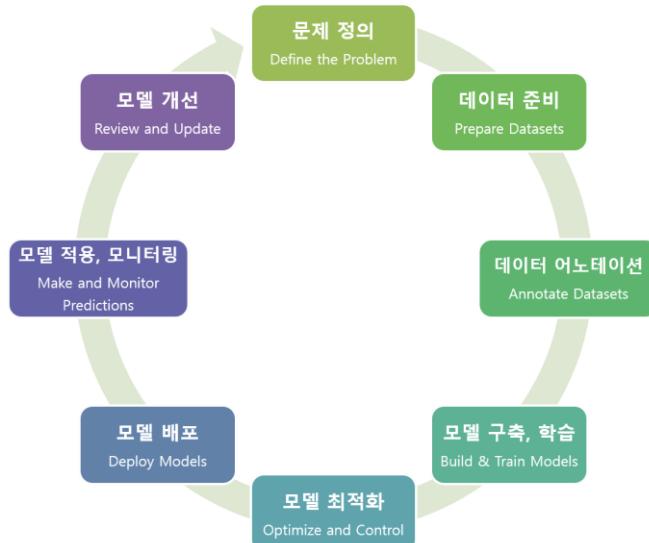
- ◆ 작성자 : Surromind
- ◆ Version : 0.82



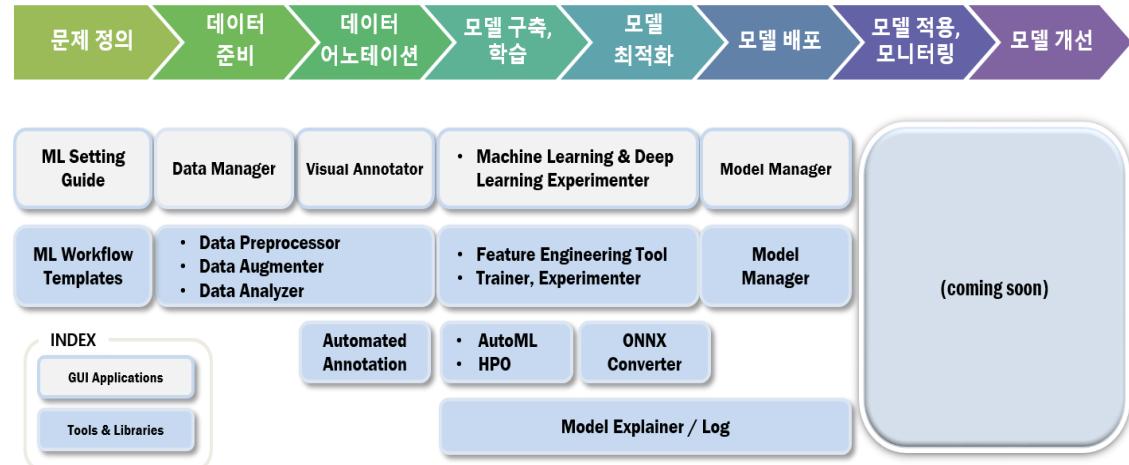
인공지능 Studio  
Surromind AI Studio

인공지능(AI) 연구개발과 응용 서비스에 이르는 전 단계에 대한 자동화 지원 종합 솔루션  
기계학습(machine learning)을 중심으로 한 데이터 기반 워크플로우 구현 지원  
“기계학습을 중심으로 한 시스템의 생명주기 비용은 최소화하고 연구개발자의 생산성을 최대화”

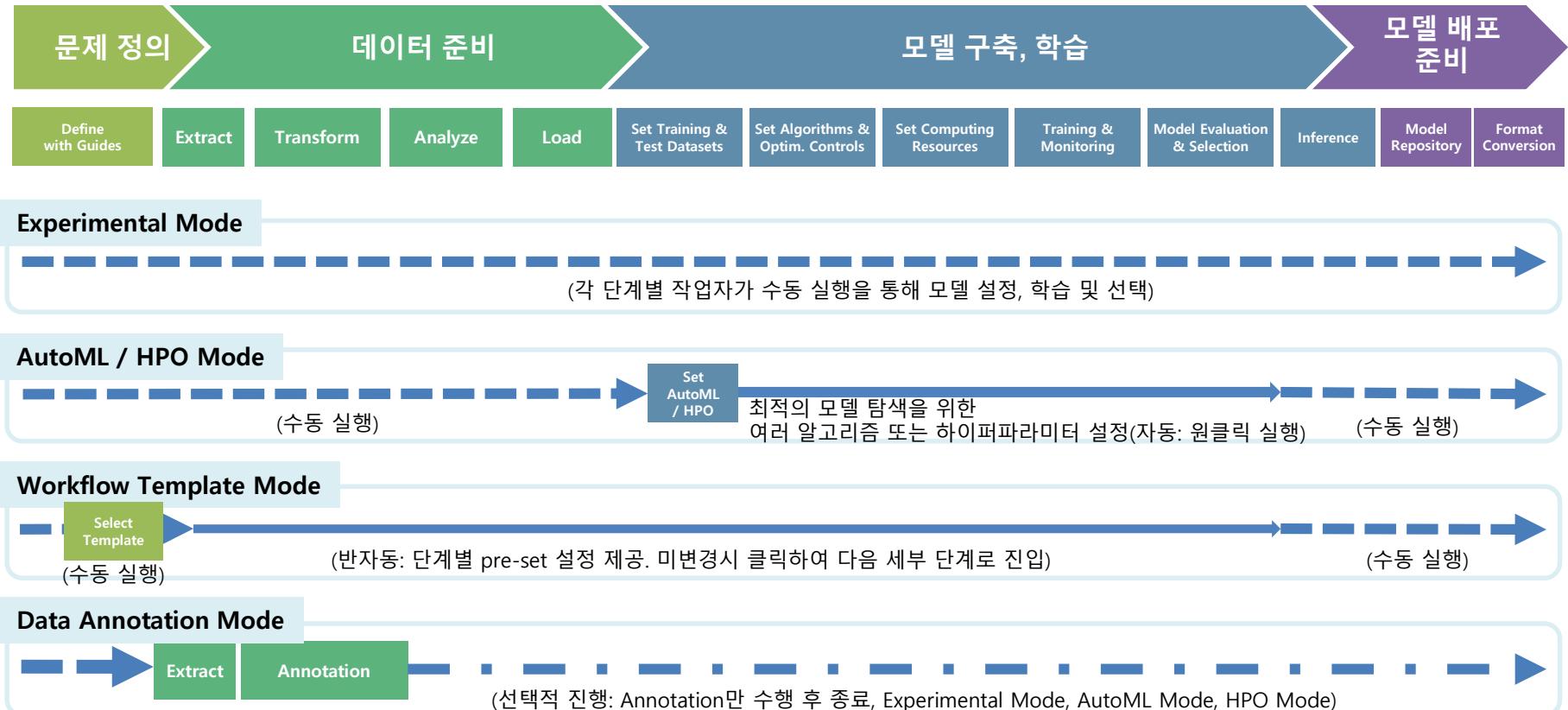
## Machine Learning Workflow



## Machine Learning Workflow의 자동화 수준 향상을 위한 AI Studio 설계



# 동작 모드 및 사용자 작업 흐름



# 머신러닝 모델링 지원 범위

입력 데이터의 종류

Tabular

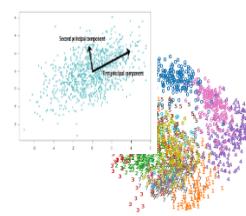


Image

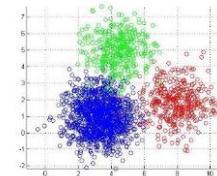


비지도학습(Unsupervised Learning)

Dimension Reduction



Clustering



지도학습(Supervised Learning)

Regression



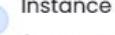
Classification



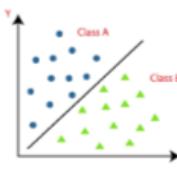
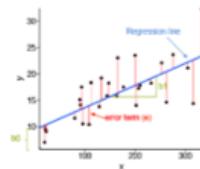
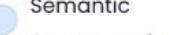
Detection



Instance Segmentation



Semantic Segmentation



# 상세 기능 목록

단계	기능	단계	기능
0. 공통	<ul style="list-style-type: none"> <li>사용자 관리(모델 개발자, 관리자, Annotator)</li> <li><b>사용자 별 권한 관리</b></li> <li>사용자 그룹 관리, 그룹별 모델 개발자 관리</li> <li>Project 관리</li> <li><b>Project 별 Templates, Dataset, Experiments 관리</b></li> <li>그룹 내 모델 개발자 간 Dataset sharing</li> <li>그룹 내 모델 개발자 간 Train 내역 조회</li> <li>Information, Alarm 관리</li> </ul>	4. 모델 구축, 학습	<ul style="list-style-type: none"> <li>문제정의에 따른 학습 Algorithm 관리</li> <li>데이터 준비 단계에서 선택한 데이터에 대해 모델을 학습하는 실험(Experiment)을 다양한 방식으로 수행               <ul style="list-style-type: none"> <li><b>Manual Train, AutoML Train 생성</b></li> <li><b>Hyper-Parameter 최적화 (HPO)</b></li> </ul> </li> <li><b>Algorithm</b>에 따른 적정 Resource 할당(Preset 기능)</li> <li>계산 클러스터에 자원 요청</li> <li>Training 상태 및 결과 조회, 모델 비교, 추론 및 배포 연계</li> <li>적정 Model에 대해 향후 재사용을 위한 Template 설정</li> </ul>
1. 문제정의	<ul style="list-style-type: none"> <li>문제정의 및 Guide 정보</li> <li><b>Project 생성</b></li> <li>딥러닝 프레임워크 설정: Tensorflow, PyTorch</li> <li>템플릿 사용 모드 설정</li> </ul>	5. 추론	<ul style="list-style-type: none"> <li>모델 검증을 위한 Dataset Upload 및 Preview</li> <li><b>학습된 모델 검증을 위한 추론 테스트</b></li> <li>진행 중인 추론 상태 및 결과 조회</li> <li>추론 결과 저장, 학습 데이터로 추가 가능</li> </ul>
2. 데이터 준비	<ul style="list-style-type: none"> <li>데이터 입력 관리</li> <li>전처리 작업을 위한 Algorithm 관리</li> <li>전처리, annotation 작업에 따른 데이터셋 버전 관리</li> <li>데이터 전처리 및 분석</li> </ul>	6. 모델 배포준비	<ul style="list-style-type: none"> <li><b>학습 완료된 모델의 배포 버전 관리</b></li> <li>모델을 ONNX 포맷으로 변환</li> <li>모델의 Tag 관리</li> </ul>
3. Annotation	<ul style="list-style-type: none"> <li><b>Image Dataset</b>에 대한 Label 작업                   <ul style="list-style-type: none"> <li>Image Classification : 수동, 자동</li> <li>Object Detection : 수동, 자동</li> <li>Instance Segmentation : 수동, 자동</li> </ul> </li> <li><b>자동 annotation</b> 기능                   <ul style="list-style-type: none"> <li>범용 모델 적용 후 label 조정</li> <li>Studio에서 학습한 모델을 적용</li> </ul> </li> <li><b>공동 annotation</b> 작업 : 개발자가 여러 명의 annotator에 작업을 분할 지정, 작업 결과 검토 및 취합하는 기능</li> </ul>	7. 계산 클러스터, 모니터링/관리	<ul style="list-style-type: none"> <li><b>GPU 중심의 계산자원 관리</b></li> <li><b>Kubernetes</b> 기반의 컨테이너 관리 시스템</li> <li>사용자별 Resource (GPU, CPU, Memory) 모니터링, 회수</li> <li><b>Repository</b> (Project, Data, Model, Docker Image) 관리</li> </ul>

## 1.6 모델링용 데이터 준비 안내 - Data Schema

1. Image dataset의 포맷은 [COCO 포맷](#)을 기반으로 약간의 변형.

- 1) Image 포맷은 jpg, png만 허용
- 2) 일부 key-value 조합만 적용

2. Tabular dataset의 포맷은 '[D3M 스키마](#)'를 참고하여 기본 요소 위주로 구성.

- 1) 저장 포맷은 csv

### COCO 포맷의 공통 데이터 구조(json)

```
{  
    "info" : info,  
    "images" : [image],  
    "annotations" : [annotation],  
    "licenses" : [license],  
}  
  
info{  
    "year" : int,  
    "version" : str,  
    "description" : str,  
    "contributor" : str,  
    "url" : str,  
    "date_created" : datetime,  
}  
  
image{  
    "id" : int,  
    "width" : int,  
    "height" : int,  
    "file_name" : str,  
    "license" : int,  
    "flickr_url" : str,  
    "coco_url" : str,  
    "date_captured" : datetime,  
}
```

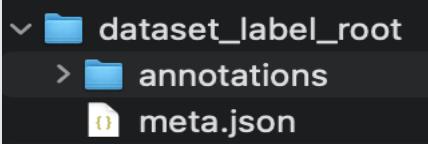
### Tabular Data Format

1. 리소스 타입: table
2. Feature (column) type: Numeric 과 Categorical로 나눔
3. Numeric
  - a. 순수 숫자형 data
  - b. 내부에서는 pandas 기준 int64 또는 float64로 표현
4. Categorical
  - a. pandas 기준 Object 또는 Bool type을 포함
  - b. 값의 요건: alphanumeric + 하이픈(-), 언더바(\_) 이외 모든 특수문자 제외
5. 이외의 모든 data type은 처리하지 않음(오류로 처리)  
예) 날짜형, time series, graph, ...

# 모델링용 데이터 준비 안내 - Data Format

Problem Type	Input	Label
Tabular Classification	- 표 형태의 파일(ex. *.csv)	- csv 파일의 마지막 열 (int형 데이터).
Tabular Regression	- 표 형태의 파일(ex. *.csv)	- csv 파일의 마지막 열 (float형 데이터).
Image Classification	- images 폴더안에 이미지의 형태로 존재하는 파일 (ex. images/0001.png, images/0002.png ... )	- annotations 폴더안에 입력 이미지와 이름이 같은 json 형태로 존재하는 파일 (ex. annotations/0001.json, annotations/0002.json ...). - json 파일의 'instances' 키의 값은 각 instance의 정보를 담은 dict의 list를 의미함. - 특정 instance dict의 'category_id' 키의 값은 해당 instance의 class를 나타냄.
Image Object Detection	- images 폴더안에 이미지의 형태로 존재하는 파일 (ex. images/0001.png, images/0002.png ... )	- annotations 폴더안에 입력 이미지와 이름이 같은 json 형태로 존재하는 파일 (ex. annotations/0001.json, annotations/0002.json ...). - json 파일의 'instances' 키의 값은 각 instance의 정보를 담은 dict의 list를 의미함. - 특정 instance dict의 'category_id' 키의 값은 해당 instance의 class를 나타냄. - 특정 instance dict의 'bbox' 키의 값은 해당 instance의 bounding box 좌표를 나타냄.
Image Semantic Segmentation	- images 폴더안에 이미지의 형태로 존재하는 파일 (ex. images/0001.png, images/0002.png ... )	- labels 폴더안에 입력 이미지와 이름이 같은 png 형태로 존재하는 mask image 파일 (ex. labels/0001.png, labels/0002.png ...).
Image Instance Segmentation	- images 폴더안에 이미지의 형태로 존재하는 파일 (ex. images/0001.png, images/0002.png ... )	- annotations 폴더안에 이미지와 이름이 같은 json 형태로 존재하는 파일 (ex. annotations/0001.json, annotations/0002.json ...). - json 파일의 'instances' 키의 값은 각 instance의 정보를 담은 dict의 list를 의미함. - 특정 instance dict의 'segmentation' 키의 값은 해당 instance의 RLE를 값으로 가지는 list를 의미함

# 모델링용 데이터 준비 안내 - Data Format

<b>Task</b>	<ul style="list-style-type: none"><li>Image Classification</li><li>Object Detection</li><li>Instance Segmentation</li></ul>	
<b>Upload Input Format</b>	 dataset_label_root.zip	 dataset_label_root.zip
<b>Label Directory Format</b>	 <pre>dataset_label_root ├── annotations └── meta.json</pre>  <pre>dataset_label_root ├── labels └── meta.json</pre> <p>Annotations and Labels directory structures:</p> <ul style="list-style-type: none"><li>Annotations directory:<ul style="list-style-type: none"><li>sample01.json</li><li>sample02.json</li></ul></li><li>Labels directory:<ul style="list-style-type: none"><li>sample01.png (or jpg)</li><li>sample02.png (or jpg)</li></ul></li></ul>	

# 모델링용 데이터 준비 안내 - SurroAnnotator

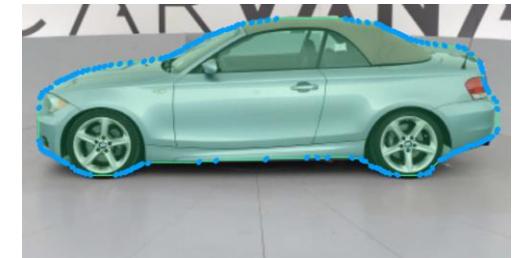
## Image annotation 기능



Image Classification



Object Detection



Instance Segmentation

## 자동 Annotation용 모델

1. 범용 선학습 모델: 문제 종류별 1건씩 설정

- ✓ Image classification: ImageNet dataset으로 선학습한 모델
- ✓ Object Detection, Instance Segmentation: COCO dataset으로 선학습한 모델

2. 사용자 학습 모델: AI Studio에서 학습한 모델 적용 가능

# 데이터 전처리 기능 목록

## ■ Tabular Data Preprocess Method

Tabular Data Preprocess Method	Library
<b>Normalize</b>	
└ MinMaxScaler	sklearn.preprocessing.MinMaxScaler
└ StandardScaler	sklearn.preprocessing.StandardScaler
<b>Discretize</b>	
└ Qcut(Qntile-based discretization)	pandas.qcut
└ cut(discretization)	pandas.cut , sklearn.preprocessing.KBinsDiscretizer
└ kBins Discritizer	sklearn.preprocessing.KBinsDiscretizer
<b>Indexing</b>	
└ Factorize	pandas.factorize
<b>ReplaceMissingvalue</b>	
└ Simple Imputer	sklearn의 SimpleImputer
└ KNN Imputer	sklearn의 KNNImputer
<b>Sort Values</b>	pandas
<b>Drop Columns</b>	pandas

## ■ Image Data Preprocess Method

Image Data Preprocess Method	Library
Flip	lagaug.augmenters.Flip
Rotate	lagaug.augmenters.Rotate
Brightness	lmaug.augmenters.Multiply
Contrast	lagaug.augmenters.GammaContrast
Shift	lagaug.augmenters.geometric.Affine
Grayscale	lagaug.augmenters.Grayscale
Crop	lagaug.augmenters.Crop
Resize	lagaug.augmenters.Resize
Equalize	lagaug.augmenters.pillike.Equalize



# 학습 및 추론 알고리즘 목록

Data Type	Problem	Algorithm 약칭	Algorithm 설명	구현 여부 및 반영 시기
Classification	Tabular	NB	Naïve Bayes classifier	
		MLP	Multi-layer Perceptron classifier	
		Logistic Regression	Logistic Regression을 사용한 classifier	
		SMO	SVM (support vector machine)의 기본형	추가 예정
		Decision Tree	의사결정나무	
Regression		XGBoost	XGBoost 라이브러리(기존 sklearn)	추가 예정
		MLP	Multi-layer Perceptron regressor	
		Support Vector Regression	Support Vector Regressor (Linear Kernel)	
		Decision Tree Regression	의사결정 나무 기반의 회귀모델	
		Linear Regression	선형회귀모델	

# 학습 및 추론 알고리즘 목록

Data Type	Problem	Framework	Algorithm 약칭	반영 여부	구현 여부 및 반영 시기
Image	Classification	PyTorch 1.* Tensorflow 2.*	ResNext	미반영	[다음은 고도화시 추가 후보] <PyTorch> 'ResNext50_32x4d', 'ResNext101_32x8d' <Tensorflow> 'ResNext50_32x4d', 'ResNext101_32x8d'
				DenseNet	Densenet121, densenet161, densenet201
			ResNet	<PyTorch> 'resnet50', 'resnet101', 'resnet152', <Tensorflow> 'resnet50', 'resnet101', 'resnet152'	[다음은 고도화시 추가 후보] <PyTorch> 'resnet34', 'resnext50_32x4d', 'resnext101_32x8d', 'wide_resnet50_2', 'wide_resnet101_2' <Tensorflow> 'resnet10', 'resnet12', 'resnet14', 'resnetbc14b', 'resnet16', 'resnet18_wd4', 'resnet18_wd2', 'resnet18_w3d4', 'resnet18', 'resnet26', 'resnetbc26b', 'resnet34', 'resnetbc38b', 'resnet50b', 'resnet101b',
			EfficientNet	'efficientnet_b0', 'efficientnet_b1',	[다음은 고도화시 추가 후보] <PyTorch> 'efficientnet_b2', 'efficientnet_b3', 'efficientnet_b4', 'efficientnet_b5', 'efficientnet_b6', 'efficientnet_b7', 'efficientnet_b8', 'efficientnet_l2' <Tensorflow> 'efficientnet_b2', 'efficientnet_b3', 'efficientnet_b4', 'efficientnet_b5', 'efficientnet_b6', 'efficientnet_b7', 'efficientnet_b8',

# 학습 및 추론 알고리즘 목록

Data Type	Problem	Framework	Algorithm 약칭	Algorithm 설명	구현 여부 및 반영 시기
Image Object Detection	PyTorch 1.* Tensorflow 2.*	ResNet101_FPN_FasterRCNN CNN	ResNet101_FPN_FasterRCNN	백본: ResNet101 Detection 기법: FPN + FasterRCNN	[다음은 고도화시 추가 후보] <PyTorch> 'DenseNet121_FPN_FasterRCNN', 'EfficientNet_b0_FPN_FasterRCNN' <Tensorflow> 'DenseNet121_FPN_FasterRCNN', 'EfficientNet_b0_FPN_FasterRCNN'
			ScaledYolov4	미반영	고도화시 추가 후보(Pytorch)
			YoloV3	미반영	고도화시 추가 후보(Pytorch, Tensorflow)
	Tensorflow 2.*	VGG16_SSD300	CenterNet	미반영	고도화시 추가 후보(Pytorch, Tensorflow)
			VGG16_SSD300	백본: VGG16 Detection 기법: Single Shot Detection 300	[다음은 고도화시 추가 후보] <PyTorch> 'ResNet50_SSD300', 'ResNet50_SSD500', 'EfficientNet_b0_SSD300', 'EfficientNet_b0_SSD500', 'DenseNet121_SSD300', 'DenseNet121_SSD500' <Tensorflow> 'ResNet50_SSD300', 'ResNet50_SSD500', 'EfficientNet_b0_SSD300', 'EfficientNet_b0_SSD500', 'DenseNet121_SSD300', 'DenseNet121_SSD500'

# 학습 및 추론 알고리즘 목록

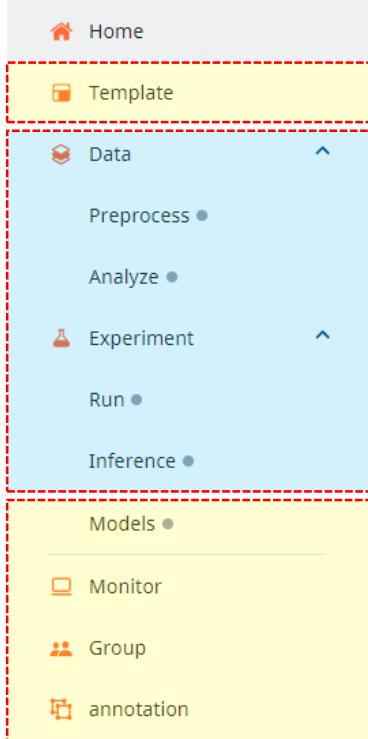
Data Type	Problem	Framework	Algorithm 약칭	Algorithm 설명	구현 여부 및 반영 시기
Image	Semantic Segmentation	PyTorch 1.* Tensorflow 2.*	WiderResNet38_DeepLabV3Plus	[PyTorch] WiderResNet38_DeepLabV3Plus ResNet50_DeepLabV3Plus SEResNet50_DeepLabV3Plus	[다음은 고도화시 추가 후보] <PyTorch> 'DenseNet121_DeepLabV3Plus', 'EfficientNet_b0_DeepLabV3Plus' <Tensorflow> 'ResNet50_DeepLabV3Plus', 'DenseNet121_DeepLabV3Plus' 'EfficientNet_b0_DeepLabV3Plus'
			HRNetV2_OCR	백본: High Resolution Network V2 헤드: Object Contextual Representations	
			UNet	미반영	[다음은 고도화시 추가 후보] <PyTorch> 'UNet' <Tensorflow> 'UNet'
			ResNet50_PSPNet	[Tensorflow] ResNet50_Pyramid Scene Parsing Network ResNet101_Pyramid Scene Parsing Network	[다음은 고도화시 추가 후보] <PyTorch> 'ResNet50_PSPNet', 'DenseNet121_PSPNet', 'EfficientNet_b0_PSPNet', <Tensorflow> 'DenseNet121_PSPNet', 'EfficientNet_b0_PSPNet',
			FastSCNN	Fast Semantic Segmentation Network	Tensorflow 반영됨
Data Type	Problem	Framework	Algorithm 약칭	Algorithm 설명	구현 여부 및 반영 시기
Image	Instance Segmentation	PyTorch 1.* Tensorflow 2.*	ResNet101_FPN_MaskRCNN	백본: ResNet101 헤드: MaskRCNN	[다음은 고도화시 추가 후보] <PyTorch> 'DenseNet121_FPN_MaskRCNN', 'EfficientNet_b0_FPN_MaskRCNN', <Tensorflow> 'DenseNet121_FPN_MaskRCNN', 'EfficientNet_b0_FPN_MaskRCNN',
			SpineNetMaskRCNN	백본: SpineNet 헤드: MaskRCNN	반영됨

# 목차

1. [Dataset 준비하기](#)
2. [학습하기](#)
3. [평가하기](#)
4. [재사용하기](#)
5. [모델 저장하기](#)
6. [Annotation](#)

# 개요

Surromind AI Studio는 프로그래밍 필요 없이 ‘비 전문가’도 쉽게 모델링 할 수 있는 AI Supporter입니다.



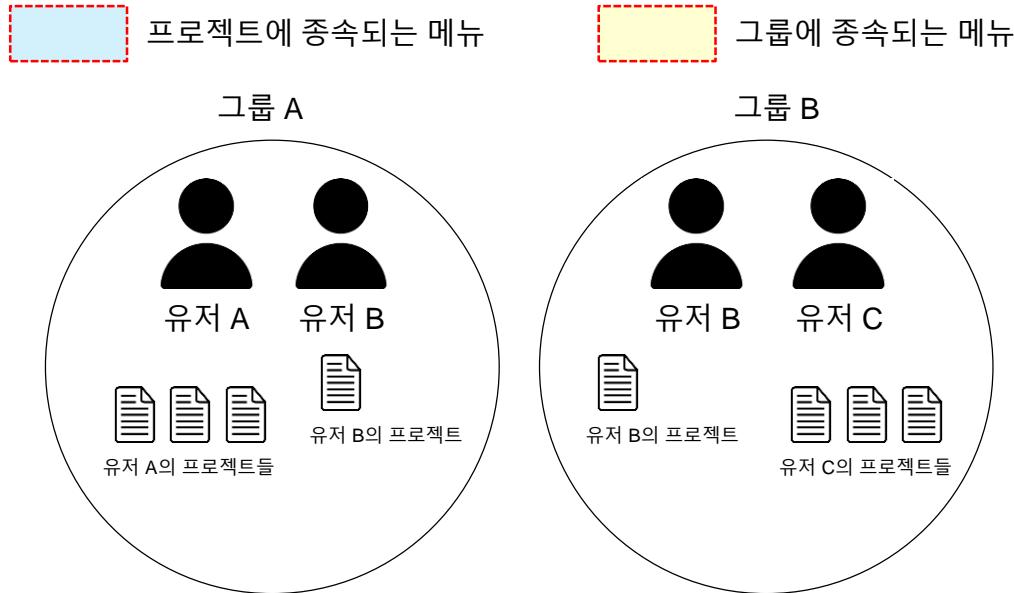
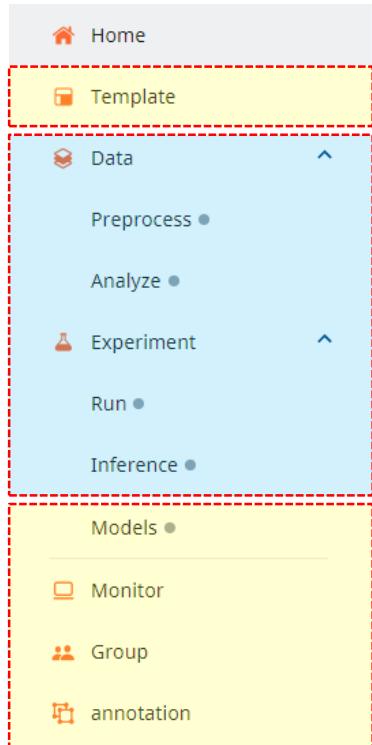
Data 메뉴에서 선택한 data를 Experiment 메뉴에서 마우스 클릭만으로 학습 가능

+

필요에 따라 유용하게 사용할 수 있는 여러 기능들 제공  
자세한 설명은 1.2 동작 모드 및 사용자 작업 흐름 참고

# 개요

각 메뉴들은 프로젝트 또는 그룹에 종속되며 사용자는 여러 그룹에 속할 수 있고 여러 프로젝트를 만들 수 있습니다.



- 각 유저들은 본인이 속한 그룹에 종속된 Template, Models, Monitor, Group, Annotation을 조회 가능
- 유저 A : 그룹 A에 속한 유저 A, B의 프로젝트만 조회 가능
- 유저 B : 그룹 A, B에 속한 유저 A, B, C의 프로젝트들 모두 조회 가능
- 유저 C : 그룹 B에 속한 유저 B, C의 프로젝트만 조회 가능

# 1. Dataset 준비하기

1.1 프로젝트 생성

1.2 Dataset 등록

1.3 데이터 전처리

1.4 Export to Experiment

## 1.1 프로젝트 생성

개발자 계정으로 로그인 하여 Home 메뉴(1)에서 Create project 버튼(2)을 클릭합니다.

The screenshot shows the SURROMIND Project management interface. On the left, there is a sidebar with a user profile for 'dev-user1' and sections for Main, Home (highlighted in orange), Template, Data, Preprocess, Analyze, Experiment, Run, Inference, Models, Monitor, Group, and annotation. The main area is titled 'Project' and displays 'Statistics' with four cards: 'Total Project' (51), 'Ready' (3), 'Working' (46), and 'Completed' (2). Below the statistics is a search bar with 'Project Name(current)' and a dropdown, a search icon, and a checkbox for '내가 생성한 프로젝트만 표시'. There are also tabs for Days, Weeks, Months, and a date input field. A large table lists projects with columns for PROJECT NAME, GROUP NAME, USER ID, DATA TYPE, MODEL, STATUS, CREATED, and UPDATED. The table includes rows for 'djlee\_od\_pytorch', 'tabular\_proj', 'tabular\_test123', 'test\_hylim\_0528', 'Test-ID\_lee0527', 'test', and 'hylim\_ss\_pytorch\_0514'. At the bottom right of the table is a red box highlighting the 'Create project +' button.

PROJECT NAME	GROUP NAME	USER ID	DATA TYPE	MODEL	STATUS	CREATED	UPDATED
djlee_od_pytorch	basic_group	dev-user1	image	Detection	Working	2021-06-02	
tabular_proj	basic_group	dev-user1	tabular	Classification	Ready	2021-06-02	
tabular_test123	basic_group	dev-user1	tabular	Classification	Working	2021-06-01	
test_hylim_0528	basic_group	dev-user1	image	Classification	Working	2021-05-28	
Test-ID_lee0527	basic_group	dev-user1	image	Classification	Completed	2021-05-27	2021-06-02
test	basic_group	dev-user1	image	Detection	Completed	2021-05-27	2021-05-27
hylim_ss_pytorch_0514	basic_group	dev-user1	image	Semantic Segmentation	Working	2021-05-14	

## 1.1 프로젝트 생성

문제 정의(1)에 따라 Data 전처리 및 Algorithm 종류, Annotation 형태가 달라지게 됩니다. 목적에 맞게 선택 후 기본 정보(2)들을 입력하여 OK버튼을 클릭합니다.

SURROMIND

Project name: 프로젝트를 로드해주세요. Template Mode

New Project

dev-user1

Main

Home

Template

Data

Preprocess

Analyze

Experiment

Run

Inference

Models

Monitor

Group

annotation

\* Definition

Data Type

Tabular  Image

Model

Regression  Classification

(1)

(2)

Cancel OK

\* Name

내용을 입력해주세요.

\* Group

basic\_group

Framework

Others

Description

내용을 입력해주세요.

## 1.2 Dataset 등록

작업할 프로젝트를 선택 후 로드합니다.

The screenshot shows the SURROMIND Project management interface. On the left, there is a sidebar with a user profile for 'dev-user1' and a navigation menu with options like Home, Template, Data, Preprocess, Analyze, Experiment, Run, Inference, Models, Monitor, Group, and annotation. The main area is titled 'Project' and displays 'Statistics' with four cards: 'Total Project' (53), 'Ready' (5), 'Working' (46), and 'Completed' (2). Below this is a search bar and a table listing datasets. The table has columns: PROJECT NAME, GROUP NAME, USER ID, DATA TYPE, MODEL, STATUS, CREATED, and UPDATED. The datasets listed are:

PROJECT NAME	GROUP NAME	USER ID	DATA TYPE	MODEL	STATUS	CREATED	UPDATED
Test-IC	basic_group	dev-user1	image	Classification	Ready	2021-06-03	
shchoitest	basic_group	dev-user1	image	Detection	Ready	2021-06-03	
djlee_od_pytorch	basic_group	dev-user1	image	Detection	Working	2021-06-02	
tabular_proj	basic_group	dev-user1	tabular	Classification	Ready	2021-06-02	
tabular_test123	basic_group	dev-user1	tabular	Classification	Working	2021-06-01	
test_hylim_0528	basic_group	dev-user1	image	Classification	Working	2021-05-28	
Test-IC-lee0527	basic_group	dev-user1	image	Classification	Completed	2021-05-27	2021-06-02
Test_ID_lee0527	basic_group	dev-user1	image	Detection	Completed	2021-05-27	2021-05-27

At the bottom right, there are buttons for 'Create project' and 'Load project'. The 'Load project' button is highlighted with a red box.

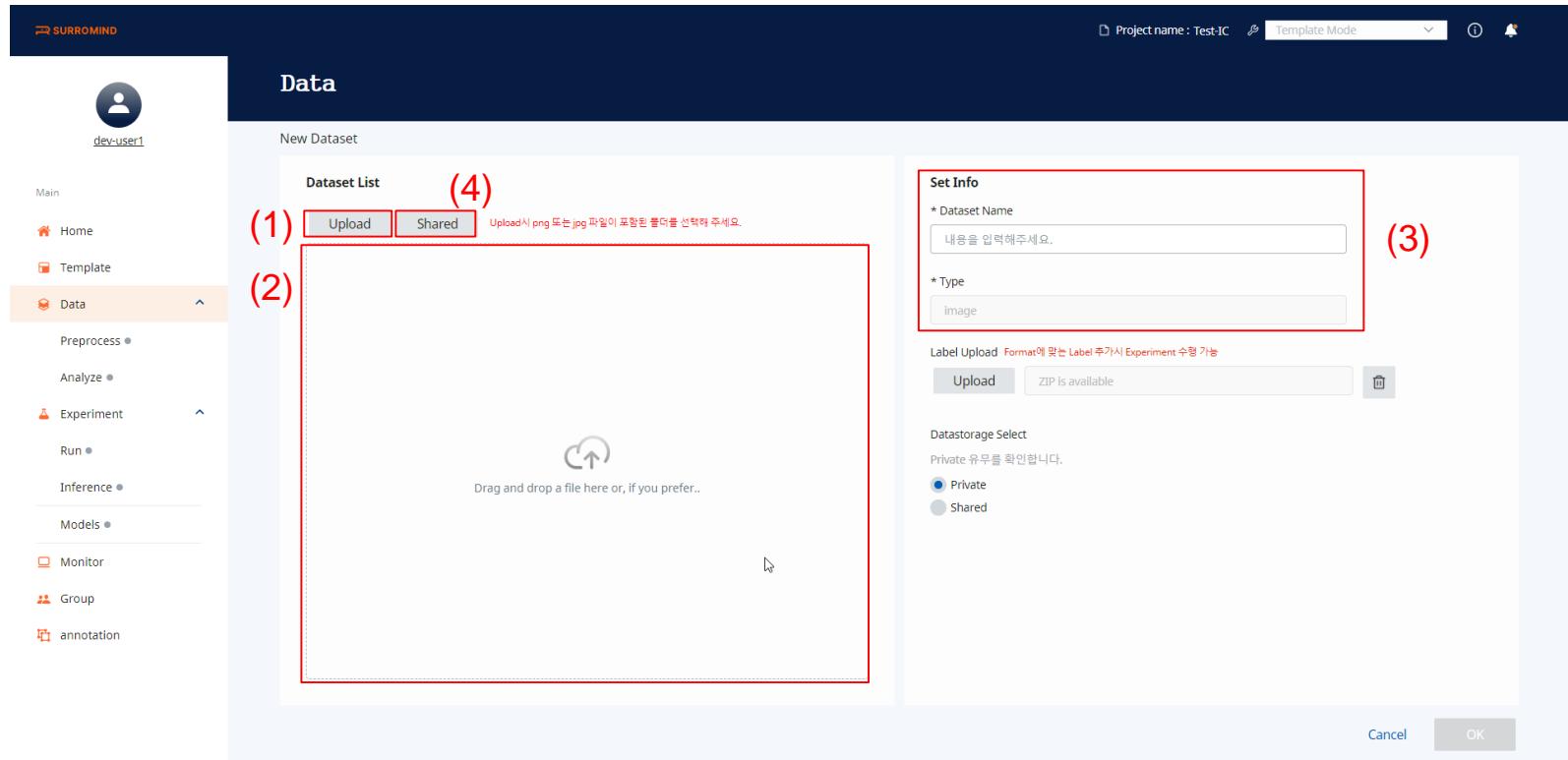
## 1.2 Dataset 등록

Data 메뉴(1)에서 New Dataset(2) 버튼을 클릭합니다.

The screenshot shows the SURROMIND Data interface. On the left, there is a sidebar with various menu items: Main, Home, Template, Data (highlighted with a red box labeled '(1)'), Preprocess, Analyze, Experiment (highlighted with a red box), Run, Inference, Models, Monitor, Group, and annotation. The main content area is titled 'Data' and shows a table with columns: DATA SET NAME, VERSION, DATA TYPE, SIZE, CREATED, and LABEL. The table is currently empty and displays the message '결과가 없습니다.' (No results). At the bottom right of the main content area, there is a blue button labeled 'New Dataset +' highlighted with a red box labeled '(2)'. The top right corner of the interface shows project information: Project name : Test-IC, Template Mode, and notification icons.

## 1.2 Dataset 등록

Upload버튼(1)이나 Drag and Drop(2)을 통해 데이터를 업로드 한 뒤, 기본정보(3)를 입력하여 data를 생성합니다.  
Shared 버튼(4)을 통해 그룹 내에서 공유된 데이터셋을 가져올 수 있습니다.



## 1.2 Dataset 등록

Annotation 작업을 통해 Label을 등록할 것이기 때문에 Label Upload(1)는 하지 않습니다.  
데이터의 공개여부(2)를 체크합니다. Shared를 선택하는 경우, 사용자 Group 내에서 공유됩니다.

The screenshot shows the SURROMIND platform interface for creating a new dataset. On the left, a sidebar menu includes 'Home', 'Template', 'Data' (selected), 'Preprocess', 'Analyze', 'Experiment', 'Run', 'Inference', 'Models', 'Monitor', 'Group', and 'annotation'. The main area is titled 'Data' and shows a 'Dataset List' with tabs for 'Upload' (selected) and 'Shared'. A central box says 'Drag and drop a file here or, if you prefer...' with a circular arrow icon. To the right, a 'Set Info' panel contains fields for 'Dataset Name' (placeholder: 내용을 입력해주세요.) and 'Type' (placeholder: image). Below these are sections for 'Label Upload' (with a note about ZIP files) and 'Datastorage Select' (with radio buttons for 'Private' and 'Shared'). Red numbers (1) and (2) highlight the 'Label Upload' section and the 'Datastorage Select' section respectively. At the bottom are 'Cancel' and 'OK' buttons.

Project name : Test-IC Template Mode

dev-user1

Main

Home

Template

Data

Preprocess

Analyze

Experiment

Run

Inference

Models

Monitor

Group

annotation

Data

New Dataset

Dataset List

Upload Shared

Upload 시 png 또는 jpg 파일이 포함된 폴더를 선택해 주세요.

Drag and drop a file here or, if you prefer...

Set Info

\* Dataset Name

내용을 입력해주세요.

\* Type

image

Label Upload Format에 맞는 Label 주가시 Experiment 수령 가능

Upload ZIP is available

(1)

Datastorage Select

Private 유무를 확인합니다.

Private Shared

(2)

Cancel OK

## 1.3 데이터 전처리

Dataset의 우측의 More 버튼(1)을 선택 후 Preprocess 버튼(2)을 클릭합니다.

The screenshot shows the SURROMIND Data interface. On the left is a sidebar with navigation links: Home, Template, Data (selected), Preprocess, Analyze, Experiment, Run, Inference, Models, Monitor, Group, and annotation. The main area is titled "Data" and shows a list of datasets under the "Private" tab. A search bar at the top allows inputting "Dataset Name(current)" with placeholder text "김생아를 입력해주세요.". Below the search bar is a toolbar with "Days", "Weeks", "Months", and a date input field. The dataset list includes columns for Dataset Name, Version, Data Type, Size, Created, and Label. A single dataset named "test\_dataset" is listed with version 1, data type "image", size 3.2446 MB, created on 2021-06-03, and no label. To the right of the dataset list is a context menu with options: Preprocess (highlighted with a red box and labeled (2)), Details, Add Label, and Download Dataset. A red box labeled (1) highlights the "More" button in the top right corner of the dataset row. At the bottom of the page are navigation arrows, a "New Dataset" button, and an "Export to Experiment" button.

Dataset Name(current)	VERSION	DATA TYPE	SIZE	CREATED	LABEL
test_dataset	1	image	3.2446 MB	2021-06-03	

(1) More button  
(2) Preprocess button

## 1.3 데이터 전처리

원하는 Method(1)를 선택한 후, Parameter(2)를 조정하여 Apply(4)를 클릭하면 전처리가 진행됩니다. 파라미터를 Default값으로 되돌리고 싶으면 Initialize(3), 전처리 완료 후 결과를 보고 싶으면 Result(5)를 클릭합니다.

The screenshot shows the SURROMIND Data Preprocess interface. On the left, a sidebar menu includes 'Home', 'Template', 'Data' (selected), 'Preprocess' (selected), 'Analyze', 'Experiment', 'Run', 'Inference', 'Models', 'Monitor', 'Group', and 'annotation'. The main area is titled 'Data' and 'Data > Preprocess'. It displays 'Dataset Info' for 'test\_dataset': Type 'image', Num Instances '42', Size '3.2446 MB', and Version '1'. To the right, a 'Choose Preprocess Method' section is highlighted with a red box, showing 'Flip' selected. Below it, the 'Parameters' section is also highlighted with a red box, showing 'flipCode' set to '1'. At the bottom, three buttons are labeled with red numbers: '(3) Initialize', '(4) Apply', and '(5) Result'. A blue 'OK' button is at the bottom right. The top right of the interface shows 'Project name : Test-IC', 'Template Mode', and notification icons.

## 1.4 Export to Experiment

학습할 Dataset을 선택 후 Export to Experiment(1)를 클릭합니다. 이 과정을 통해 학습할 수 있는 draft 상태의 Experiment를 생성합니다. Experiment(2)메뉴에서 확인 해보실 수 있습니다.

The screenshot shows the SURROMIND Data interface. On the left, there is a sidebar with user information (dev-user1) and navigation links: Home, Template, Data (selected), Preprocess, Analyze, Experiment (highlighted with a red box and labeled (2)), Run, Inference, Models, Monitor, Group, and annotation. The main area is titled 'Data' and shows a list of datasets under the 'Private' tab. A search bar at the top allows for filtering datasets by name. The table below lists four datasets:

Dataset Name (current)	VERSION	DATA TYPE	SIZE	CREATED	LABEL
test	2	image	3.3172 MB	2021-06-03	label.zip
test	1	image	3.2497 MB	2021-06-03	label.zip
test_dataset	2	image	3.3121 MB	2021-06-03	
test_dataset	1	image	3.2446 MB	2021-06-03	

At the bottom right, there are buttons for 'New Dataset' and 'Export to Experiment' (highlighted with a red box and labeled (1)). The status bar at the top right indicates 'Project name : Test-IC' and 'Template Mode'.

## 2. 학습하기

2.1 Run

2.2 HPO

2.3 AutoML

2.4 학습 완료 후

## 2.1 Run

저희 Studio에는 총 3가지 모드로 학습할 수 있습니다.

Data 메뉴에서 draft 상태(1)의 Experiment의 More버튼(2)을 클릭하면 Run/HPO/AutoML(3) 총 3가지의 학습 메뉴가 나타납니다.

The screenshot shows the SURROMIND Studio interface. On the left is a sidebar with navigation links: Home, Template, Data, Preprocess, Analyze, Experiment (which is selected and highlighted in orange), Run, Inference, Models, Monitor, Group, and annotation. The main area is titled "Experiment". It displays a table with one row of data:

ID	MODEL NAME	ALGORITHM NAME	DATASET NAME	STATUS	CREATED	RUN TIME	DETAILS
996	●		test / v2	(1) draft	2021-06-03	0m	(2)

Three specific areas are highlighted with red boxes and numbered 1, 2, and 3:

- (1) The "STATUS" column, which contains the word "draft".
- (2) The "DETAILS" button next to the status, which has a tooltip showing "Run", "HPO", and "AutoML".
- (3) The tooltip itself, which lists "Run", "HPO", and "AutoML".

## 2.1 Run

Run으로 학습을 시킬 경우, 사용자가 직접 Data Split(1), Algorithm(2), Resource(3)를 설정해주어야 합니다.

The screenshot shows the SURROMIND platform's 'Run' configuration screen. At the top, there is a dark header bar with the 'SURROMIND' logo, project name 'Test-IC', 'Template Mode' dropdown, and user notification icons. On the left, a sidebar menu lists categories: Main, Home, Template, Data, Preprocess, Analyze, Experiment, Inference, Models, Monitor, Group, and annotation. The 'Experiment' category is currently selected and highlighted in orange. The main content area is titled 'Run' and contains a 'Set Configuration' section with three tabs: 'Data Split' (selected), 'Algorithm', and 'Resource'. Below the tabs, the word 'Configuration' is written in red with large parentheses around '(1)', '(2)', and '(3)'. A table with four columns ('Data', 'Algorithm', 'Parameter', 'Resource') is shown, but it is currently empty. To the right of the table is a 'Start Train' button. At the bottom, there are two panels: 'Log' (containing a black redacted area) and 'Progress' (containing a gray redacted area).

## 2.1 Run

모든 Setting이 완료되면 Start Train을 클릭하여 학습을 시작할 수 있습니다.

The screenshot shows the SURROMIND platform's 'Run' configuration page. The left sidebar includes a user profile for 'dev-user1' and sections for Main, Home, Template, Data, Preprocess, Analyze, Experiment (with 'Run' selected), Inference, Models, Monitor, Group, and annotation. The main area has tabs for Set Configuration (Data Split, Algorithm, Resource) and Configuration. The Configuration table details the experiment setup:

Data	Algorithm	Parameter	Resource
Name: test	Name: resnet18	batch_size: 4	Name: Preset3
Train: 80(%)	Size: $1 \times 10^6$	epoch: 10	GPU: RTX3090 1, 24000
Validation: 10(%)	learning_rate: 0.001	CPU: 4 core	
Test: 10(%)	momentum: 0.9	RAM: 16 GB	
	weight_decay: 0.0005		

A large red button labeled 'Start Train' is positioned at the bottom right of the configuration area. Below it, there are two panels: 'Log' (black placeholder) and 'Progress' (grey placeholder).

## 2.1 Run

학습이 진행되면 실시간으로 Log와 Progress 그래프를 볼 수 있습니다. 다른 메뉴로 이동해도 계속 진행됩니다.

The screenshot shows the SURROMIND platform's 'Run' interface. On the left, a sidebar menu includes 'Main', 'Home', 'Template', 'Data', 'Preprocess', 'Analyze', 'Experiment' (selected), 'Inference', 'Models', 'Monitor', 'Group', and 'annotation'. The 'Run' section is highlighted. In the center, a 'Set Configuration' panel has tabs for 'Data Split' (selected), 'Algorithm', and 'Resource'. The 'Configuration' table lists parameters:

Data	Algorithm	Parameter	Resource
Name: test	Name: resnet18	batch_size: 4	Name: Preset3
Train: 80(%)	Size: 1X10 <sup>6</sup>	epoch: 10	GPU: RTX3090 1, 24000
Validation: 10(%)	learning_rate: 0.001	momentum: 0.9	CPU: 4 core
Test: 10(%)	weight_decay: 0.0005	RAM: 16 GB	

The 'Log' section displays command-line output:

```
tensorrt/tensorrt_executor/platform/default/dso_loader.cc:55] Successfully opened dynamic library libcurl.so.11
2021-06-03 01:50:30.966329 [I tensorflow/stream_executor/cuda/cuda_blas.cc:183]
TensorFloat-32 will be used for the matrix multiplication. This will only be logged once.

train 1 / 10: 11% | 1/9 [00:02<00:16, 2.01it/s]
train 1 / 10: 22% | 2/9 [00:02<00:06, 1.03it/s]
train 1 / 10: 33% | 3/9 [00:02<00:04, 1.42it/s]
train 1 / 10: 44% | 4/9 [00:03<00:02, 1.74it/s]
train 1 / 10: 56% | 5/9 [00:03<00:01, 2.21it/s]
train 1 / 10: 67% | 6/9 [00:03<00:01, 2.31it/s]
train 1 / 10: 78% | 7/9 [00:03<00:00, 2.56it/s]
train 1 / 10: 89% | 8/9 [00:04<00:00, 2.82it/s]
train 1 / 10: 100% | 9/9 [00:04<00:00, 2.90it/s]
```

The 'Progress' section shows accuracy metrics over 10 epochs:

Epoch	train_accuracy	valid_accuracy
1	40	45
2	55	50
3	60	58
4	50	65
5	65	68
6	55	70
7	60	95
8	65	85
9	70	75
10	75	40

## 2.2 HPO

HPO로 학습을 시킬 경우, 사용자가 직접 Data Split, Algorithm, Resource를 설정해주어야 하는 것은 Run과 동일합니다. 단, Algorithm화면에서 HPO를 적용할 파라미터를 체크해야 합니다. HPO 여부가 체크된 항목은 최적의 성능을 찾기 위해 내부 알고리즘에 따라 자동으로 값을 바꾸어 여러 번 학습됩니다.

The screenshot shows the SURROMIND platform's 'Run / Hyper Parameter Optimization' interface. On the left, there is a sidebar with user information (dev-user1) and navigation links for Main, Home, Template, Data, Preprocess, Analyze, Experiment, Run, Inference, Models, Monitor, Group, and annotation. The 'Run' section is currently selected. The main area displays the 'Hyper Parameter Optimization (HPO)' configuration screen. A modal window titled 'Algorithm' is open, listing two algorithms: 'resnet18' and 'densenet121'. Below this, another modal window titled 'Parameter' shows four hyperparameters: 'batch\_size' (value 4), 'epoch' (value 10), 'learning\_rate' (value 0.001), and 'momentum' (value 0.9). The 'learning\_rate' row has a red box drawn around its 'HPO 여부' (HPO Enabled) checkbox, which is checked. At the bottom of the 'Parameter' modal are 'Cancel' and 'OK' buttons.

## 2.2 HPO

Start HPO를 클릭하면 나타나는 HPO Settings입니다. HPO 구동에 관한 설정(1)과 모델 성능 평가를 위한 지표(2)를 설정합니다.

The screenshot shows the SURROMIND platform's 'Run / Hyper Parameter Optimization' interface. A modal window titled 'HPO Settings' is displayed, containing two main sections: 'HPO Settings' and 'Objective'. The 'HPO Settings' section includes fields for MaxTrialCount (10), ParallelTrialCount (1), and MaxFailedTrialCount (4). The 'Objective' section includes fields for ObjectiveMetricName (validation\_accuracy), Type (maximize), Goal (90), and AdditionalMetricNames (accuracy). A large blue 'Start HPO' button is visible in the background.

(1)

(2)

## 2.2 HPO

앞에서 설정한 ObjectMetric(1)이 Goal이상 도달할 경우 HPO가 중단되며 이전 단계에서 HPO여부가 체크되어 내부에서 선택된 파라미터값(2)을 확인할 수 있습니다. 원하는 Trial을 선택한 후 Submit(3) 할 수 있습니다. Trial을 Submit 하기 전까지 해당 Experiment는 계속 running상태로 유지됩니다.

The screenshot shows the SURROMIND platform interface for Hyper Parameter Optimization. The left sidebar shows the user profile (dev-user1) and navigation menu with 'Run' selected. The main area has tabs for Data Split, Algorithm, and Resource, with Configuration selected. The Configuration table includes columns for Data, Algorithm, Parameter, Resource, and HPO Settings. The Trial List table shows two trials: hpo-20210603-105812-l4gdj5nw and hpo-20210603-105812-b7jg4frb, with columns for Trial Name, Status, Validation Accuracy, Accuracy, Learning Rate, Batch Size, and Details. The Validation Accuracy column for the first trial is highlighted with a red box and labeled (1). The Learning Rate and Batch Size columns for both trials are also highlighted with red boxes and labeled (2). A large red box highlights the 'Submit' button at the bottom right, labeled (3).

Data	Algorithm	Parameter	Resource	HPO Settings
Name: test Train: 80% Validation: 10% Test: 10%	Name: resnet18 Size: $1 \times 10^6$	batch_size: epoch learning_rate: momentum weight_decay	Name: GPU CPU RAM Preset: Preset3 RTX3090 1, 24000 4 core 16 GB	Max Trial Count: 10 Parallel Trial Count: 1 Max Failed Trial Count: 4 Objective Type: maximize Objective Goal: 90

Trial List					
TRIAL NAME	STATUS	VALIDATION_ACCURACY	ACCURACY	LEARNING_RATE	BATCH_SIZE
hpo-20210603-105812-l4gdj5nw	succeeded	50.0	52.9412	0.1	32
hpo-20210603-105812-b7jg4frb	succeeded	100.0	88.2353	0.001	4

(1) (2) (3)

## 2.3 AutoML

AutoML은 Run이나 HPO와는 달리 별다른 설정 없이 바로 학습을 진행할 수 있습니다. 원하는 알고리즘을 선택(모두 선택 가능)후 바로 Start AutoML(1)을 클릭하여 학습을 시작해주세요. 알고리즘별 Data split, Parameter값을 변경(2)할 수도 있습니다.

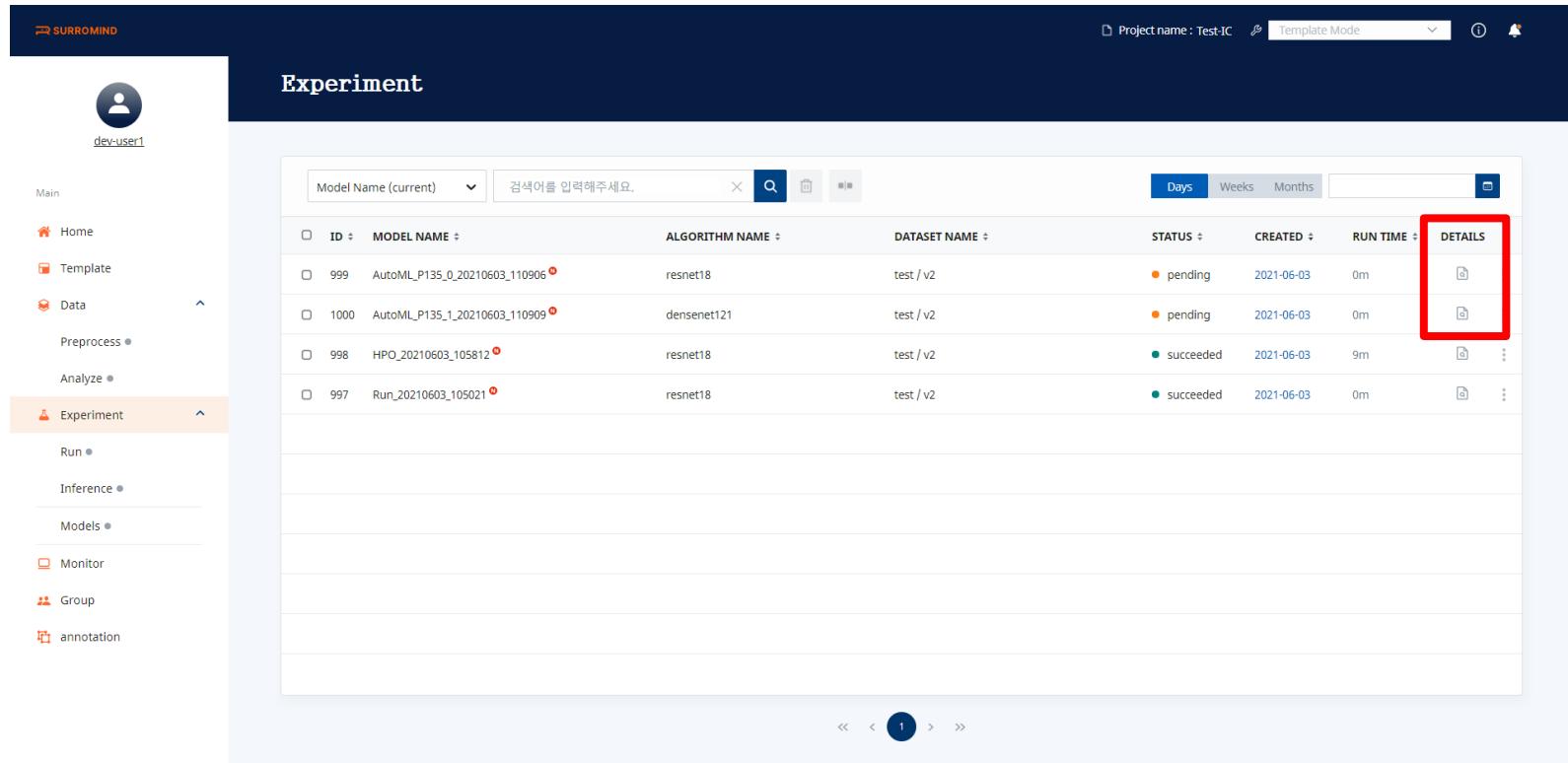
The screenshot shows the SURROMIND platform's Run / AutoML interface. On the left, a sidebar menu is visible with options like Home, Template, Data, Preprocess, Analyze, Experiment, Run, Inference, Models, Monitor, Group, and annotation. The 'Run' option is currently selected and highlighted in orange. The main area is titled 'Run / AutoML' and contains two main sections:

- Algorithm List:** A table showing two algorithms:

INDEX	ALGORITHM NAME	COMPLEXITY	SIZE(#Parmas)	RESOURCE NEEDS
1	resnet18	$1 \times 10^6$	11.37	
2	densenet121	$5 \times 10^6$	7.41	
- Configuration Panel:** This panel is divided into two sections by a red box:
  - (1) Start AutoML:** A large blue button at the bottom of the main panel.
  - (2) Configuration Options:** A detailed configuration section for the selected algorithm 'resnet18'. It includes fields for Data (Data name: test), Train (80 %), Validation (10 %), Test (10 %), Algorithm (Algorithm name: resnet18), Parameter (batch\_size: 4, epoch: 10, learning\_rate: 0.001, momentum: 0.9).

## 2.3 AutoML

AutoML은 이전 단계에서 선택한 Algorithm 개수만큼 Experiment가 생성됩니다. Details을 클릭하면 Run화면과 동일한 화면을 볼 수 있습니다.



The screenshot shows the SURROMIND AutoML interface. On the left is a sidebar with navigation links: Main, Home, Template, Data, Preprocess, Analyze, Experiment (which is selected and highlighted in orange), Run, Inference, Models, Monitor, Group, and annotation. The main content area is titled 'Experiment'. It features a search bar at the top with a placeholder '검색어를 입력해주세요.' and a date filter section with 'Days', 'Weeks', and 'Months' buttons. Below is a table listing four experiments:

ID	MODEL NAME	ALGORITHM NAME	DATASET NAME	STATUS	CREATED	RUN TIME	DETAILS
999	AutoML_P135_0_20210603_110906	resnet18	test / v2	pending	2021-06-03	0m	[Details]
1000	AutoML_P135_1_20210603_110909	densenet121	test / v2	pending	2021-06-03	0m	[Details]
998	HPO_20210603_105812	resnet18	test / v2	succeeded	2021-06-03	9m	[Details]
997	Run_20210603_105021	resnet18	test / v2	succeeded	2021-06-03	0m	[Details]

At the bottom of the table, there are navigation arrows: '<<', '<', '1', '>', and '>>'. The 'DETAILS' column header is highlighted with a red box.

## 2.4 학습 완료 후

학습이 정상적으로 완료되면 Experiment가 succeeded(1)로 변경됩니다. Details(2)를 클릭하여 상세정보를 볼 수 있으며, More 버튼(3)을 클릭하여 해당 Experiment를 Test해 볼 수 있는 메뉴(4), 지금까지 설정 값들(Dataset 전처리, 학습 setting값들)을 Template으로 저장 해놓을 수 있는 메뉴(5) 모델 Repository에 등록할 수 있는 메뉴(6)들이 있습니다.

The screenshot shows the SURROMIND Experiment management interface. On the left, there is a sidebar with user information (dev-user1) and navigation links for Main, Home, Template, Data, Preprocess, Analyze, Experiment (which is currently selected and highlighted in orange), Run, Inference, Models, Monitor, Group, and annotation. The main content area is titled "Experiment". It displays a table of completed experiments (Status: succeeded). The columns are ID, MODEL NAME, ALGORITHM NAME, DATASET NAME, STATUS, CREATED, RUN TIME, and DETAILS. The first experiment (ID 999) has its DETAILS row highlighted with a red box and labeled (2). The DETAILS row for the second experiment (ID 1000) is also highlighted with a red box and labeled (3). A "More" button in the DETAILS row of the second experiment is highlighted with a red box and labeled (4). A "Save as a template" button is highlighted with a red box and labeled (5). A "Register Model" button is highlighted with a red box and labeled (6).

ID	MODEL NAME	ALGORITHM NAME	DATASET NAME	STATUS	CREATED	RUN TIME	DETAILS
999	AutoML_P135_0_20210603_110906	resnet18	test / v2	succeeded	2021-06-03	1m	(4)
1000	AutoML_P135_1_20210603_110909	densenet121	test / v2	succeeded	2021-06-03	1m	(5)
998	HPO_20210603_105812	resnet18	test / v2	succeeded	2021-06-03	9m	(6)
997	Run_20210603_105021	resnet18	test / v2	succeeded	2021-06-03	0m	Change Model Name

# 3. 평가하기

## 3.1 Inference

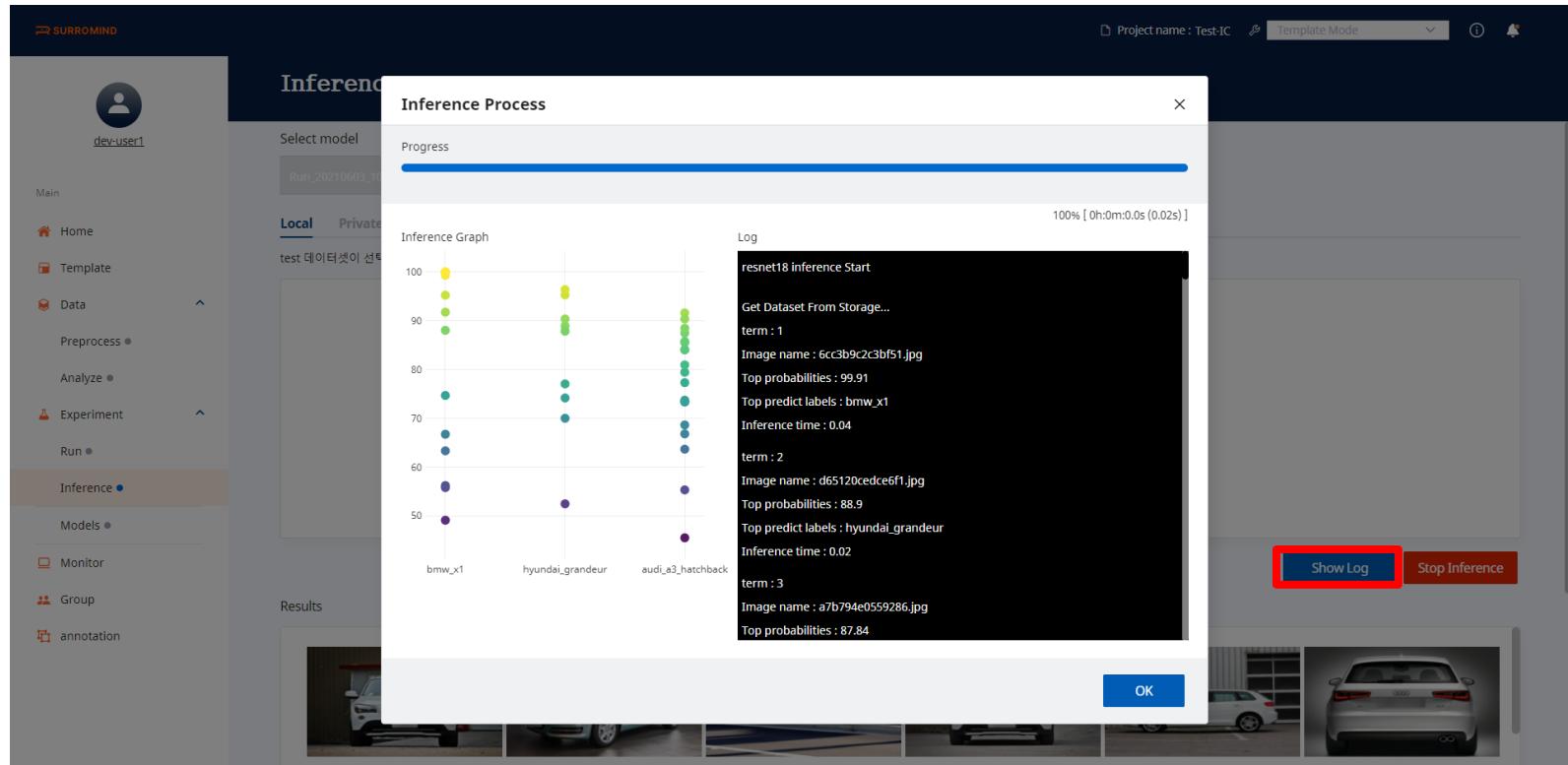
### 3.1 Inference

Experiment에서 Inference를 선택하면 보이는 화면입니다. 사용자의 PC에서 Data(1)/프로젝트 내에서만 공유된 Data(2)/그룹 내에서 공유된 Data(3)를 업로드할 수 있습니다. Data가 업로드 되면 바로 Start Inference(4)를 클릭하여 모델을 평가할 수 있습니다.

The screenshot shows the SURROMIND Inference interface. On the left is a sidebar with a user profile (dev-user1) and sections for Main, Home, Template, Data, Preprocess, Analyze, Experiment, Run, Inference (which is highlighted), Models, Monitor, Group, and annotation. The main area has a dark header with 'Project name : Test-IC' and 'Template Mode'. Below the header, the title 'Inference' is displayed. A dropdown menu labeled 'Select model' contains three items: (1) Run\_20210429\_105021(current), (2) Local, and (3) Shared. The 'Local' button is highlighted with a red box. Below the dropdown is a section titled '데이터셋을 선택해주세요.' with a 'Upload' button, a 'Delete' button, and a placeholder text 'Drag and drop a file here or, if you prefer..'. At the bottom right of this section is a red box around the 'Start Inference' button. To the left of the 'Start Inference' button is a 'Show Log' button.

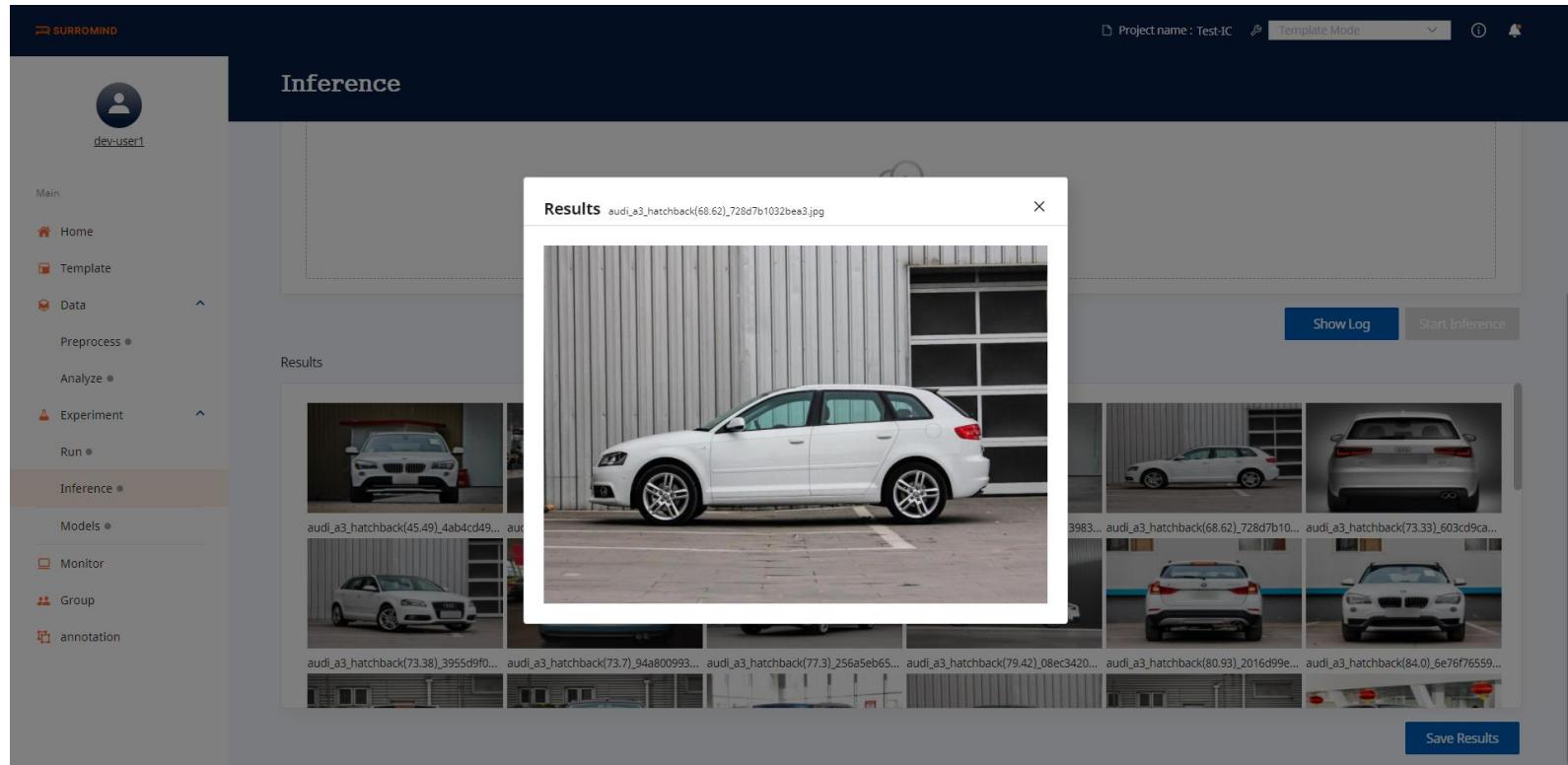
### 3.1 Inference

Show Log 버튼을 클릭하면 진행 상황을 볼 수 있습니다.



### 3.1 Inference

Inference가 완료되면 이미지를 클릭하여 결과를 확인해볼 수 있습니다. 로드한 프로젝트의 문제정의별로 Result가 다릅니다.



# 4. 재사용하기

4.1 Template

4.2 데이터 전처리

4.3 학습

## 4.1 Template

Experiment에서 Save as a template을 통해 Template으로 저장하면, 학습 시 설정한 값들(Dataset 전처리, 학습 Algorithm, Parameter 등)이 저장되어 Template을 로드만 하면 동일한 조건으로 dataset만 바꾸어 사용할 수 있습니다. 상단의 'Template mode로 변경' 영역(1)이나 Template 메뉴(2)를 통해 해당 화면으로 이동합니다. 원하는 Template 선택 후 Load Template(3)을 클릭합니다.

The screenshot shows the SURROMIND WorkFlow Template interface. On the left, there is a sidebar with a user profile for 'dev-user1' and several menu items: Main, Home, Data, Preprocess, Analyze, Experiment, Inference, Models, Monitor, Group, and annotation. The 'Template' item in the Experiment section is highlighted with a red box and labeled '(2)'. At the top right, there is a 'Template Mode' overlay window with a red border. Inside the overlay, it says 'Status: Template Mode 중지' and 'Start date:'. Below that is a blue button labeled 'Template mode로 변경'. At the bottom right of the main interface, there is another red box around a blue button labeled 'Load Template' and it is also labeled '(3)'. The main content area displays a table with columns: TEMPLATE NAME, GROUP NAME, DATASET NAME, CREATED, and CREATOR. One row is selected, showing 'IC-template' under TEMPLATE NAME, 'basic\_group' under GROUP NAME, 'test' under DATASET NAME, '2021-06-03' under CREATED, and 'dev-user1' under CREATOR. To the right of the table, there is a 'Detail' section with a line chart titled 'metric' showing 'train\_accuracy' and 'valid\_accuracy' over time steps 0 to 10. Below the chart, sections for 'Model' and 'Resource' provide specific details about the run.

## 4.2 데이터 전처리

Template이 로드되면 Template mode(1)가 됩니다. 어떤 Dataset을 선택해도 Template에 저장된 전처리 Method(1)로만 진행됩니다. Method를 변경할 수 없으며, 변경하고 싶을 경우 Template Mode를 취소(2)해야 합니다.

The screenshot shows the SURROMIND Data Preprocess interface. On the left, there's a sidebar with navigation items like Home, Template, Data, Preprocess (which is selected), Analyze, Experiment, Run, Inference, Models, Monitor, Group, and annotation. The main area has tabs for Data and Pipeline. Under Data, it shows a 'Dataset Info' table with columns: Dataset (test), Type (image), Num Instances (42), Size (3.3172 MB), and Version (2). Below that is a 'Pipeline' table with columns: NUM, DATA, and PREPROCESS, showing one row with value 1, date 2021-06-03, and method Flip. At the bottom right of the main area are 'Apply' and 'Result' buttons. A large red box labeled (1) encloses the 'Choose Preprocess Method' dialog, which has a dropdown for 'Flip', a text area for 'About' (Flip/mirror input images horizontally/vertically.), and a 'Parameters' section with a 'flipCode' input field containing the value 1. An 'OK' button is at the bottom right of this dialog. Above the pipeline table, a smaller red box labeled (2) encloses a 'Template Mode' dialog with a status message 'Status: 동작중 Start date : 2021-06-03 11:16:25' and a 'Template mode 취소' button.

## 4.3 학습

Template이 로드되면 Template mode가 됩니다. Template에 저장된 학습 Setting으로 진행되기 때문에 Run 메뉴만 나타납니다.

The screenshot shows the SURROMIND platform interface in 'Template Mode'. The top navigation bar includes 'Project name : Test-IC', 'Template Mode', and user icons. The left sidebar menu is collapsed, showing 'dev-user1' at the top, followed by 'Main', 'Home', 'Template', 'Data', 'Preprocess', 'Analyze', 'Experiment' (which is expanded), 'Run', 'Inference', 'Models', 'Monitor', 'Group', and 'annotation'. The main content area is titled 'Experiment' and displays a table of model runs. The table columns are: ID, MODEL NAME, ALGORITHM NAME, DATASET NAME, STATUS, CREATED, RUN TIME, and DETAILS. The table rows show the following data:

ID	MODEL NAME	ALGORITHM NAME	DATASET NAME	STATUS	CREATED	RUN TIME	DETAILS
1001	AutoML_P135_0_20210603_110906	resnet18	test / v2	draft	2021-06-03	0m	Run
999	AutoML_P135_1_20210603_110909	densenet121	test / v2	succeeded	2021-06-03	1m	Change Model Name
1000	HPO_20210603_105812	resnet18	test / v2	succeeded	2021-06-03	1m	
998	Run_20210603_105021	resnet18	test / v2	succeeded	2021-06-03	9m	
997				succeeded	2021-06-03	0m	

Pagination controls at the bottom indicate page 1 of 1.

## 4.3 학습

Template mode인 경우에는 Template에 저장된 학습 Setting으로만 진행됩니다. Setting 값을 변경할 수 없으며, 변경하고 싶을 경우 Template Mode를 취소해야 합니다. Resource만 변경 가능합니다.

The screenshot shows the SURROMIND platform's 'Run' configuration interface. On the left, there is a sidebar with user information (dev-user1) and navigation links: Main, Home, Template, Data, Preprocess, Analyze, Experiment, Run (highlighted in orange), Inference, Models, Monitor, Group, and annotation. The main area has tabs for 'Data Split', 'Algorithm', and 'Resource' (selected). Below these tabs is a 'Configuration' table:

Data	Algorithm	Parameter	Resource
Name: Train	Name: resnet18	batch_size: 4	Name: Preset3
Validation	Size: 1 X 10 <sup>6</sup>	epoch: 10	GPU: RTX3090 1, 24000
Test		learning_rate: 0.001	CPU: 4 core
		momentum: 0.9	RAM: 16 GB
		weight_decay: 0.0005	

Below the table is a 'Start Train' button. To the right of the table are two sections: 'Log' (black placeholder) and 'Progress' (grey placeholder).

# 5. 모델 저장하기

5.1 Models

5.2 포맷 변환

## 5.1 Models

Experiment에서 Register Model을 통해 저장하면, 같은 그룹 내에서 공유할 수 있도록 Model Repository에 등록이 됩니다. 모델을 선택하면 상세 화면으로 이동됩니다.

The screenshot shows the SURROMIND platform's Model Repository page. On the left, there is a sidebar with navigation links: Home, Template, Data, Preprocess, Analyze, Experiment, Run, Inference, Models (which is highlighted in orange), Monitor, Group, and annotation. The main area has a dark header with a user icon and the text "dev-user1". Below the header is a search bar with dropdowns for "Project name: Test-IC" and "Template Mode". The main content area is titled "Models" and contains a table titled "Model Repository". The table has columns: MODEL NAME, GROUP NAME, VERSION, VERSION LIMIT, CREATED, and UPDATED. A red box highlights the row for "gg", which has a version of 3 and was created on 2021-05-18 16:29:08 and updated on 2021-05-28 17:58:52. The table also lists other models like "cluster-transfer-test", "hae\_model", "tag\_test", etc. At the bottom of the table, there are navigation arrows and a page number indicator (1).

MODEL NAME	GROUP NAME	VERSION	VERSION LIMIT	CREATED	UPDATED
cluster-transfer-test	basic_group	3	-	2021-05-28 11:36:32	2021-06-01 10:19:59
gg	basic_group	3	-	2021-05-18 16:29:08	2021-05-28 17:58:52
hae_model	basic_group	2	-	2021-05-14 14:33:13	2021-05-14 14:36:04
tag_test	basic_group	5	-	2021-05-07 14:20:37	2021-05-10 10:15:26
semantic-segmentation-pytorch-model	basic_group	12	-	2021-05-03 17:46:22	2021-05-18 13:07:21
tabular-classification-model1	basic_group	1	-	2021-05-03 16:30:02	2021-05-03 16:30:02
IC-resNet-format-conversion	basic_group	7	-	2021-04-30 15:04:53	2021-05-07 14:02:48
IC-denseNet-format-conversion	basic_group	5	-	2021-04-30 15:03:44	2021-05-07 14:52:50
CCTV	basic_group	2	-	2021-04-29 17:55:12	2021-05-03 16:05:43
test	basic_group	1	-	2021-04-28 17:45:00	2021-04-28 17:45:00
PT_OD_Auto	basic_group	8	-	2021-04-28 17:25:45	2021-04-28 17:26:24

## 5.1 Models

같은 이름으로 모델을 등록하면 Version up(1)이 되어 저장됩니다. 최대 버전 수를 지정(2)하여 자동 삭제(3)되도록 설정할 수 있습니다. 또한, 모델을 클릭(4)하여 모델 환경 설정을 할 수 있습니다.

The screenshot shows the SURROMIND platform's Model Repository. On the left, a sidebar menu includes Main, Home, Template, Data, Preprocess, Analyze, Experiment, Run, Inference, Models (which is selected and highlighted in orange), Monitor, Group, and annotation. The Models section contains a sub-menu with Home, Template, and Data. The main content area is titled 'Models' and shows a 'Model Repository' for a model named 'gg'. The repository displays three versions of the model, each created by 'dev-user1'. The versions are listed in a table:

VERSION	DEVELOPER	STATUS	CREATED	UPDATED
1	dev-user1		2021-05-18 16:29:08	2021-05-28 18:25:00
3	dev-user1		2021-05-28 17:58:52	2021-05-28 17:58:52
2	dev-user1		2021-05-18 16:29:50	2021-05-18 16:29:50

Below the table, a note states: "지정한 최대 버전 수 초과시 가장 오래된 버전부터 자동 삭제됩니다." (Automatically delete the oldest version when the specified maximum version count is exceeded). A modal window for 'gg' shows the creation date '2021-05-18 16:29:08', update date '2021-05-28 17:58:52', and two configuration options: '(2) 자동 삭제' (Automatic Deletion) with a radio button and '(3) 최대 버전 수' (Maximum Version Count) with a text input field set to '3'. A 'Save' button is at the bottom of the modal. Red boxes highlight four specific areas: (1) the 'VERSION' column header of the table, (2) the '최대 버전 수' input field in the modal, (3) the '자동 삭제' radio button in the modal, and (4) the row containing version 3 in the table.

## 5.1 Models

모델을 Download(1)하거나 Description(2), 포맷 변환(3), Tag설정(4)을 하면 Log(5)에 기록됩니다.

The screenshot shows the SURROMIND Model Repository interface. On the left, there is a sidebar with navigation links: Home, Template, Data, Preprocess, Analyze, Experiment, Run, Inference, Models (which is highlighted in orange), Monitor, Group, and annotation. The main area is titled "Models" and contains a "Model Repository" section. It displays a model entry for "gg-Version1". The entry includes details such as Add date: 2021-05-18 16:29:08, Modified date: 2021-05-28 18:25:00, Python Ver: 3.6.9, Framework: Pytorch, ID: 056d752232d74273b9cded13e23d0a6c, and Developer: dev-user1. Below this, there are five numbered callout boxes:

- (1) A red box highlights the "gg-Version1" model name.
- (2) A red box highlights the "Description" button.
- (3) A red box highlights the "Format Conversion" section, which shows "현재 포맷" (Pytorch) and "변환할 포맷" (ONNX) fields with a "Start" button.
- (4) A red box highlights the "Tag" button.
- (5) A red box highlights the "Log" table, which has columns: INDEX, MODIFIED, CONTENT, and MODIFIER. The table shows one entry: INDEX 1, MODIFIED 2021-05-18 16:29:05, CONTENT ONNX conversion completed, and MODIFIER dev-user1.

## 5.2 포맷 변환

모델의 Framework를 Pytorch나 Tensorflow에서 ONNX로 변환할 수 있습니다.

The screenshot shows the SURROMIND Model Repository interface. On the left, there is a sidebar with various project management and monitoring options. The main area is titled 'Models' and displays a list of models. One model, 'gg-Version1', is selected and shown in detail. The 'Format Conversion' section is highlighted with a red box. It contains fields for '현재 포맷' (Current Format) set to 'Pytorch' and '변환할 포맷' (Convert to Format) set to 'ONNX'. A 'Start' button is located below these fields. Below this section, there are sections for 'Tag' and 'Log', which show a single log entry indicating an ONNX conversion was completed.

Project name : Test-IC Template Mode ⓘ ⚡

Model Repository

gg-Version1 ↴

Add date : 2021-05-18 16:29:08 Python Ver : 3.6.9  
Modified date : 2021-05-28 18:25:00 Framework : Pytorch  
ID : 056d752232d74273b9cded13e23d0a6c Developer : dev-user1

Description ↴

▶ 포맷 변환

현재 포맷 : Pytorch      변환할 포맷 : ONNX

Start

▶ Tag

▼ Log

INDEX	MODIFIED	CONTENT	MODIFIER
1	2021-05-18 16:29:50	ONNX conversion completed	dev-user1

# 6. Annotation

6.1 일감 발행

6.2 공동 Annotation 작업(작업 분배)

6.3 Annotation 시작

6.4 화면 구성

6.5 Auto Annotation

6.6 작업 저장과 완료

6.7 Image Classification

6.8 Object Detection

6.9 Instance Segmentation

## 6.1 일감 발행

방금 생성한 Dataset의 우측의 More 버튼(1)을 선택 후 Add Label 버튼(2)을 클릭합니다.

The screenshot shows the SURROMIND Data management interface. On the left is a sidebar with navigation links like Home, Template, Data (which is selected and highlighted in orange), Preprocess, Analyze, Experiment, Run, Inference, Models, Monitor, Group, and annotation. The main area is titled "Data" and shows a list of datasets under the "Private" tab. The list includes:

Dataset Name (current)	VERSION	DATA TYPE	SIZE	CREATED	LABEL
test-annotation	1	image	3.2446 MB	2021-06-03	label.zip
test	2	image	3.3172 MB	2021-06-03	label.zip
test	1	image	3.2497 MB	2021-06-03	label.zip
test_dataset	2	image	3.3121 MB	2021-06-03	label.zip
test_dataset	1	image	3.2446 MB	2021-06-03	label.zip

A context menu is open over the first dataset in the list, with two items highlighted in red boxes:

- (1) More: A dropdown menu with options: Preprocess, Details, Label (highlighted), and Download Dataset.
- (2) Add Label: A button in the dropdown menu.

At the bottom of the page, there are navigation buttons (double arrows, single arrows, page number 1, double arrows) and buttons for "New Dataset" and "Export to Experiment".

## 6.2 공동 Annotation 작업(작업 분배)

Annotator를 통해 Label을 추가할 경우, 보이는 화면입니다. 아래 화면은 로그인한 계정의 그룹에 이미 Annotator 3명과 개발자 1명이 있는 상태입니다. 작업을 분배할 대상을 선택(1)하면 이미지 개수/작업자 인원수로 자동 분배(2)합니다.

(1)

INDEX	NAME	START	END
1	dev-user1(mine)	0	0
2	anno-user1	0	0
3	anno-user2	0	0
4	anno-user3	0	0

선택한 Annotation의 수 만큼 자동으로 이미지 갯수를 분배합니다.

Cancel OK

(2)

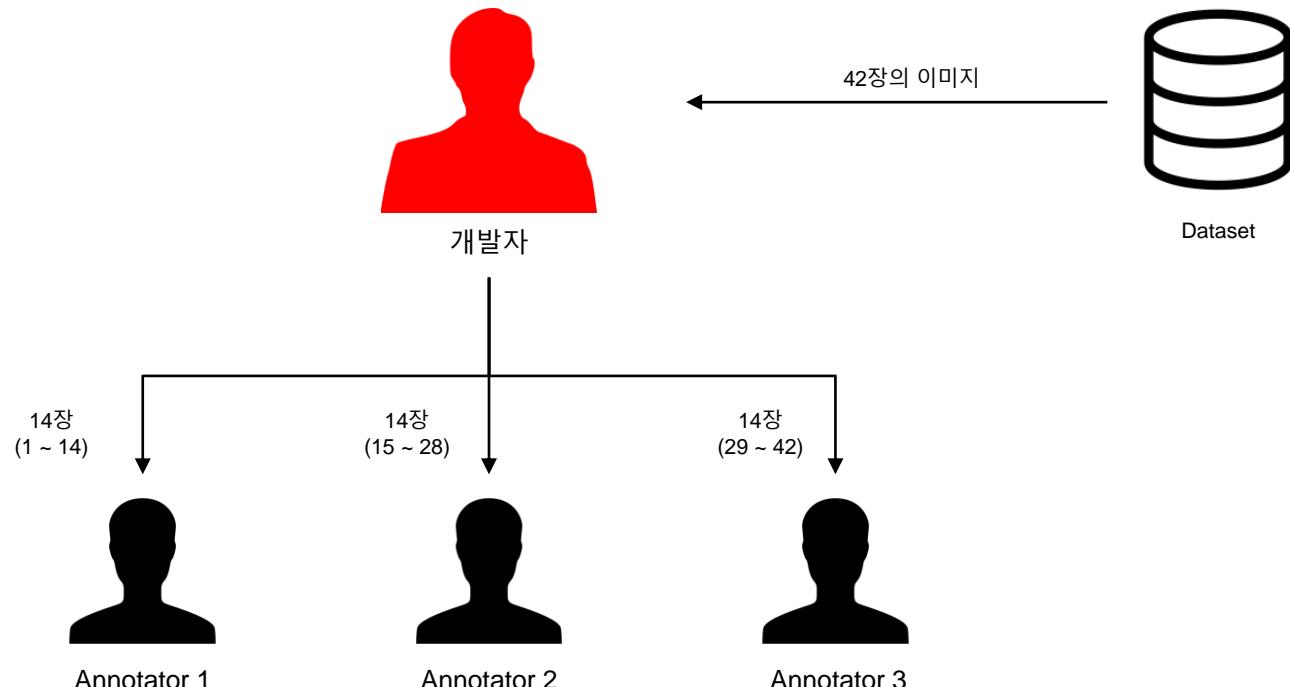
INDEX	NAME	START	END
1	dev-user1(mine)	0	0
2	anno-user1	1	14
3	anno-user2	15	28
4	anno-user3	29	42

선택한 Annotation의 수 만큼 자동으로 이미지 갯수를 분배합니다.

Cancel OK

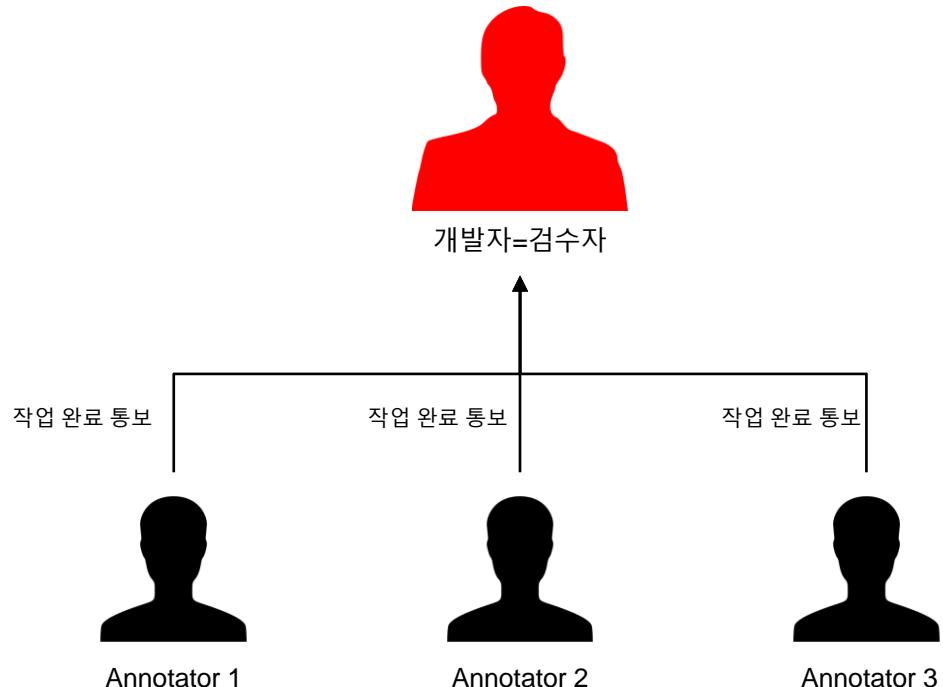
## 6.2 공동 Annotation 작업(작업 분배)

만약 개발자가 Annotator 3명에 작업을 분배했다면, 각 Annotator들은 ‘이미지 개수/작업자’ 만큼 일감을 할당 받습니다.



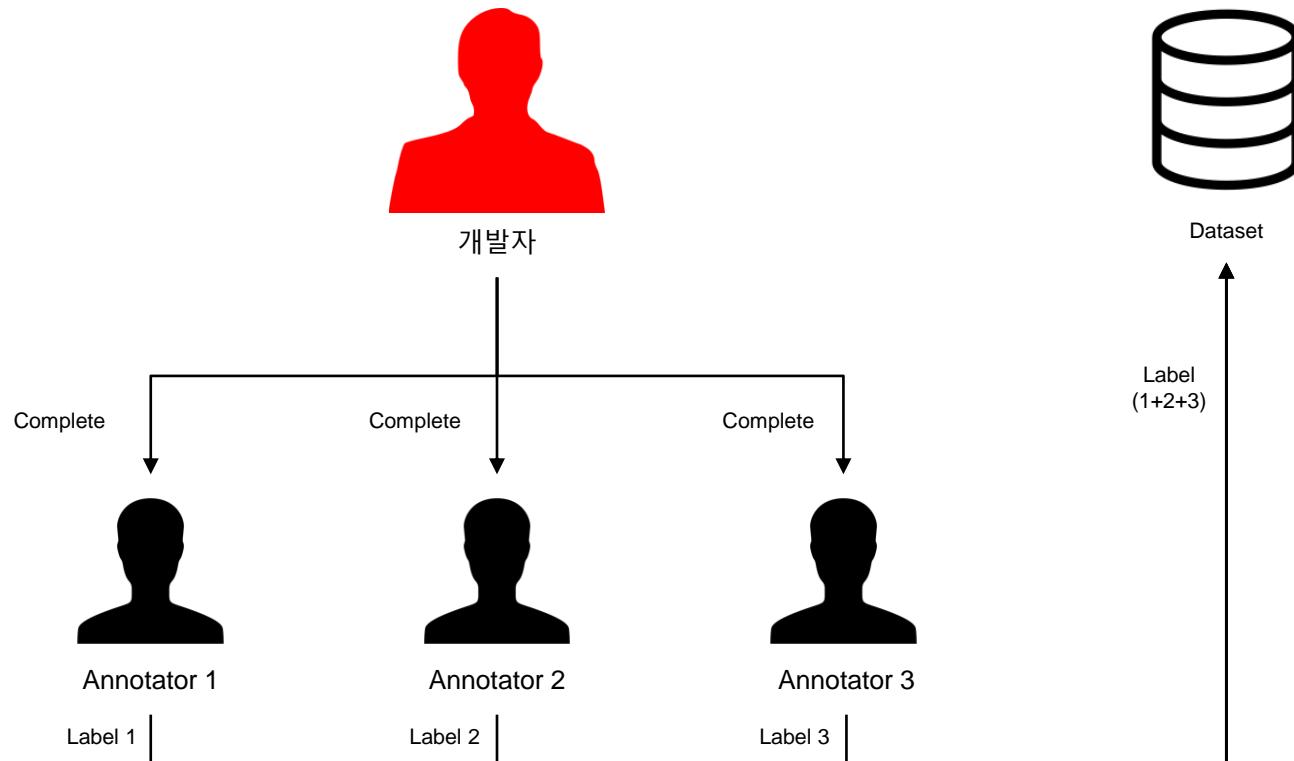
## 6.2 공동 Annotation 작업(작업 분배)

개발자는 Annotator 작업자들의 ‘검수자’ 역할을 합니다. 각 Annotator들은 작업을 마친 후 검수자에게 완료 통보를 합니다.



## 6.2 공동 Annotation 작업(작업 분배)

분배되었던 모든 Annotation 작업이 완료 처리가 되면, 모든 Label을 하나로 합쳐 자동으로 해당 Dataset에 Label이 추가됩니다.



## 6.3 Annotation 시작

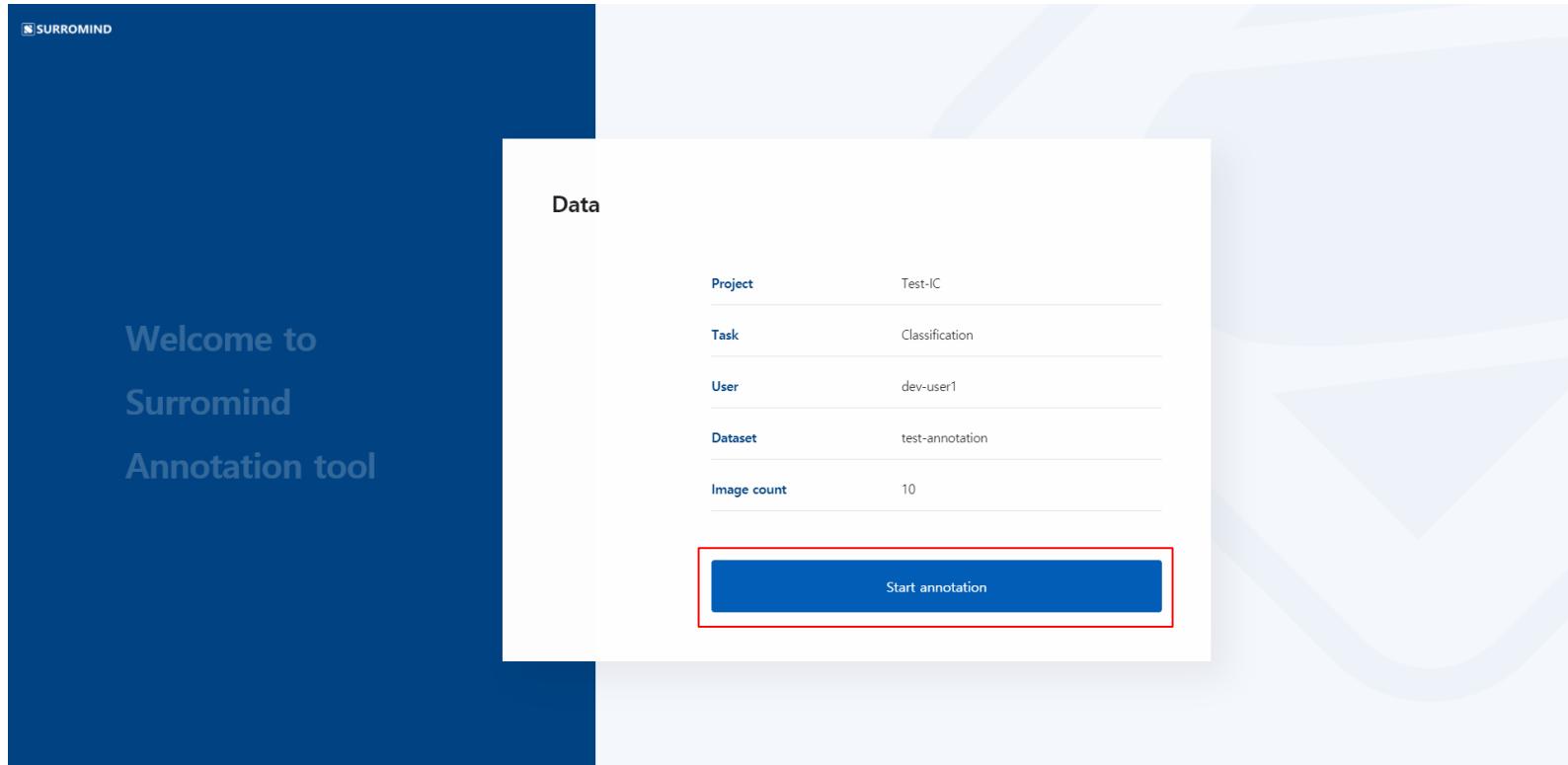
Annotation 메뉴(1)에서 앞 단계에서 Add Label을 추가한 Dataset(2)이 생성되었습니다. More버튼(3)을 클릭하여 Show annotation 버튼(4)을 통해 Annotation 전용 화면으로 이동합니다.

The screenshot shows the SURROMIND Annotation interface. On the left, there is a navigation sidebar with various project and experiment management options. The 'annotation' option under the 'Data' section is highlighted with a red box and labeled (1). The main area displays a table of datasets, with one dataset named 'test-annotation' highlighted by a red box and labeled (2). To the right of the dataset table, there is a 'More' button (3) and a 'Show annotation' button (4), also both highlighted with red boxes. The top right corner of the interface shows project details ('Project name : Test-IC'), template mode settings, and user notifications.

Dataset Name (current)	Project Name	Type	Model	Total	Status	Annotator	Created	Updated
test-annotation	Test-IC	image	Classification	0-9	Working	dev-user1	2021-06-03	(3)
pecotek_0528_hylim_172	test_hylim_0528	image	Classification	0-171	Completed	dev-user1	2021-05-28	(4)
test888	je-instance-segmentation-pytorch	image	Instance Segmentation	0-1	Completed	dev-user1	2021-05-28	2021-05-28
test777	je-instance-segmentation-pytorch	image	Instance Segmentation	0-1	Completed	dev-user1	2021-05-28	2021-05-28
test666	je-instance-segmentation-pytorch	image	Instance Segmentation	0-1	Completed	dev-user1	2021-05-28	2021-05-28
test555	je-instance-segmentation-pytorch	image	Instance Segmentation	2-3	Completed	dev-user1	2021-05-28	2021-05-28
IC-lee052702	Test-IC-lee0527	image	Classification	0-41	Completed	dev-user1	2021-05-27	2021-05-27
OD_LEE072701	Test_OD_lee0527	image	Detection	0-29	Completed	dev-user1	2021-05-27	2021-05-27
test444	je-instance-segmentation-pytorch	image	Instance Segmentation	0-1	Completed	dev-user1	2021-05-24	2021-05-24
test333	je-instance-segmentation-pytorch	image	Instance Segmentation	0-1	Completed	dev-user1	2021-05-24	2021-05-24
test222	je-instance-segmentation-pytorch	image	Instance Segmentation	0-1	Completed	dev-user1	2021-05-24	2021-05-24

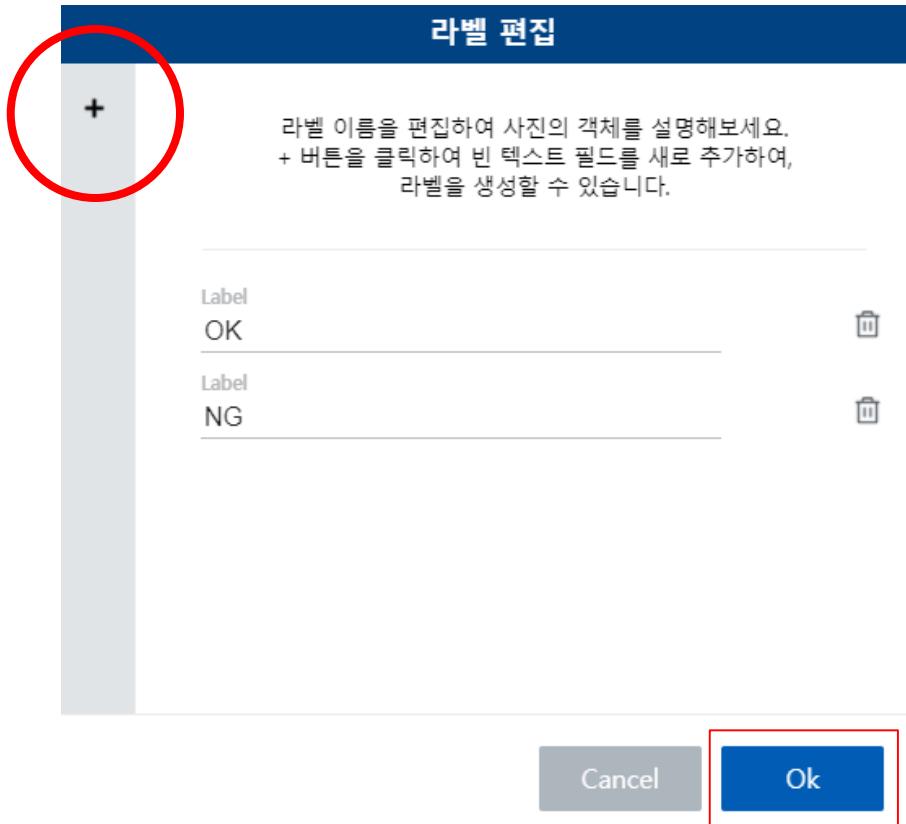
## 6.3 Annotation 시작

Annotation 전용 화면으로 이동후, Start Annotation 버튼을 클릭합니다.



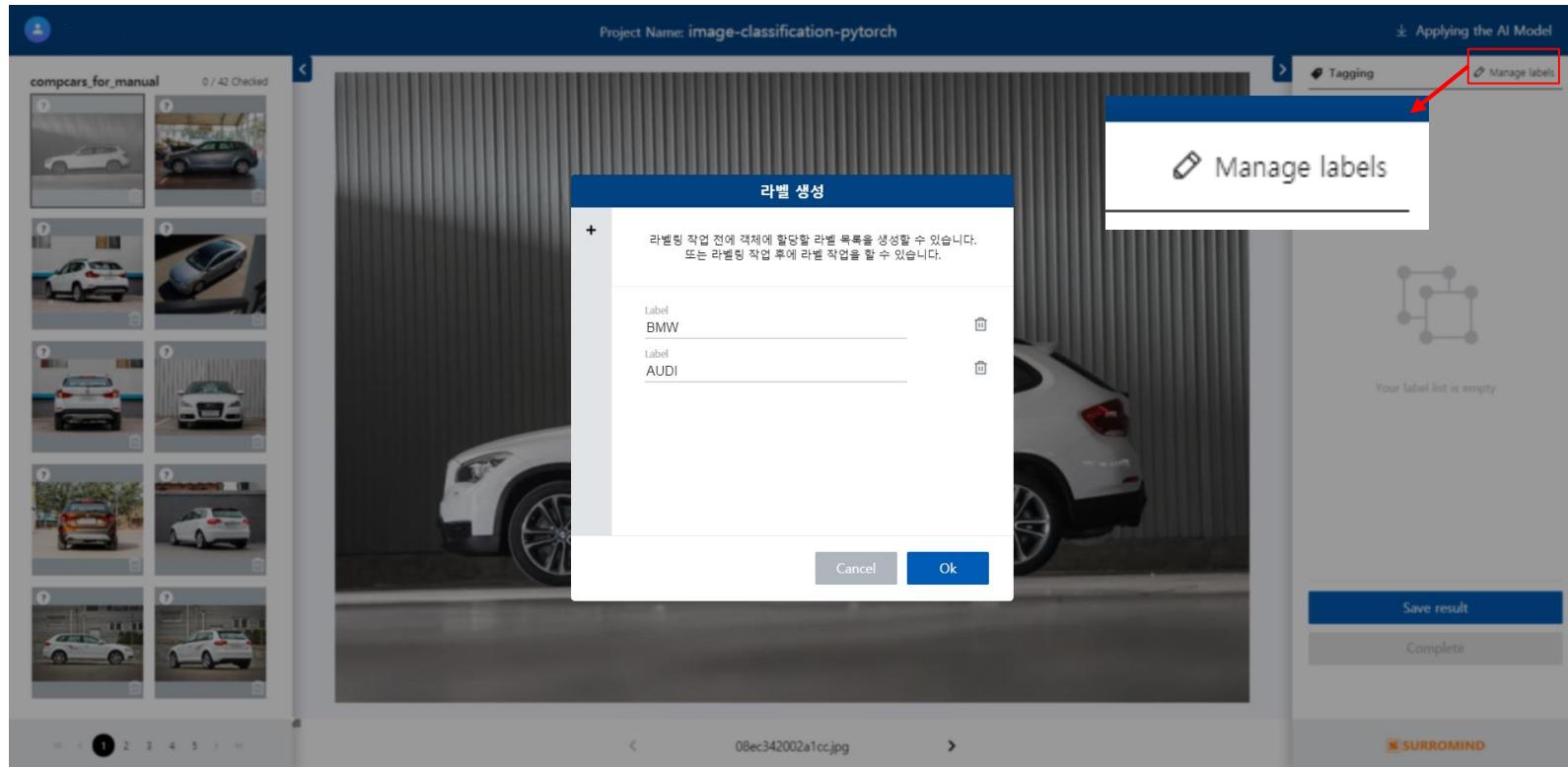
## 6.3 Annotation 시작

Label 편집기의 + 버튼을 눌러 Label을 추가합니다.  
Label을 모두 추가한 후에 Ok 버튼을 클릭합니다.



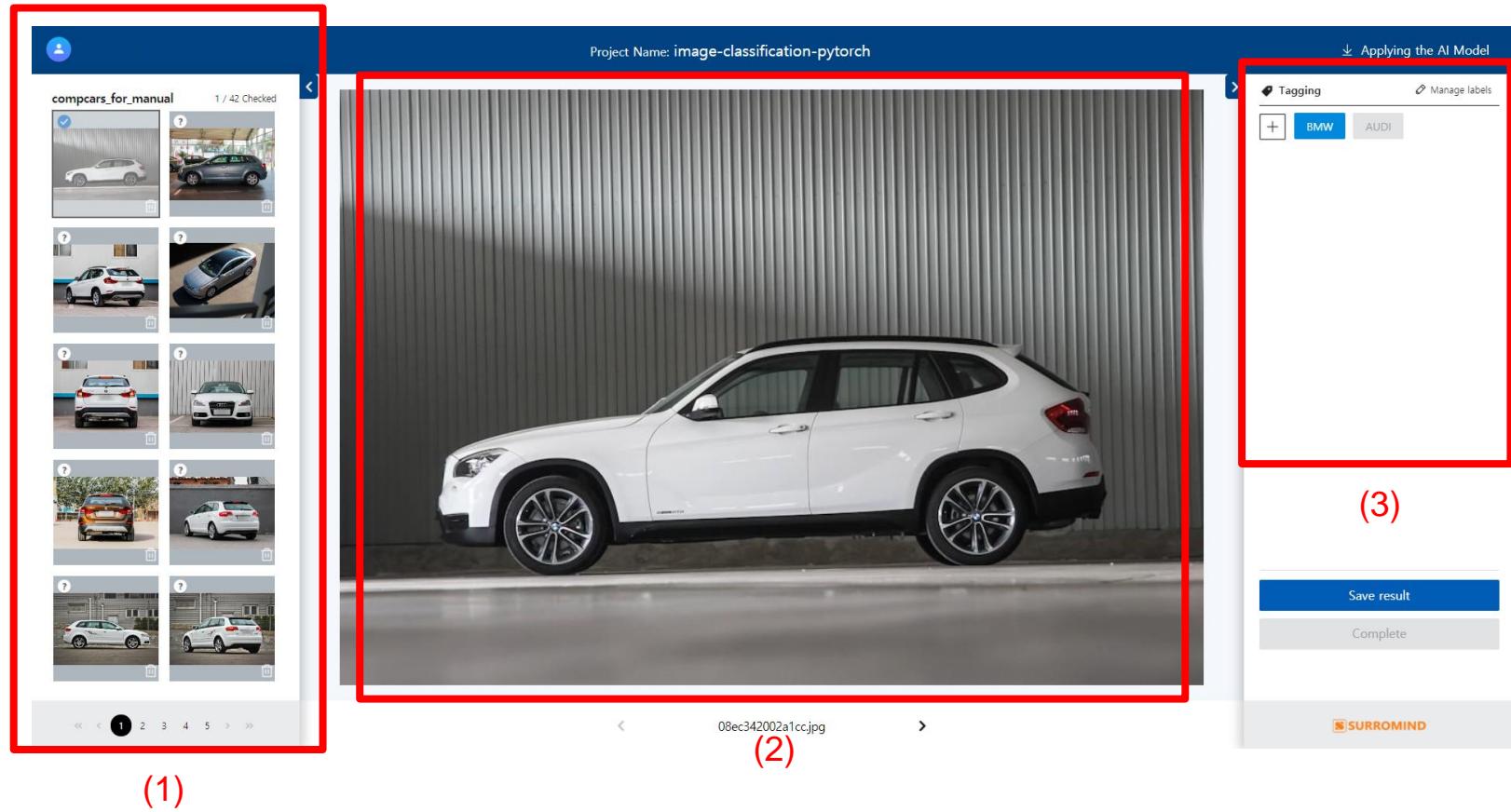
## 6.3 Annotation 시작

작업 중에도 화면 우측 상단에 있는 **Manage labels** 버튼을 클릭하여 Label 편집기를 오픈하여 수정할 수 있습니다.



## 6.4 화면 구성

크게 작업 대상이 될 이미지 영역(1), annotation 작업 영역(2), Tagging 영역(3)으로 구성됩니다.



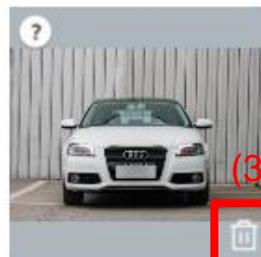
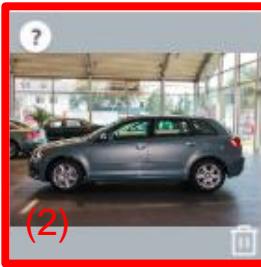
(1)

(2)

(3)

## 6.4 화면 구성

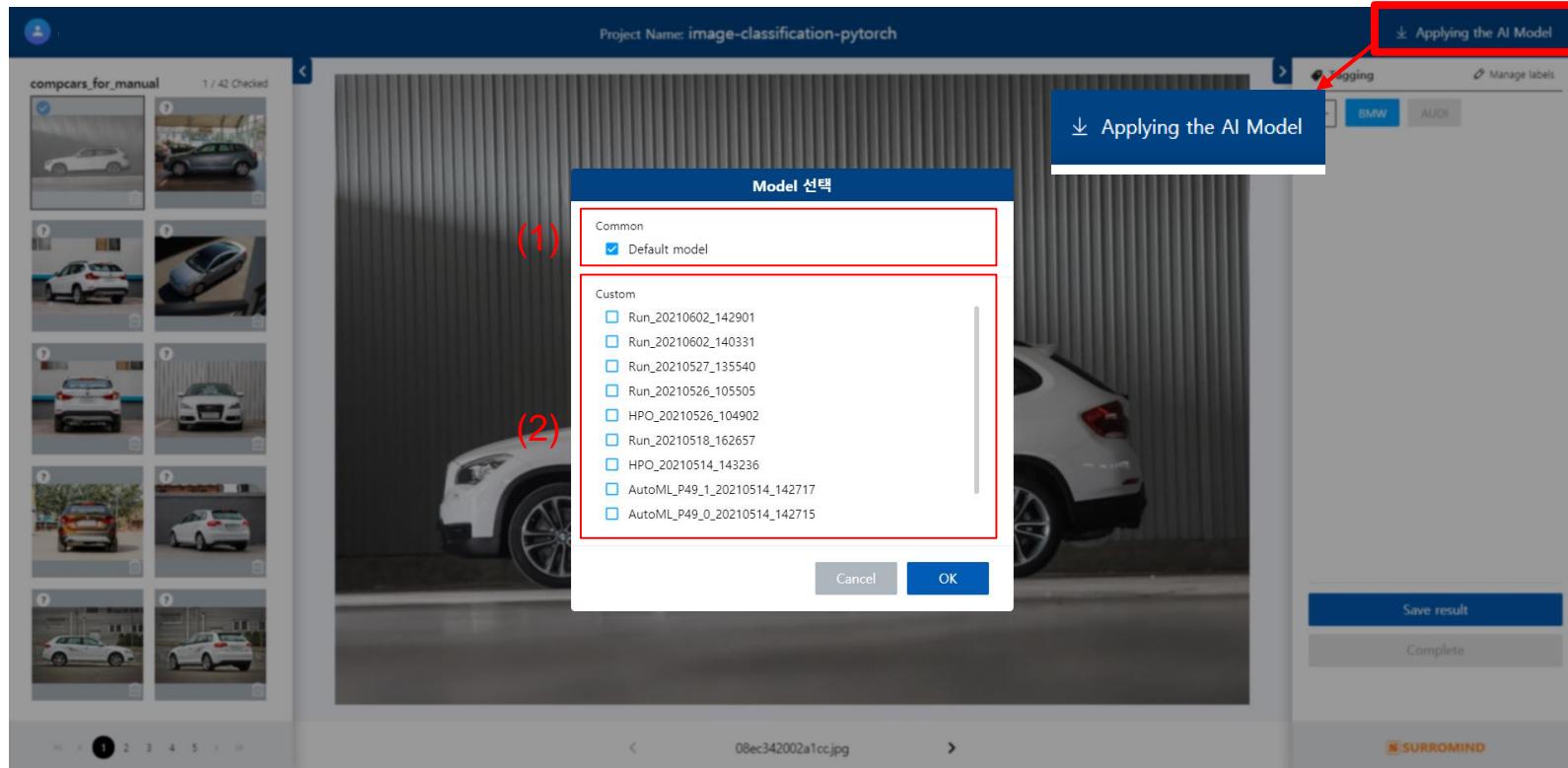
작업이 완료된 이미지(1)와 완료되지 않은 이미지(2)를 구분하실 수 있으며, 작업에서 삭제(3)하실 수 있습니다.



삭제 버튼을 클릭할 경우, 삭제 아이콘이 빨간색으로 표시되며 Complete 처리를 하기 전까진 물리적으로 삭제되지 않습니다.

## 6.5 Auto Annotation

Applying the AI Model 버튼을 클릭하여 Auto Annotation을 진행하실 수 있습니다.  
Default 모델(1)과 Custom 모델(2)을 선택하여 진행하실 수 있습니다. Custom 모델은 직접 학습시킨 모델이 었어야 사용 가능합니다.  
Default 모델은 사용을 권장하지 않습니다.



## 6.6 작업 저장과 완료

작업을 저장하시려면 Save result 버튼(1), 작업을 완료하시려면 Complete 버튼(2)을 클릭해주시기 바랍니다.  
Complete 처리는 개발자 계정만 가능합니다.

Project Name: image-classification-pytorch      ↴ Applying the AI Model

Tagging      Manage labels

+ BMW      AUDI

(1) Save result  
(2) Complete

SURROMIND

compcars\_for\_manual      1 / 42 Checked

08ec342002a1cc.jpg

## 6.6 작업 저장과 완료

Complete 처리된 Annotation 일감은 Status가 ‘Completed’로 변경(1)되며, Data 화면에서도 Label이 자동으로 추가됩니다.

Annotation								
Data								
Dataset Name(current)		검색어를 입력해주세요.		나에게 할당된 일감 보기		Days	Weeks	Months
DATASET NAME	PROJECT NAME	TYPE	MODEL	TOTAL	STATUS	ANNOTATOR	CREATED	UPDATED
test222	je-instance-segmentation-pytorch	image	Instance Segmentation	0-1	● Completed	dev-user1	2021-05-24	2021-05-24
222	test	image	Classification	2-3	● Working	dev-user1	2021-05-24	-
carvana_10_data	je-instance-segmentation-test	image	Instance Segmentation	5-10	● Completed	dev-user1	2021-05-21	2021-05-24

(1)

Data						
Private Shared						
Dataset Name(current)		검색어를 입력해주세요.		Days	Weeks	Months
□	DATASET NAME	VERSION	DATA TYPE	SIZE	CREATED	LABEL
□	car-dataset	1	image	100.7015 MB	2021-05-25	(2) for_train.zip

(2)

## 6.7 Image Classification

Image Classification에 적합한 Dataset은 아래와 같습니다

조건	예시	특징
검출할 불량이 다양하지 않음	A blue circle labeled "양품" (Good Product) and a blue triangle labeled "불량" (Defect).	양품과 불량만 구별하면 됨
양품과 불량의 형상 차이가 큼	A blue circle labeled "양품", a red circle labeled "불량1", and a blue triangle labeled "불량2".	

## 6.7 Image Classification

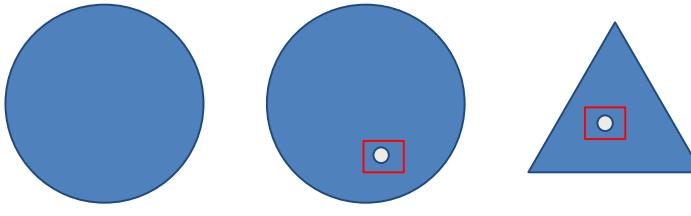
좌측에 있는 이미지(1)를 하나씩 확인 한 후 미리 지정한 Label을 선택(2)해줍니다.

The screenshot shows a user interface for image classification. On the left, a grid of small car images is displayed, with one image in the top-left corner having a checkmark and a red border around the entire grid, labeled '(1)'. In the center, a large image of a white BMW SUV is shown from a side-front angle against a dark background. On the right, there is a 'Tagging' panel with a '+' button, a blue 'BMW' button, and a grey 'AUDI' button, also with a red border and labeled '(2)'. The top bar indicates the project name is 'image-classification-pytorch'. At the bottom, there are buttons for 'Save result' and 'Complete', and a 'SURROMIND' logo.

(1)

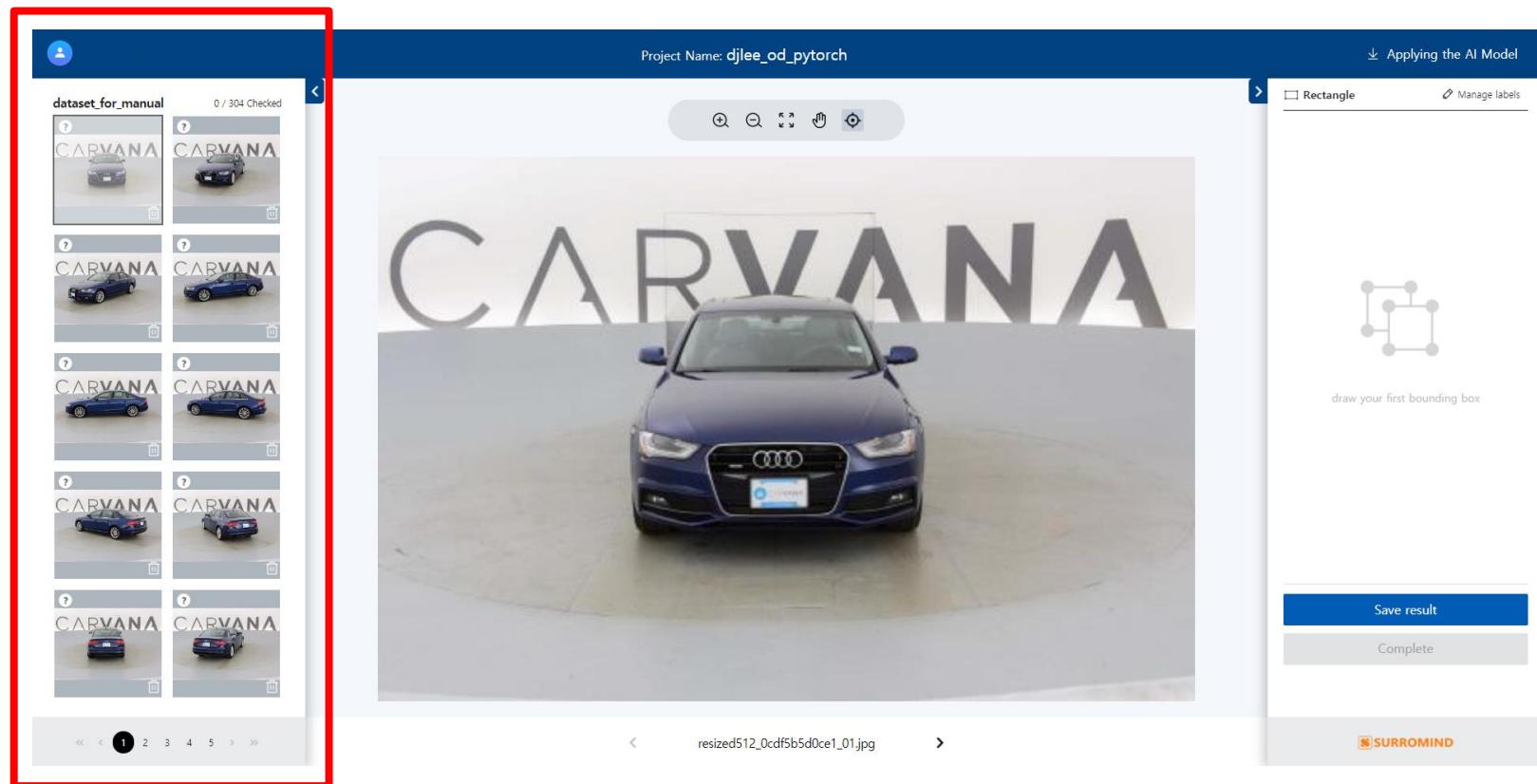
## 6.8 Object Detection

Object Detection에 적합한 Dataset은 아래와 같습니다.

조건	예시	특징
복잡한 환경 속에서 특정 패턴을 찾아야 할 경우 (예: 물류 라인에서 상자 내 바코 드 영역 검출, OCR 검출, dent 검출)	 <p>양품                      불량1                      불량2</p>	검출할 대상은 패턴이 일정하지만 그 외의 환경 이 복잡하고 다양함

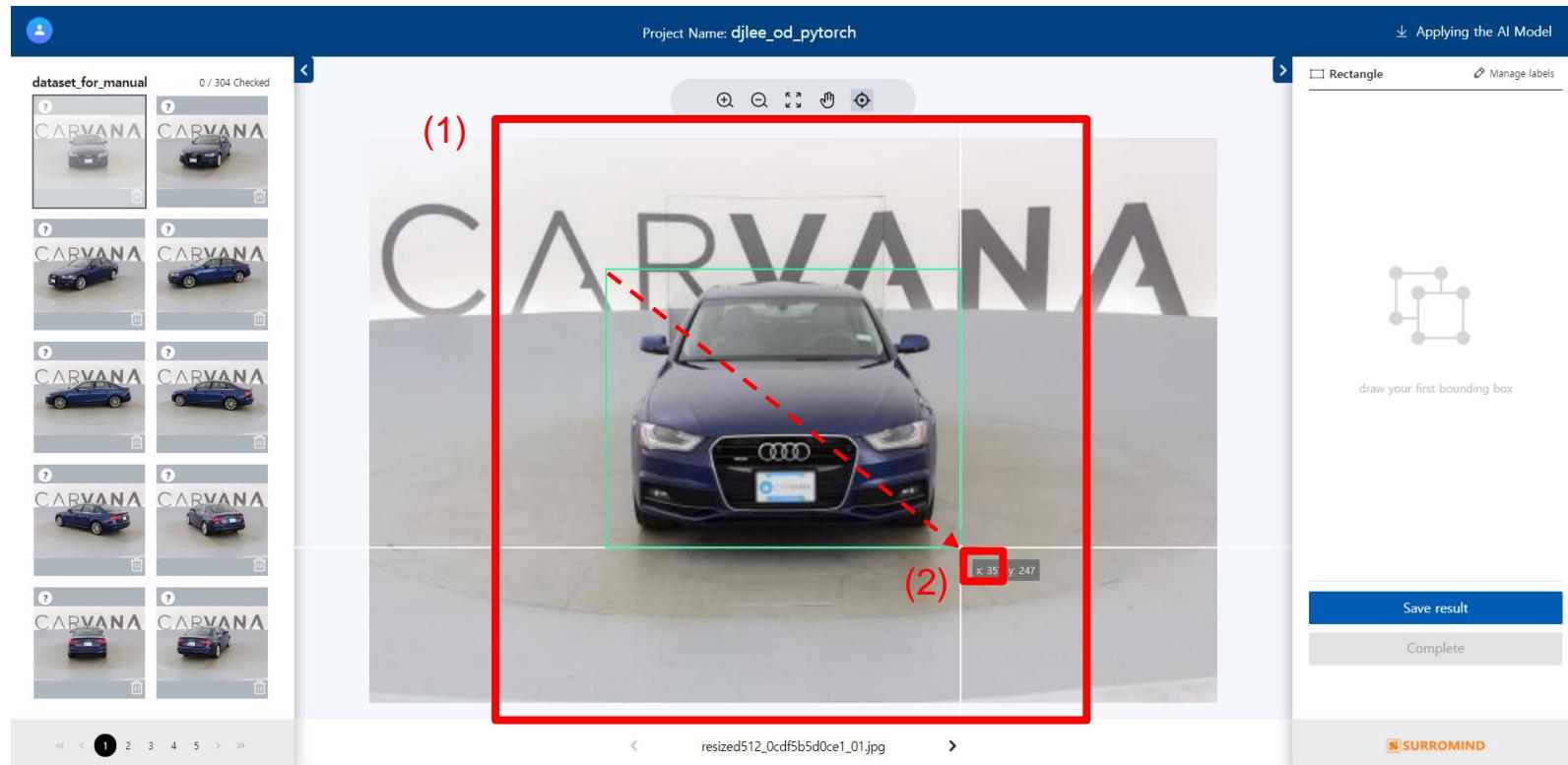
## 6.8 Object Detection

좌측에 있는 이미지를 하나씩 확인한 후 작업을 진행합니다.



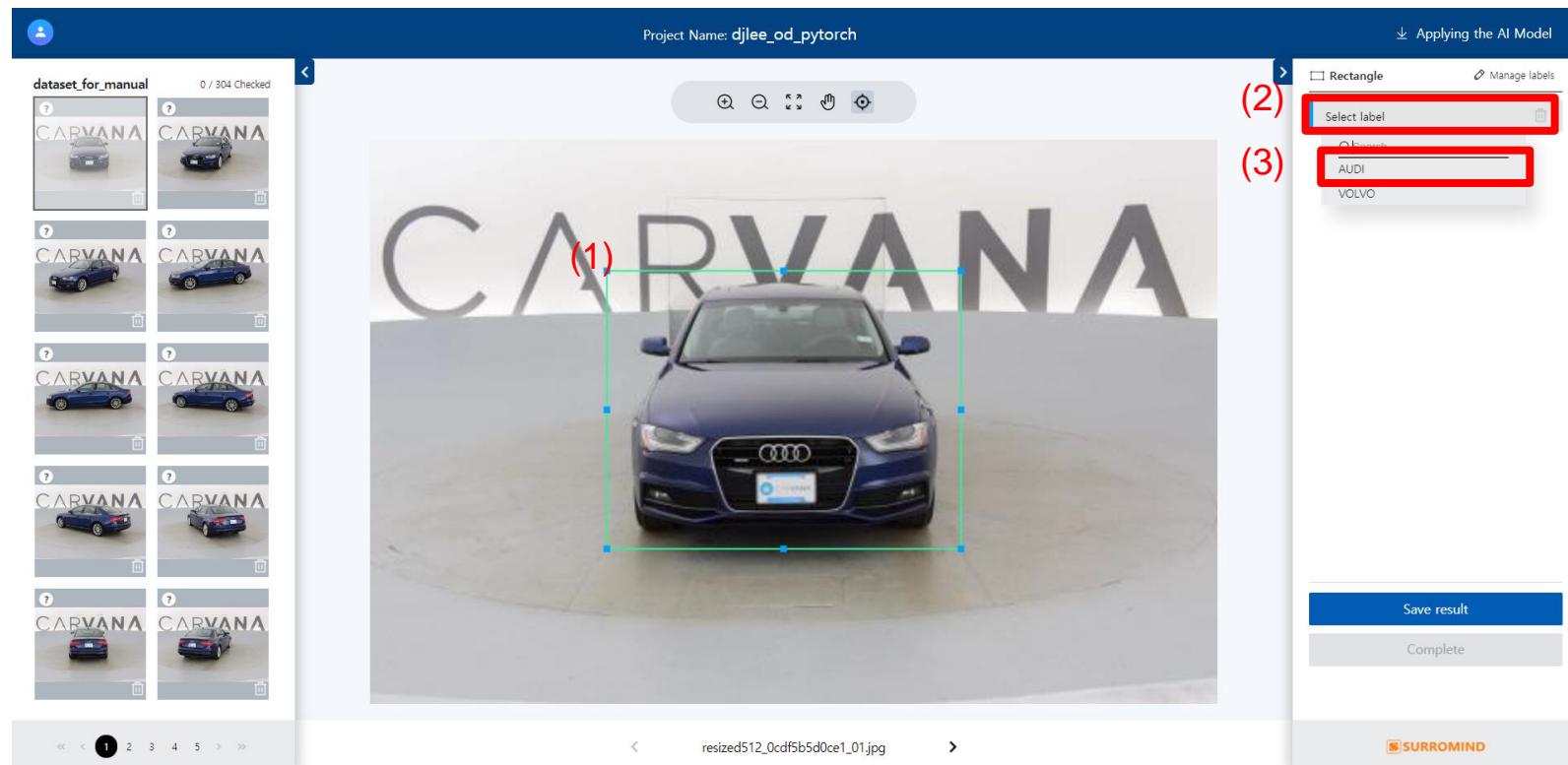
## 6.8 Object Detection

Image 영역(1) 내부에서 왼쪽 마우스 버튼을 누른 채 원하는 크기만큼 사선으로 Drag(2) 하여 Bounding Box를 생성 합니다.



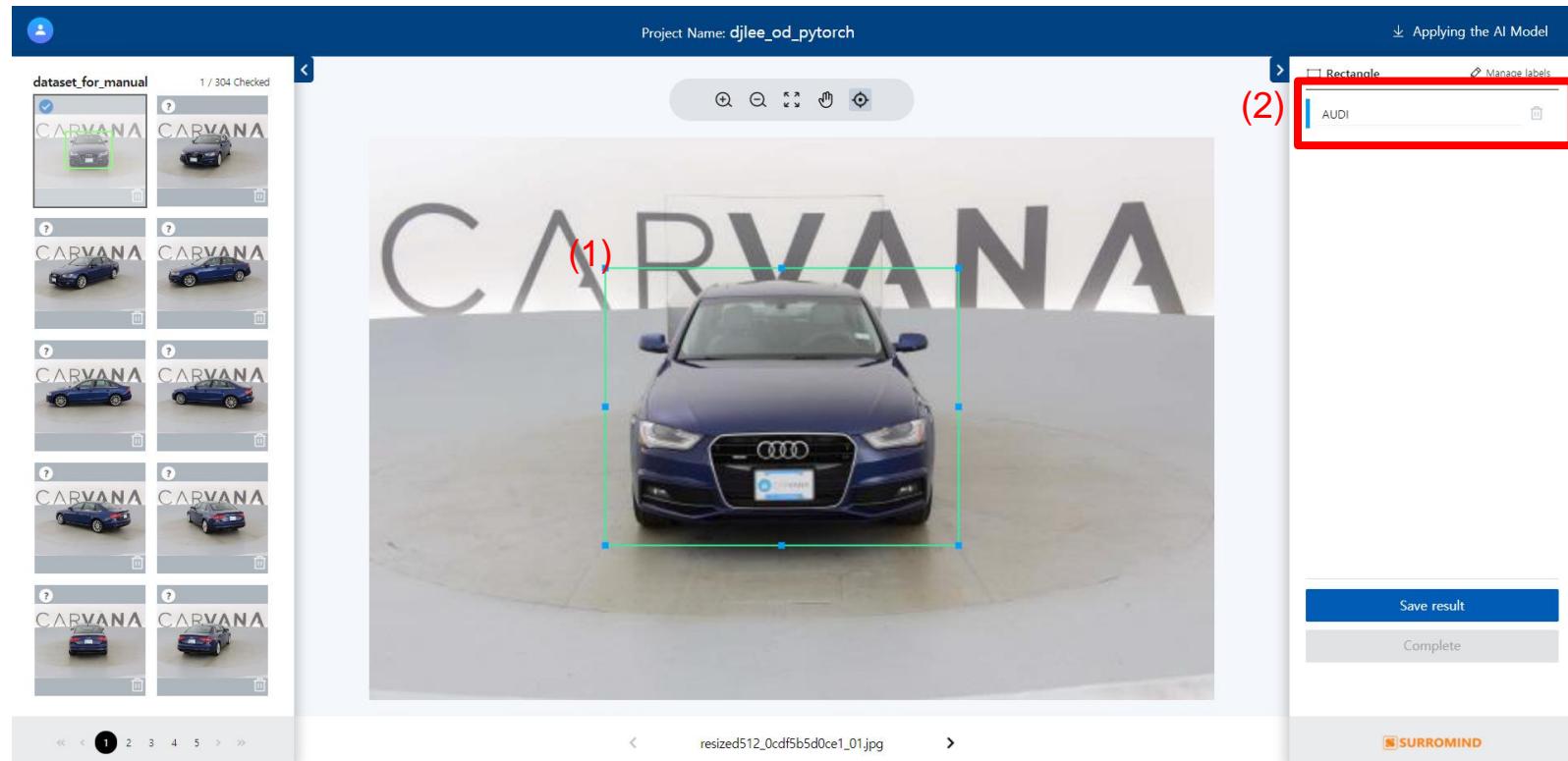
## 6.8 Object Detection

Bounding Box(1)를 선택한 후 화면 우측 Select label(2)을 클릭, 사전에 등록 한 Label(3)을 선택하여 label을 지정합니다.



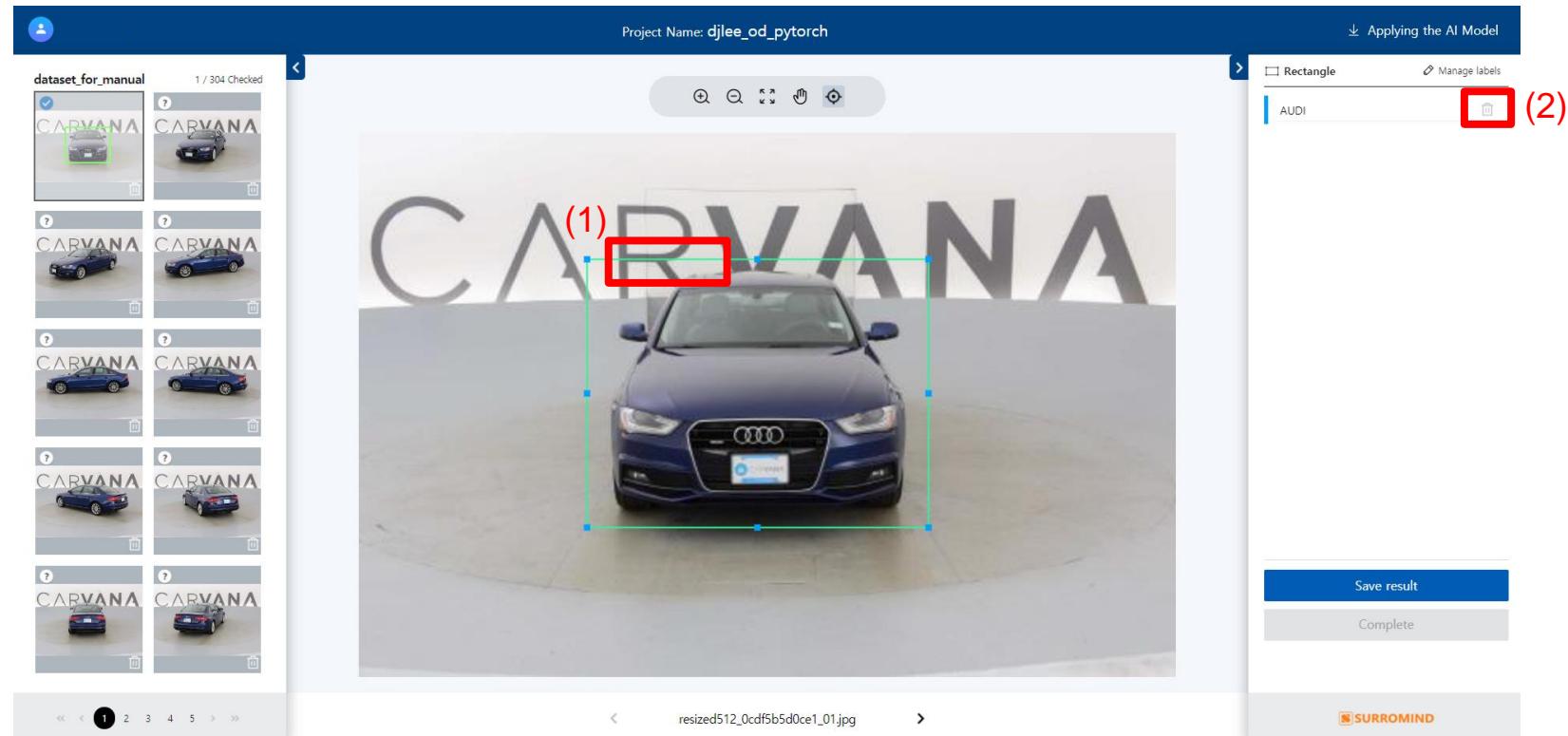
## 6.8 Object Detection

Bounding Box(1) 선택시 대응 되는 Label(2) 이 활성화 되고, Label(2) 선택시 대응 되는 Bounding Box(1)가 활성화 됩니다.



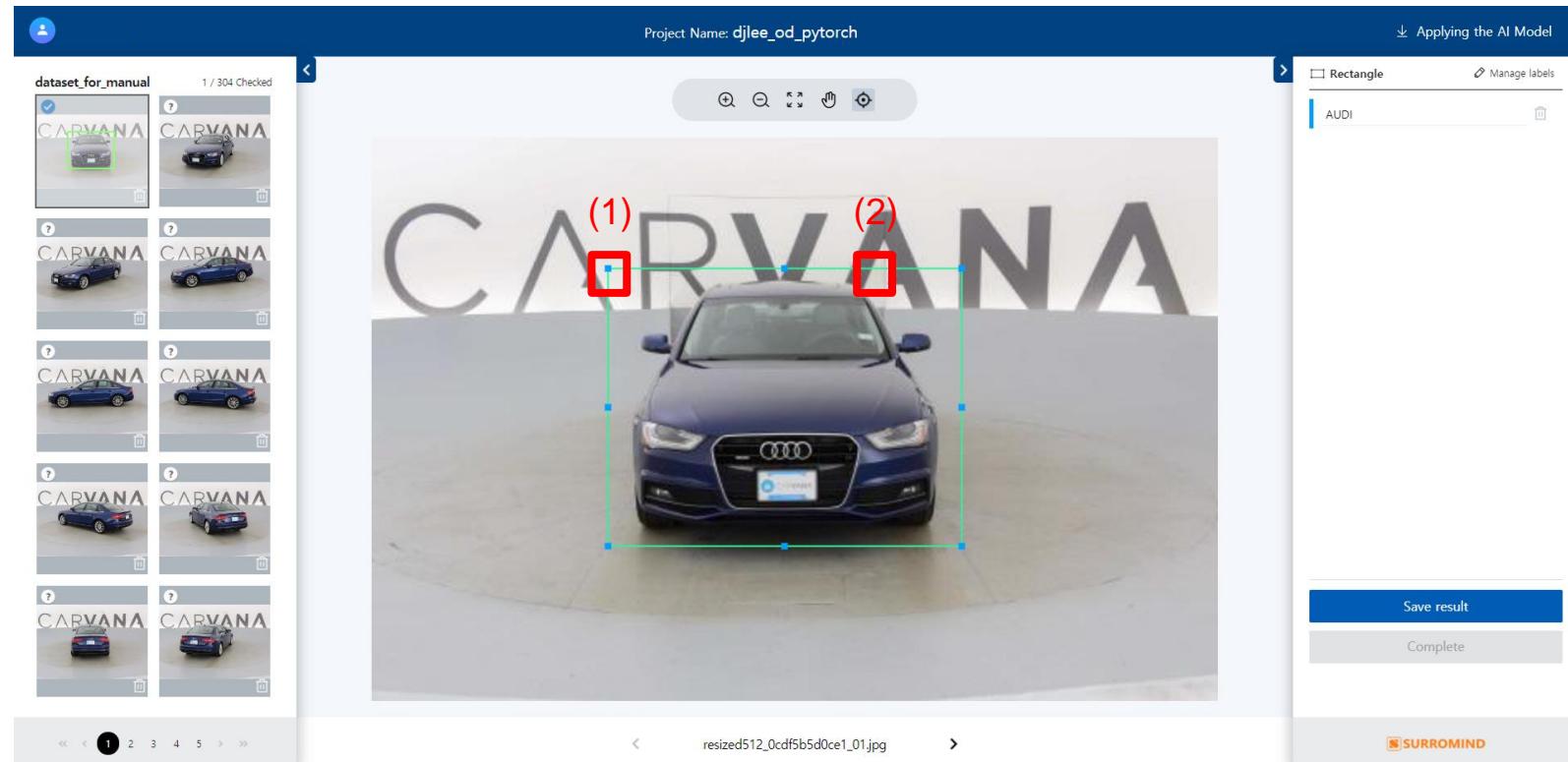
## 6.8 Object Detection

Bounding Box(1)의 경계를 선택 후 키보드의 Delete 키를 누르거나 Delete버튼(2)을 클릭하여, Bounding Box와 Label을 함께 삭제합니다.



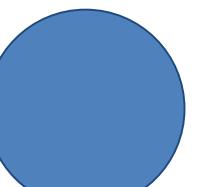
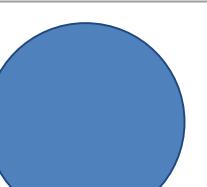
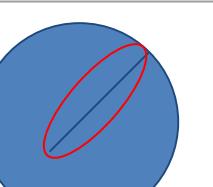
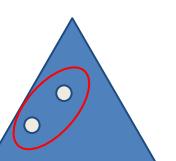
## 6.8 Object Detection

Bounding Box의 경계에서 파란색 점(1) 클릭 후 Drag시 크기 조절, 초록색 선(2) 클릭 후 Drag시 Bounding Box의 위치 이동이 가능합니다.



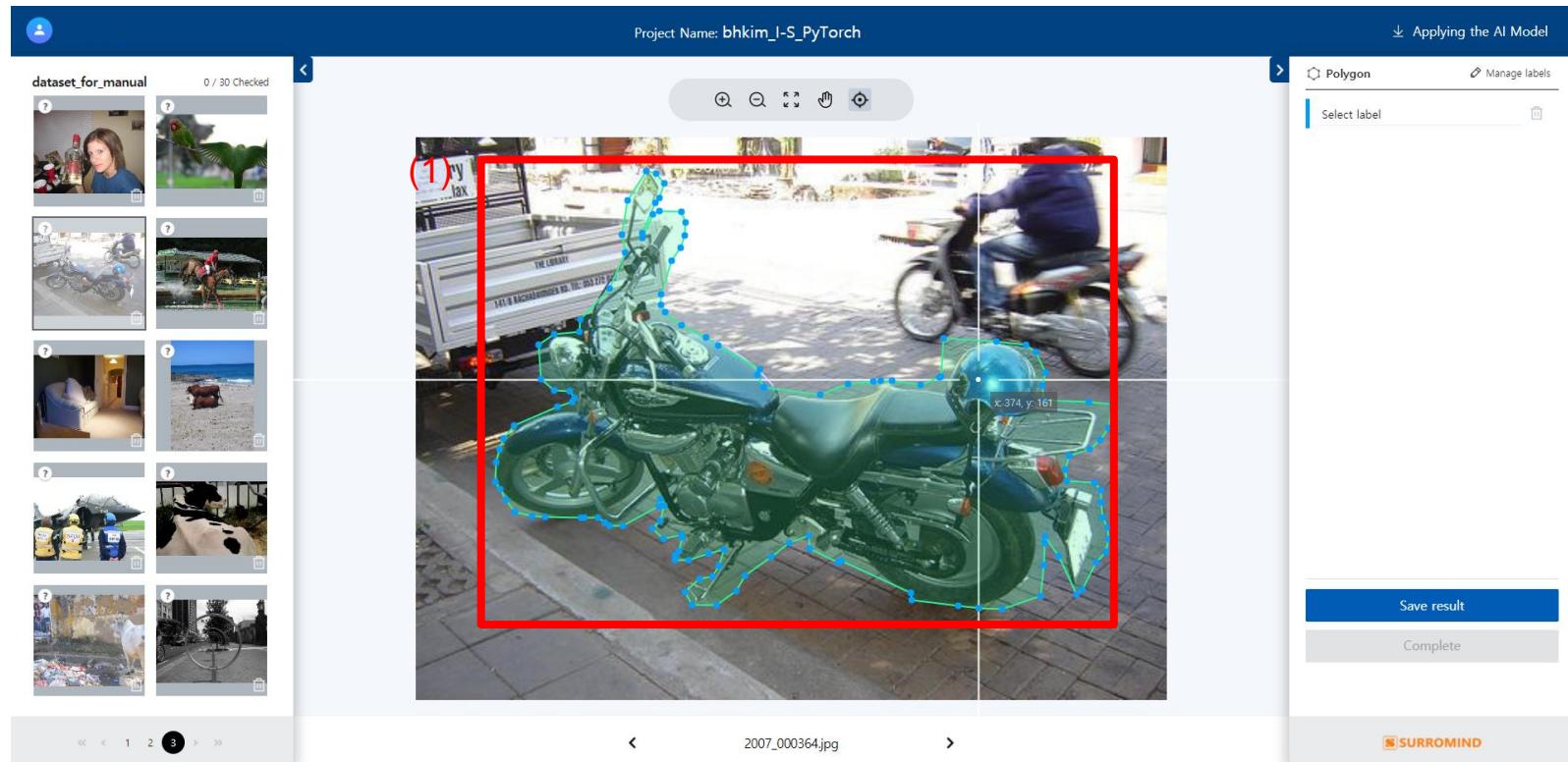
## 6.9 Instance Segmentation

Instance Segmentation에 적합한 Dataset은 아래와 같습니다.

조건	예시	특징
양품 형상이 다양	 	형상이 다르지만 양품으로 판정해야 함
불량 유형이 다양	  	불량 유형이 다양하여 구분해주어야 함

## 6.9 Instance Segmentation

마우스를 이용해, Label을 부여하고자 하는 영역의 경계점(1)을 표시합니다(경계점은 왼쪽 클릭으로 추가).



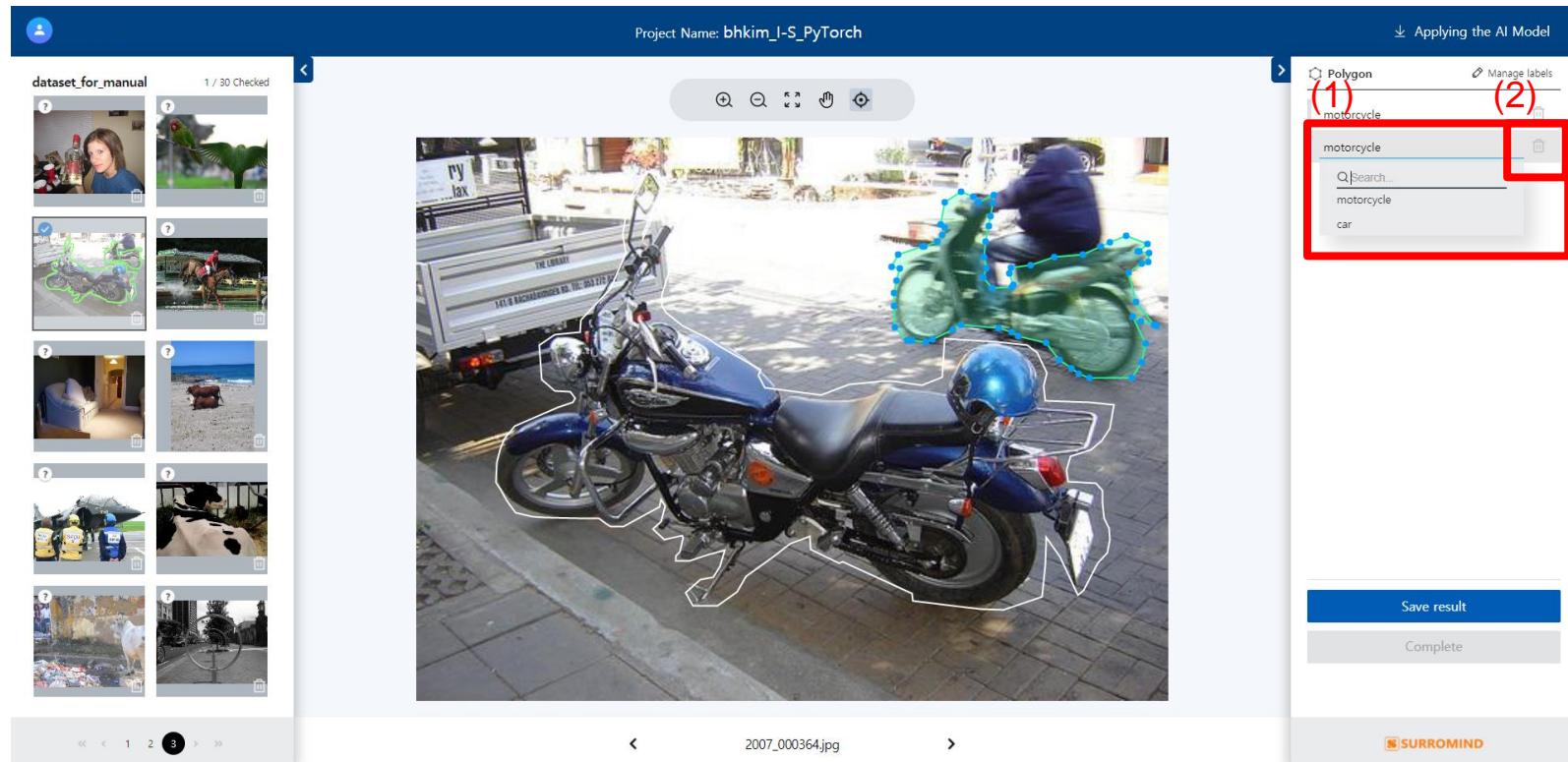
## 6.9 Instance Segmentation

시작점과 끝점이 맞닿으면, 영역(1)이 표시됩니다.  
화면 우측에 있는 Polygon 목록에서 지정한 영역(2)을 항목으로 확인할 수 있습니다.



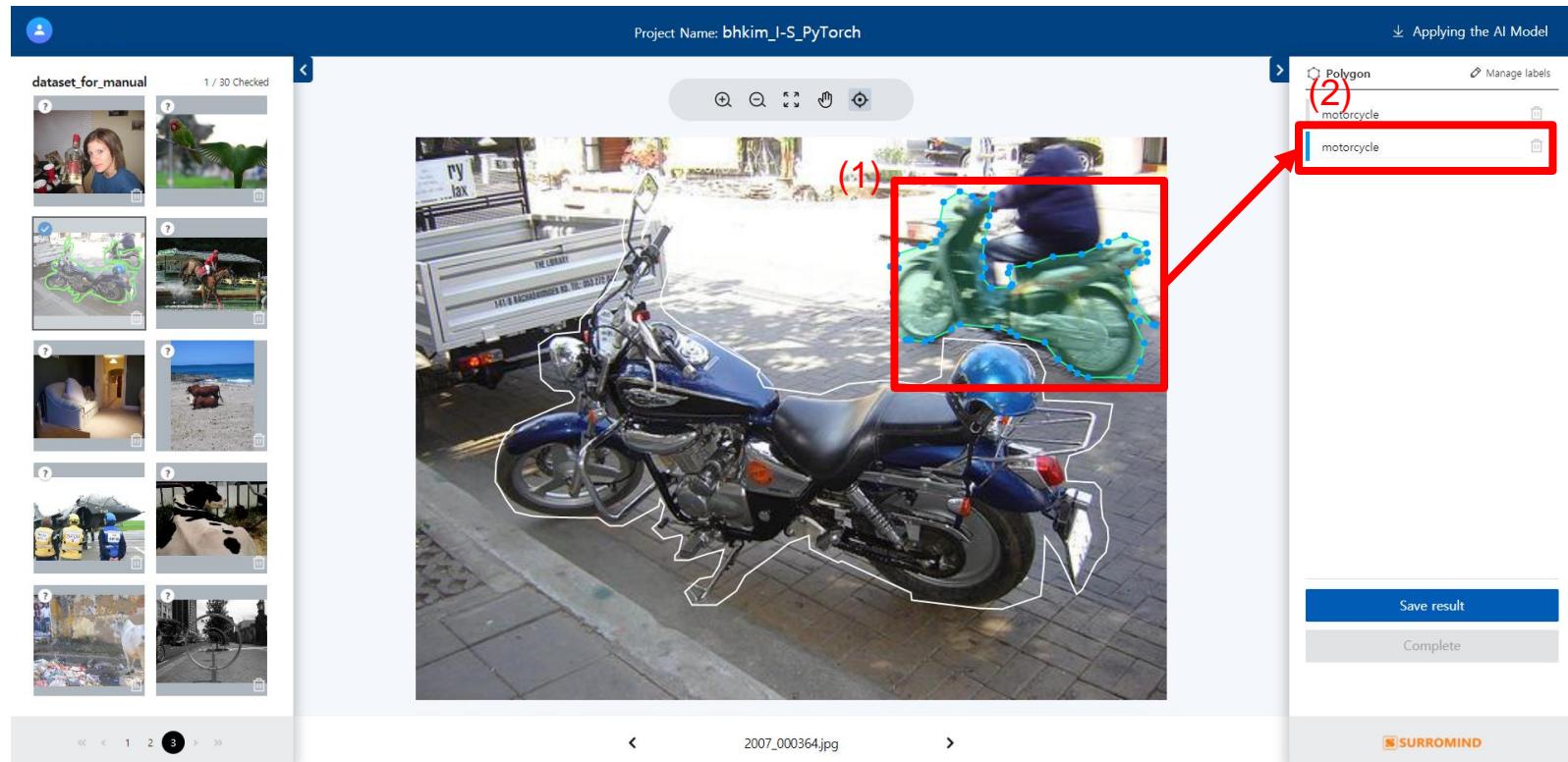
## 6.9 Instance Segmentation

우측 Polygon 목록에서 영역에 해당하는 항목(1)을 클릭하면, 그림과 같이 Label을 지정할 수 있습니다.  
Label이 지정된 항목을 클릭하면 Label을 다시 지정할 수 있고, 항목 옆의 휴지통 버튼(2)을 누르면 영역이 삭제됩니다.



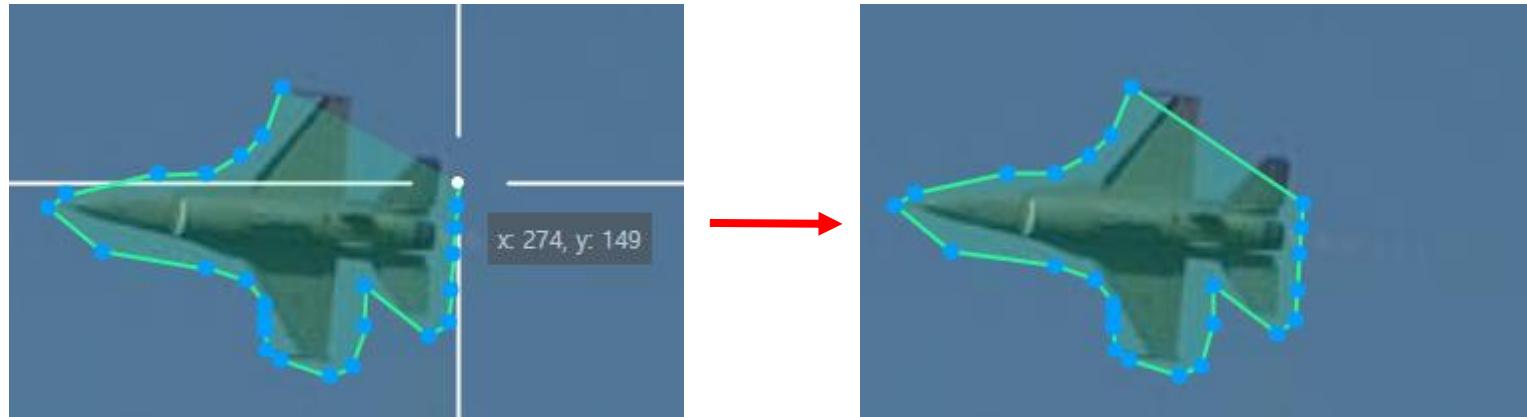
## 6.9 Instance Segmentation

오른쪽 폴리곤 목록에서 영역이 지정된 항목(1)을 클릭하면, 이미지에서 항목에 해당하는 영역(2)이 활성화됩니다.

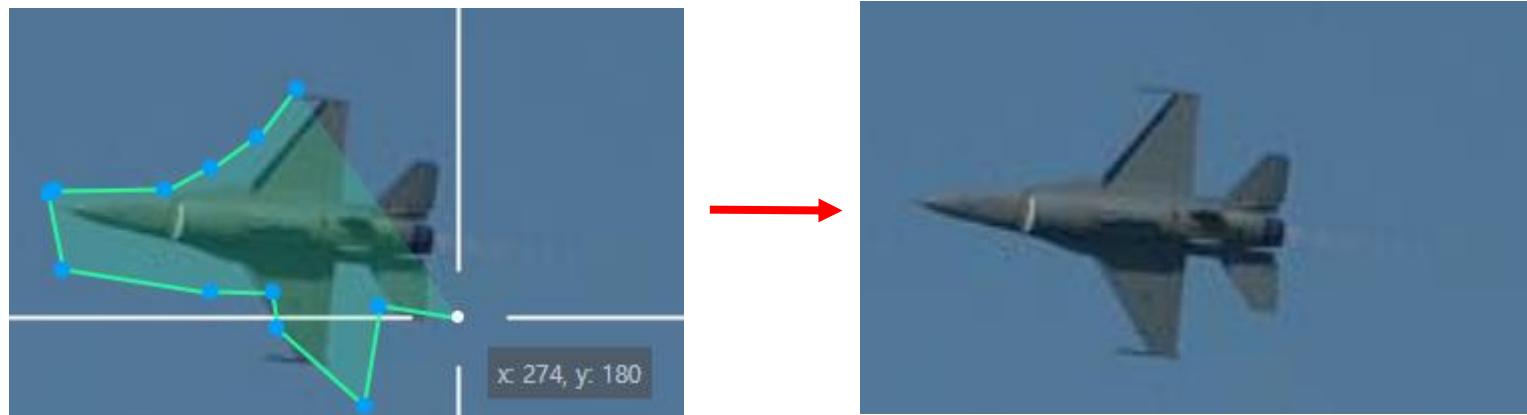


## 6.9 Instance Segmentation

경계점을 표시하는 도중 Enter키를 입력하면 경계점의 시작점과 끝점이 자동으로 연결됩니다.



경계점을 표시하는 도중 ESC키를 누르면, 표시하던 영역이 사라집니다.



## 6.9 Instance Segmentation

경계점 표시가 완료된 영역의 경계점을 드래그하면 경계점을 변경할 수 있습니다.



## 6.9 Instance Segmentation

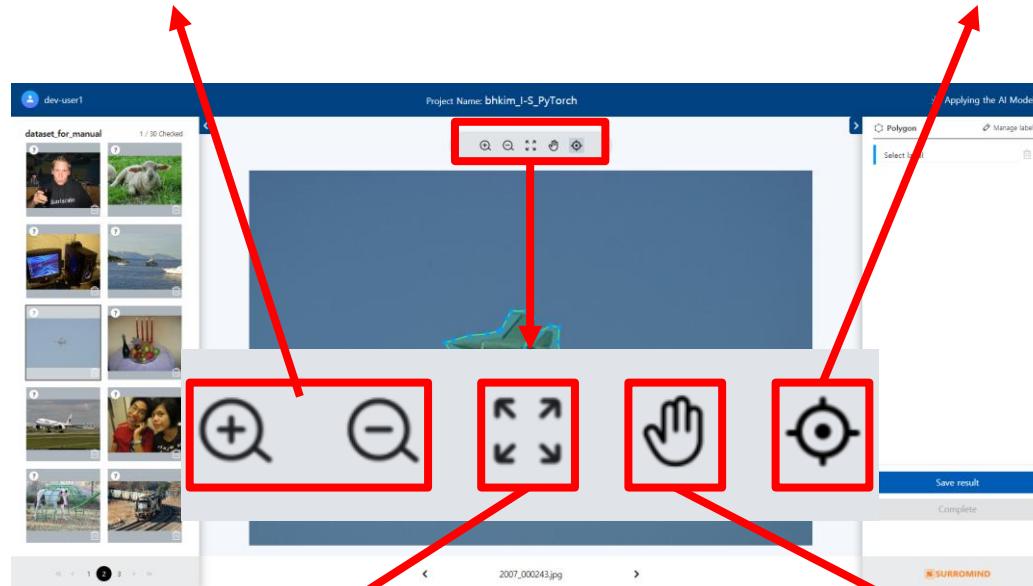
경계점과 경계점 사이에 새로운 경계점을 만들 수 있습니다.



## 6.9 Instance Segmentation

이미지의 배율을 조절할 수 있습니다.

포인터 주변 십자선 표시 여부를 설정할 수 있습니다.



배율을 초기화합니다.

활성화하면 마우스 드래그를 통해 이미지를 탐색합니다.