

A close-up photograph of green strawberry leaves with water droplets, overlaid with the word "LEAF" in large white letters. The background is a dense field of green leaves, some in focus and some blurred. Numerous small, bright water droplets are scattered across the leaves and in the air, catching the light. The word "LEAF" is written in a large, white, sans-serif font, with the letters "L" and "E" on the top line and "A" and "F" on the bottom line. The text is positioned on the right side of the image, partially overlapping the leaves.

LEAF

LEAF



LEAF

Live Efficiently All Farmers



Sustainable
지속가능한 농업

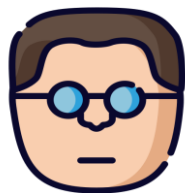


Ultra more produce
매우 더 많은 생산성



Consumer Friendly
소비자 친화적인

LEAF



홍성민

Visualizing, Concept, Vision CNN



LE
AF

background



population

현재의 인구는 약 80억명
농업 생산성이 최소 60% 증가
하지 않으면 전 세계적으로
2050년까지 예상되는 세계 인구
에 제공할 식량
부족



climate

지구 온난화, 해수온도 상승, 이상기후
지구에 농작이 가능한 지역 감소

갑작스러운 기후 변화에 따른 흉작으로
일부 지역의 식량난 발생

농부들의 노동력 대비 생산량 감소



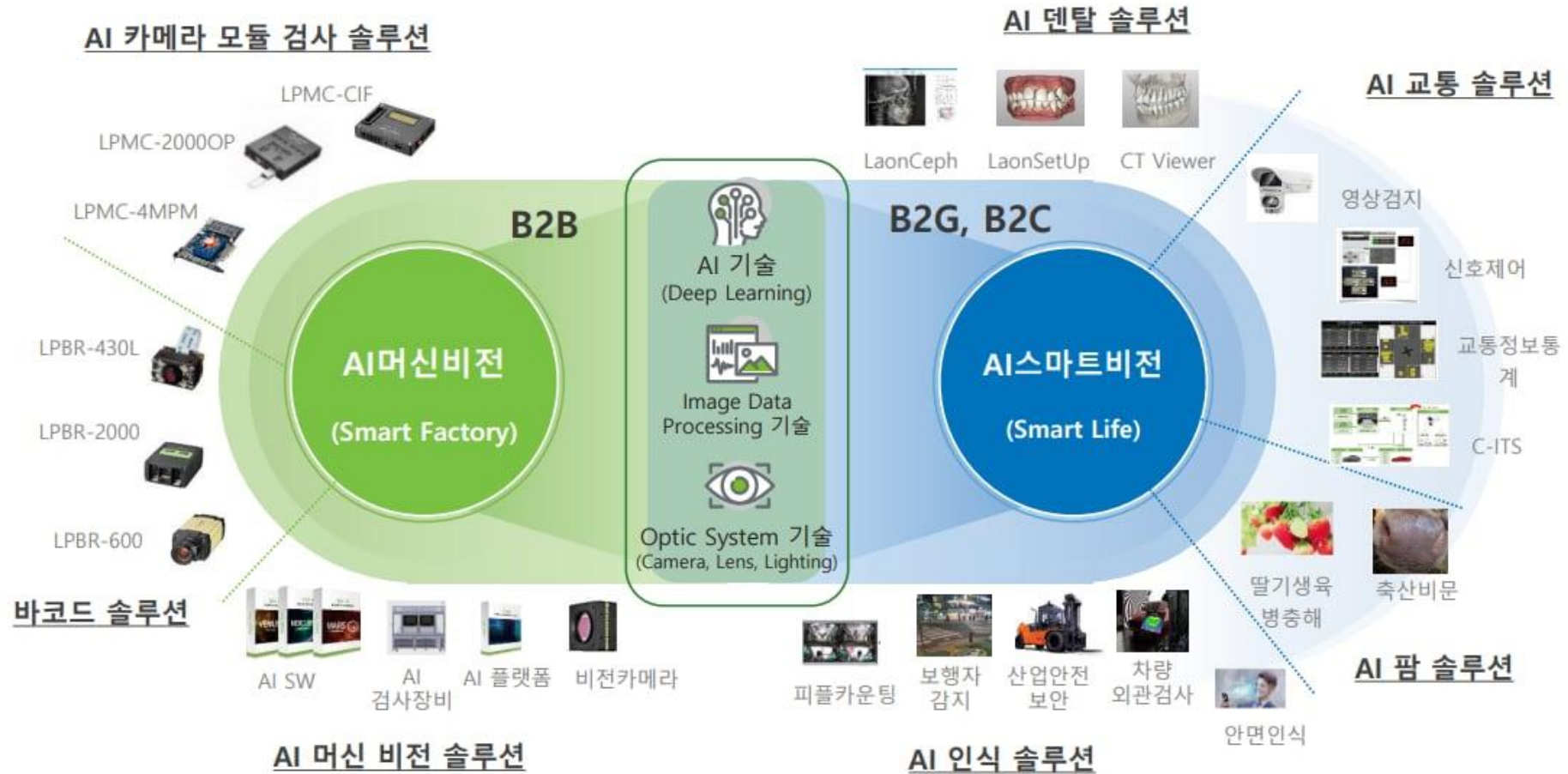
pollution

플라스틱 쓰레기, 핵폐기물의 해양 오염
인구증가, 산업 발전에 의해
지구의 환경 오염 증가 및 독성 물질
잔류

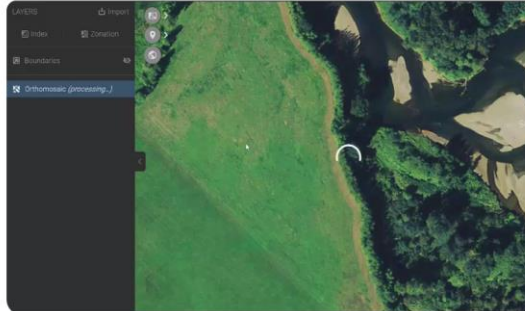
생물 증폭에 따른 오염물 섭취

온·습도만 재던 스마트팜, AI로 업그레이드

농작물이 잘 자랄 수 있는 기본 조건은 토양과 기후다. 온도와 습도가 조금만 변해도 수확물이 달라진다. 이에 농장에 센서를 달아 온도와 습도를 체크하고, 농작물의 생육 환경을 적정 수준으로 유지하는 스마트팜이 생겨났다. 스마트팜은 환경 변화를 감지해 관수밸브를 자동 혹은 원격으로 조절하거나, 온도를 조정하는 식이다.

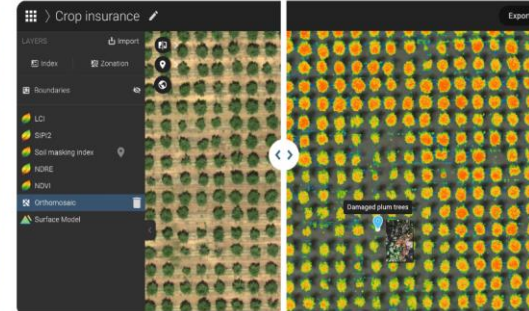


Analyze field data in real time



Create maps instantly

Process images with PIX4Dfields' instant processing engine and get results in minutes, whether in the office or in the field. No Internet connection required



Compare progress over time

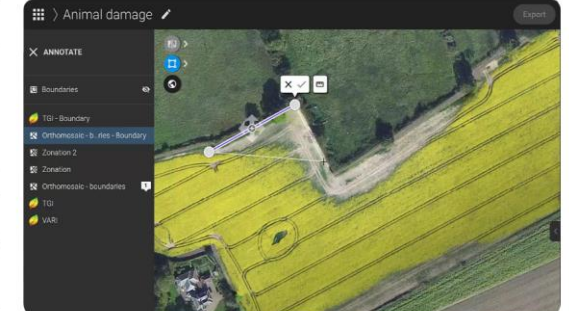
Link different layers of information and compare your crop trends over time on a split or dual display

Explore your farm and target your inputs



Zonate and prescribe

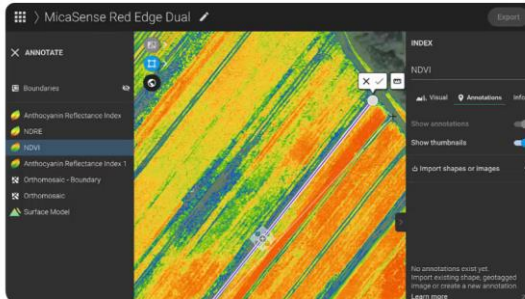
Plan fertilizer and phytosanitary product applications according to plant needs.



Measure and annotate

Easily measure an area or several areas at once. Add annotations or import GPS-tagged images as geolocated annotations directly into the layer.

Boost your productivity and team collaboration



Manage your trial plots

Calculate the mean and standard deviation of DSM and index layers, as well as area annotations.



Share and collaborate

Share your maps with all project stakeholders and the team for seamless collaboration using the PIX4Dfields export tool.

Example _fix4D fields

- 드론과 GPS를 이용한 농작물의 생장 상태 확인
- 구역화와 농작물 개선을 위한 처방
- 실제 면적 측정
- 대규모의 농장을 관리하는데 좋은 솔루션

Iron Ox uses AI and robots to grow 30 times more produce than traditional farms

아이언 옥스는 인공지능과 로봇을 사용하여 기존 농업방식 보다 약 30배의 수확을 거두었다



미국 내 200만 개가 넘는 농장에서는 파종, 파종, 검사 등의 작업을 수행하기 위해 약 925,000명의 인력을 고용하여 2017년 총 생산 비용이 3,500억 달러에 달했습니다. 하지만 농업 생산성이 최소 60% 증가하지 않으면 전 세계적으로 2050년까지 예상되는 세계 인구에 비해 충분히 성장하지 못할 수도 있습니다.

다가오는 위기는 3년 된 스타트업 Iron Ox의 공동 창업자들에게 미래를 위한 실내 농장을 설계하는 데 원예 지식과 로봇 공학 데이터베이스를 활용하도록 영감을 주었습니다.

Iron Ox는 농업에 자동화를 적용하는 유일한 회사가 아닙니다.

뉴저지의 AeroFarms는 스타트업 Bowery 및 Plenty와 마찬가지로 지능형 기계에 크게 의존하고 있지만 성장 프로세스를 처음부터 끝까지 완전히 자동화한 최초의 회사라고 주장합니다.

출처 : <https://venturebeat.com/>



We are on a bold mission to grow the best plants possible for the betterment of humanity.

Aerofarms는 수직 농장을 추구하고 있으며 빛과 공기가 차단된 상태에서 모든 환경을 제어하는 Smartfarming



AEROFARMS micro cookbook

자사의 농작물로 만든 건강한 음식에 대한 레시피를 책으로 만들어 제공하고 있으며, 소셜미디어 활동도 활발히 진행

출처 : AeroFarms 홈페이지
<https://www.aerofarms.com/>

Wherever food is needed, we can grow it.

수직 농장은 거의 모든 곳에 건설될 수 있습니다.
그로 인하여 우리가 서비스를 제공하는 시장 근처에
배치함으로써 공급망을 근본적으로 단축하여 장거리
운송으로 인한 혼란과 환경 비용을 줄이고 전반적인
식품 시스템의 탄력성을 높일 수 있습니다

A new agricultural revolution has started. Congress must invest now to secure the future of U.S. food



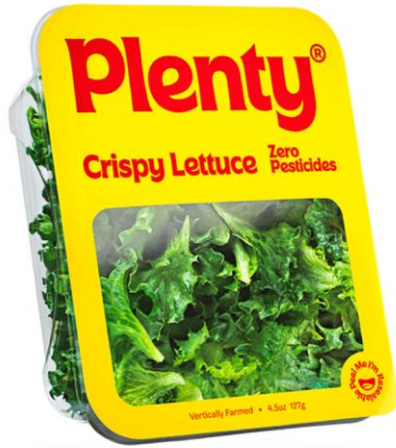
We need to grow more food
over the next 30 years than we've
needed for the last 10,000 years

Bowery cut our energy footprint, reduced water use, and updated packaging in the last year.

작물이 바닥에서 천장까지 쌓여 있는 대규모 스마트 실내 환경에서 신선하고 지역적이며 무농약으로 성장할 수 있습니다.
재생 가능 에너지로 구동되는 이 접근 방식은 토지와 물을 훨씬 적게 사용합니다. 변화하는 기후 조건이나 악천후에 관계없이 작동하는 지속 가능한 모델
입니다.

많은 미국인들은 아마도 수직농장 농산물이 얼마나 다르게 재배되는지 깨닫지 못한 채 먹어 왔을 것입니다. 그러나 수직 농업은 세계가 수행해야 하는 더
넓은 변화의 일부일 뿐입니다. 2050년까지 전 세계 인구가 거의 100억 명에 이를 것으로 예상되고 기후 위기가 악화됨에 따라 우리는 더 많은 식량을 더
지속 가능하게 생산해야 합니다

출처 : bowery 홈페이지
<https://bowery.co/>



Plants love Plenty®.

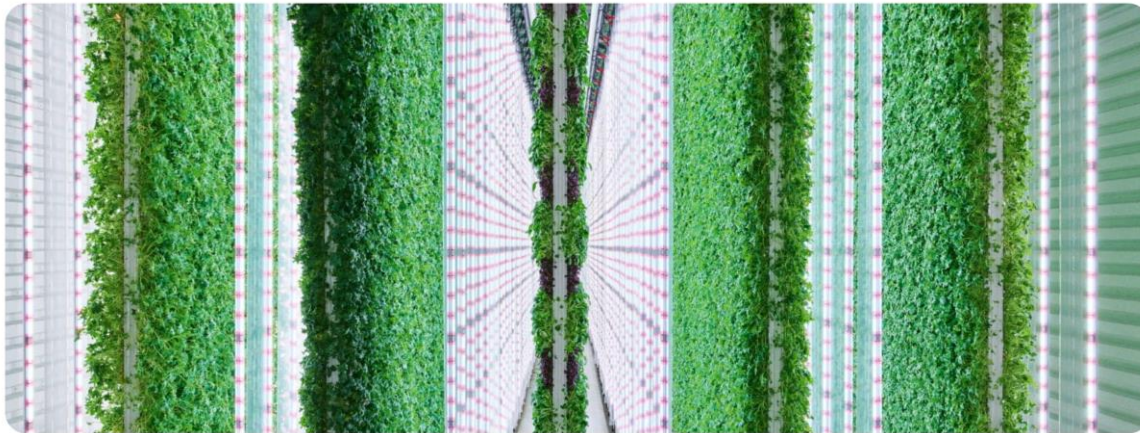
적절한 양의 빛, 영양분, 물이 적용된 상품 제공
인류와 지구를 생각한 생산법

We Create the Ideal Environment So Nature Happens Naturally.

적절한 양의 빛, 영양분, 물이 적용된 상품 제공
인류와 지구를 생각한 생산법

Plenty®

대부분의 스타트업은 브랜딩과 소셜미디어 활용
제품화까지의 모든 프로세스를 진행하고 있음



- Plenty는 기둥 모양의 배지를 이용하여 농작물을 키우고 배지 전체를 들어올릴 수 있는 로봇을 이용하여 배지를 장착하거나 해체하여 수확하는 방식
- 각 기둥모양의 배지에 다른 작물을 생산 가능하며, 작물의 종류에 따라 LED의 파장을 조정하여 다양한 작물에 대한 최적화가 가능함



Genetics
Instead of modifying plants to fit the environment, we modified the environment to fit the plants.

Software & Hardware
Our farms are our hardware, Plenty OS is our software, it's built around the needs of our plants.

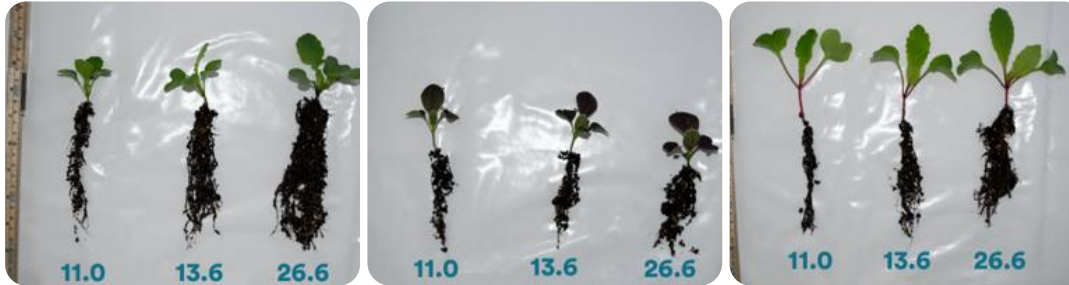
Plants
Our farms are designed to grow leafy greens you can eat without washing.

출처 : plenty 홈페이지
<https://www.plenty.ag/>

How to understand the impact of DLI: In-house Research Examples

- DLI가 다양한 작물에 어떤 영향을 미치는지 테스트 Heliospectra의 다이내믹 조명인 ELIXIA를 사용
- LED의 각각의 채널을 조정 하여 파장대역 간의 비율을 변경하여 DLI 목표수치를 높여 테스트 진행

Roots were larger in the higher DLI.



The DLI had an impact on the development of the crop

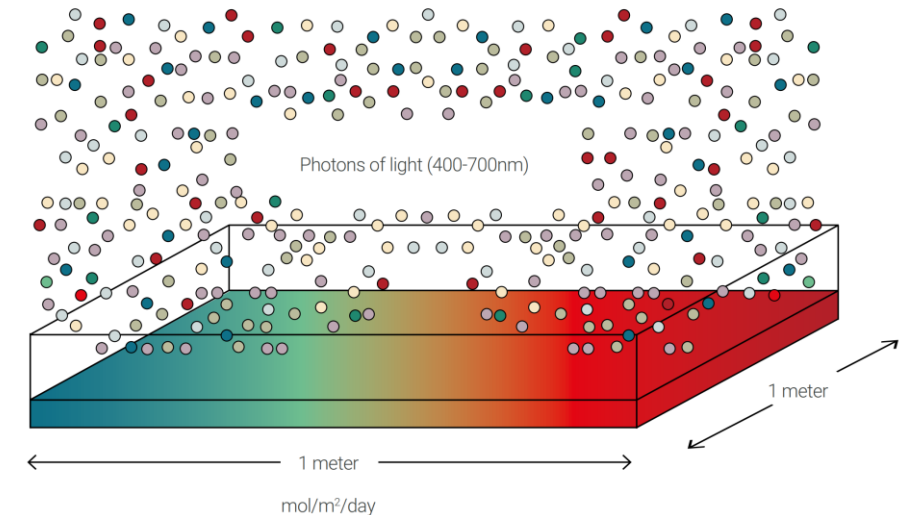


Helio Spectra의 차별점

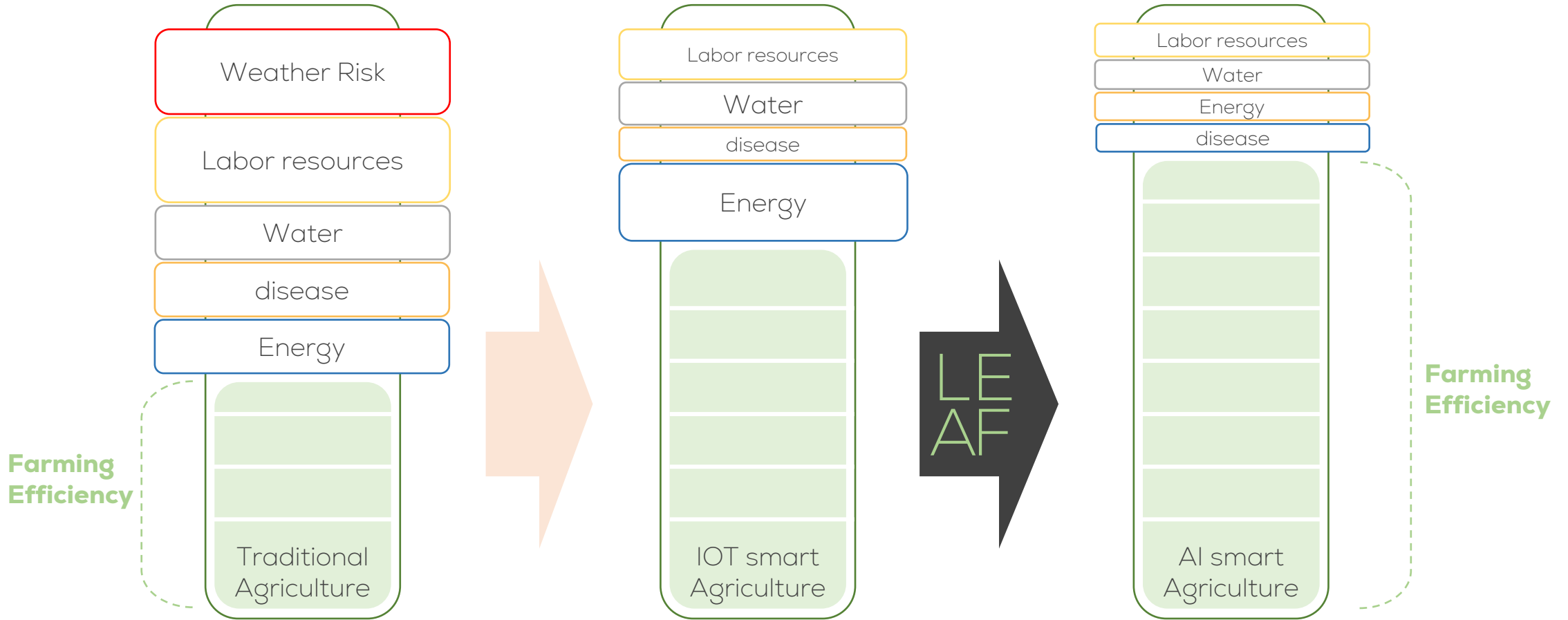
- 현재 시설을 바꾸지 않고 조명 시스템 교체만으로 품질 개선과 경작 기간확장 가능 솔루션
- 기후 변화에 의한 일조량 감소로 인한 농작물의 품질 저하에 대응가능

DLI. Daily Light Integral

DLI is the amount of PAR (Photosynthetically Active Radiation) that is received by your plants over the course of 24 hours
DLI 광합성 합성 가능 광량(PAR)을 단위면적($1m^2$) 24시간 측정하여 수치화 한 자료



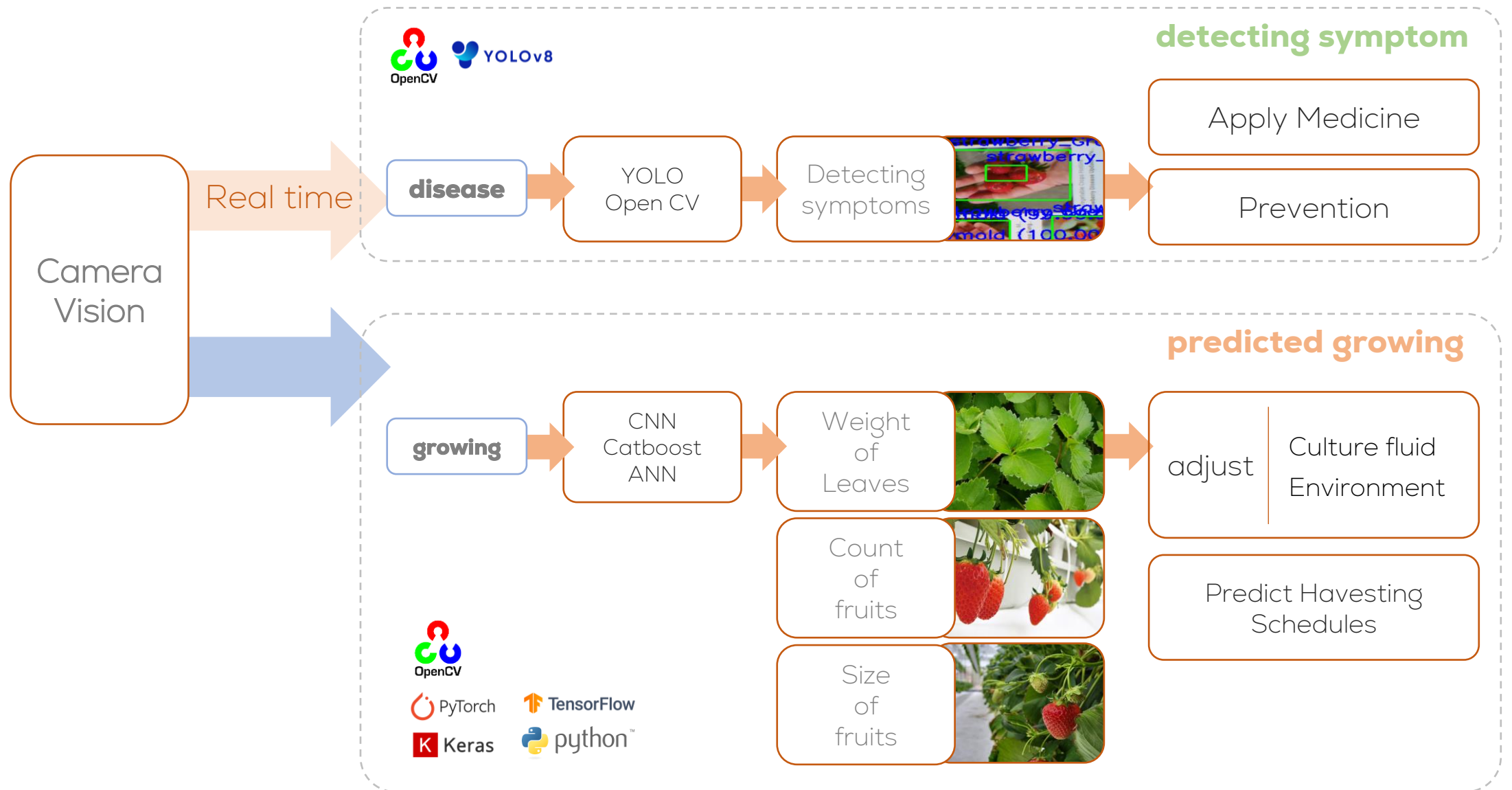
출처 : Helio Spectra
<https://heliospectra.com/>





LE
AF

our service



Preprocessing Data



153 images

전처리

Preprocessing

Auto-Orient: Applied

Resize: Stretch to 640x640

증폭

Augmentations

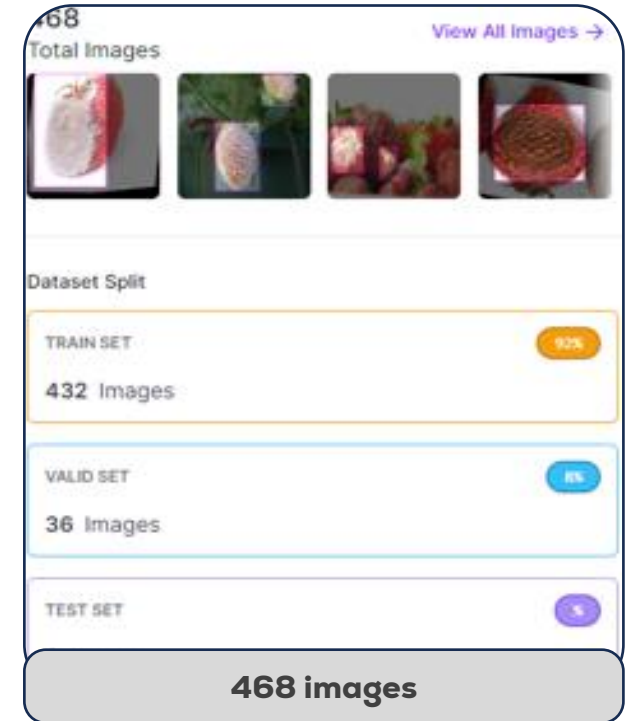
Outputs per training example: 3

Flip: Horizontal, Vertical

Crop: 0% Minimum Zoom, 20% Maximum Zoom

Rotation: Between -20° and $+20^\circ$

Shear: $\pm 15^\circ$ Horizontal, $\pm 15^\circ$ Vertical



Preprocessing Data

기능	라이브러리, 클래스	함수	파라미터
Image 불러오기		load_image image_to_array image.append	target_size = (128,128)
Image np.array 및 정규화	numpy	np.array	
Image 증식	Scikit_learn.model_selection Tensorflow Tensorflow addons math	rotate Random_brightness Random_contrast Random_saturation Random_zoom Random_shear Random_shift Random_shift	minval=-30, maxval=30, dtype=tf.float32 Max_delta = 0.5 Lower = 0.2, upper = 1.8 Lower = 0.5, upper = 1.5 zoom_range=(0.8, 1.2), row_axis=0, col_axis=1, channel_axis=2 intensity=0.2, row_axis=0, col_axis=1, channel_axis=2 wrg=0.2, hrg=0, row_axis=0, col_axis=1, channel_axis=2 wrg=0, hrg=0.2, row_axis=0, col_axis=1, channel_axis=2

Preprocessing Data

기능	라이브러리, 클래스	함수	파라미터
Label 생성	numpy	numpy.repeat numpy.array	[0,1,2],2000 (0,1,2를 2000번 반복하여 라벨링)
		image_to_array	9번 반복하여 라벨링
train test split	Scikit_learn.model_selection	Train test split	test_size=0.2, shuffle=True, random_state=42,stratify=original_labels
데이터 정규화	Tensorflow.keras	To_categorical	Num_classes = 3

1st Model

기능	라이브러리, 클래스	함수	파라미터
1st Convolution Block	tensorflow.keras.models	Conv2D	32, (3, 3), input_shape=(128, 128, 3),
		BatchNormalization	
		LeakyReLU	Alpha = 0.1
		Maxpooling2D	(2,2)
		Dropout	0.25
2nd Convolution Block	tensorflow.keras.models	Conv2D	64, (3, 3)
		BatchNormalization	
		LeakyReLU	Alpha = 0.1
		Maxpooling2D	(2,2)
		Dropout	0.25
3rd Convolution Block	tensorflow.keras.models	Conv2D	128, (3, 3)
		BatchNormalization	
		LeakyReLU	Alpha = 0.1
		Maxpooling2D	(2,2)
		Dropout	0.25
Flatten 및 완전 연결 레이어	tensorflow.keras.models	Flatten	
		Dense	128, activation = 'relu'
		Dropout	(0.5)
		Dense	Num_classes, activation = 'softmax'

validation_accuracy가 0.50으로 고정되는 overfitting 발생하여 class 개수를 2개로 줄여 두번째 Model Compile진행

2nd Model

기능	라이브러리, 클래스	함수	파라미터
1st Convolution Block	tensorflow.keras.models	Conv2D	32, (3, 3), input_shape=(128, 128, 3), padding = 'same'
		BatchNormalization	
		LeakyReLU	Alpha = 0.1
		Maxpooling2D	(2,2)
		Dropout	0.25
2nd Convolution Block	tensorflow.keras.models	Conv2D	64, (3, 3), padding = 'same'
		BatchNormalization	
		LeakyReLU	Alpha = 0.1
		Maxpooling2D	(2,2)
		Dropout	0.25
3rd Convolution Block	tensorflow.keras.models	Conv2D	128, (3, 3), padding = 'same'
		BatchNormalization	
		LeakyReLU	Alpha = 0.1
		Maxpooling2D	(2,2)
		Dropout	0.25
Flatten 및 완전 연결 레이어	tensorflow.keras.models	Flatten	
		Dense	128, activation = 'relu'
		Dropout	(0.5)
		Dense	Num_classes, activation = 'softmax'

1st model에 Conv2D의 매개 변수에 padding = 'same'을 추가하여 가상의 padding을 추가하여, 손실되는 부분을 줄이고 출력 이미지가 입력 이미지와 동일하게 만들어 손실을 줄일 수 있도록 수정하여 Model 제작 진행

Model Compile & Save

Model Compile

기능	라이브러리, 클래스	함수	파라미터
Model Compile	<code>tensorflow.keras.models</code>	<code>model.compile</code>	<code>optimizer=Adam(learning_rate=0.0001), loss='categorical_crossentropy', metrics=['accuracy']</code>
Model Train	<code>tensorflow.keras.models</code>	<code>model.fit</code>	<code>validation_data=(val_x, val_y), epochs=100, batch_size=64 , callbacks=[early_stopping, model_checkpoint]</code>

Model Save

기능	라이브러리, 클래스	함수	파라미터
Save model	<code>tensorflow.keras.models</code>	<code>model.save</code>	

Model Load

기능	라이브러리, 클래스	함수	파라미터
Load model	<code>tensorflow.keras.models</code>	<code>Load_model</code>	

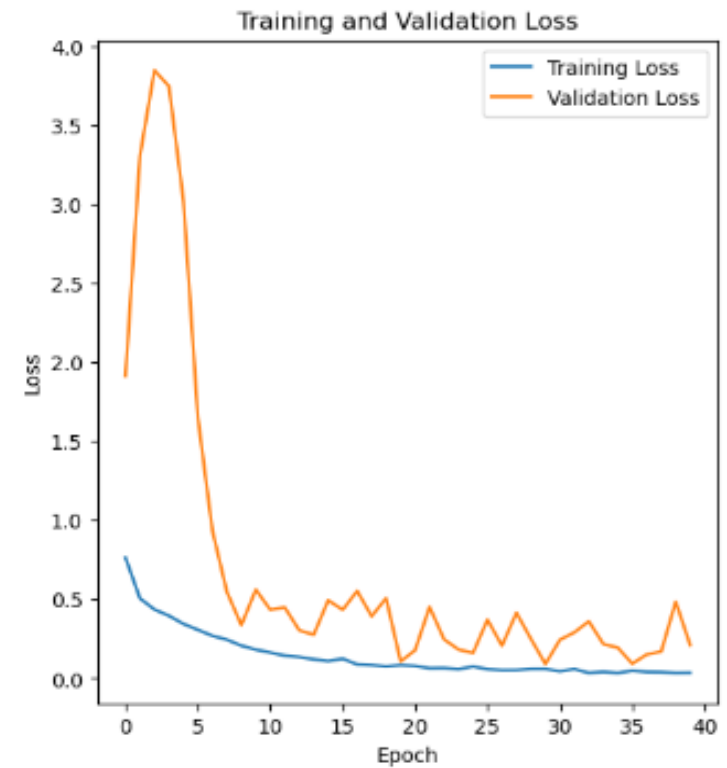
Model Visualization Accuracy

기능	라이브러리, 클래스	함수	파라미터
정확도 그래프	<code>matplotlib.pyplot</code>	<code>plt.figure</code>	<code>figsize=(12, 6)</code>
		<code>plt.subplot</code>	<code>(1, 2, 1)</code> # 1행 2열의 첫 번째 그래프
		<code>plt.plot</code>	<code>history['accuracy'], label='Training Accuracy'</code>
		<code>plt.plot</code>	<code>history['val_accuracy'], label='Validation Accuracy'</code>
		<code>plt.title plt.xlabel plt.ylabel</code>	
		<code>plt.legend</code>	



Model Visualization Loss

기능	라이브러리, 클래스	함수	파라미터
손실 그래프	<code>matplotlib.pyplot</code>	<code>plt.figure</code>	<code>figsize=(12, 6)</code>
		<code>plt.subplot</code>	<code>(1, 2, 2)</code> # 1행 2열의 첫 번째 그래프
		<code>plt.plot</code>	<code>history['loss'], label='Training Loss'</code>
		<code>plt.plot</code>	<code>history['val_loss'], label='Validation Loss'</code>
		<code>plt.title plt.xlabel plt.ylabel</code>	
		<code>plt.legend</code>	



Model Compile & Predict

Model Compile

기능	라이브러리, 클래스	함수	파라미터
Model Compile	<code>tensorflow.keras.models</code>	<code>model.evaluate</code>	

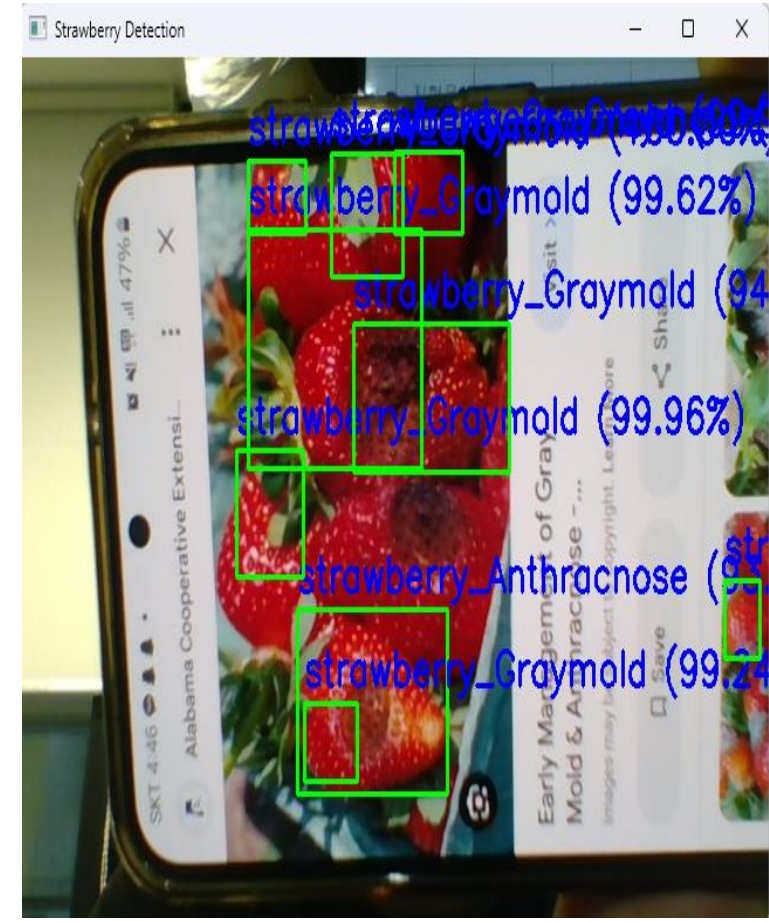
38/38 [=====] - 5s 114ms/step - loss: 0.2101 - accuracy: 0.9475 Loss: 0.21005438268184662,

Accuracy: 0.9474999904632568

Model Predict

기능	라이브러리, 클래스	함수	파라미터
Model Predict	<code>tensorflow.keras.models</code>	<code>model.predict</code>	
		<code>model.argmax</code>	<code>axis = 1</code>

Model Working



기능

data 경로 지정

train_image 추출
test_image 추출

Image data 전처리

meta data 결측값 처리

outlier 확인 및 제거

모듈, 클래스

OS, glob

cv2.cvtColor

convertScaleAbs

imread

resize

fillna

Plotly.scatter

파라미터

image, cv2.COLOR_BGR2GRAY

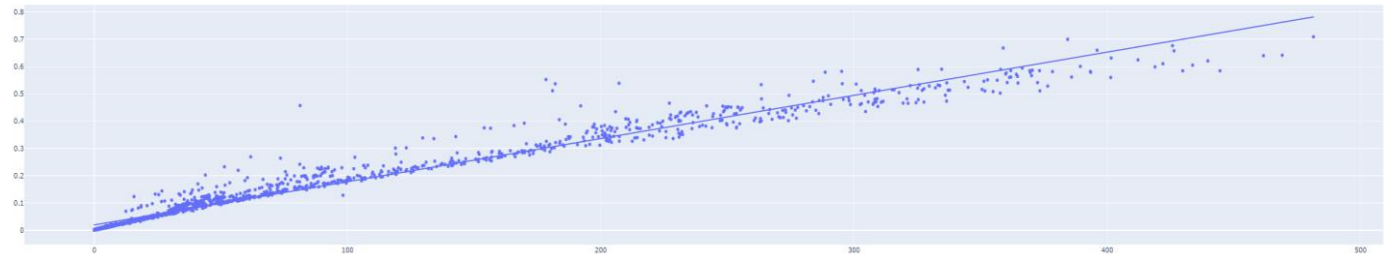
image, alpha=alpha, beta=beta

Imagesize 1000 x 750

method='bfill', limit=1

method='ffill', limit=1

trendline="ols"



```
image_outliers = ['CASE73_10', 'CASE23_08', 'CASE23_07', 'CASE23_06', 'CASE23_09', 'CASE23_05', 'CASE23_04',
                  'CASE70_21', 'CASE02_09', 'CASE02_10', 'CASE23_02', 'CASE23_03', 'CASE70_23', 'CASE23_01', 'CASE23_02', 'CASE05_21']
```



기능	라이브러리, 함수	파라미터
Image data tesor	pytorch. transform.compose	ToTensor() Resize((CFG['IMG_SIZE'], CFG['IMG_SIZE'])) Normalize(mean=(0.5, 0.5, 0.5), std=(0.5, 0.5, 0.5))
CNN 모델 학습	Layer 1	Conv2d(3, 8, kernel_size=3, stride=1, padding=1) Relu(), MaxPool2d(kernel_size=2, stride=2))
	Layer 2	Conv2d(8, 16, kernel_size=3, stride=1, padding=1) Relu(), MaxPool2d(kernel_size=2, stride=2))
	Layer 3	Conv2d(16, 32, kernel_size=3, stride=1, padding=1) Relu(), MaxPool2d(kernel_size=2, stride=2))
	Layer 4	Conv2d(32, 64, kernel_size=4, stride=1, padding=1) Relu(), MaxPool2d(kernel_size=2, stride=2))
	regressor	Linear(3136, 1)

기능

Metadata
Outlier 판단 및 제거

모듈, 함수

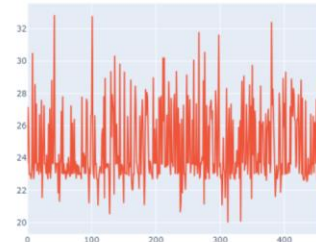
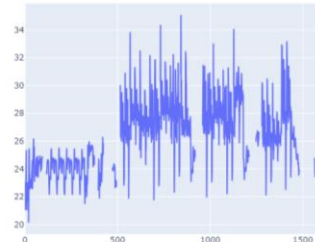
plotly.graph_objects

파라미터

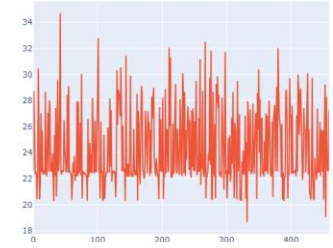
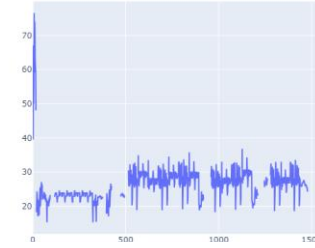
```
Scatter(x = train_df[feat].index, y = train_df[feat], text=train_df["이미지"])
```

```
Scatter(x = test_df[feat].index, y = test_df[feat], text=test_df["이미지"])
```

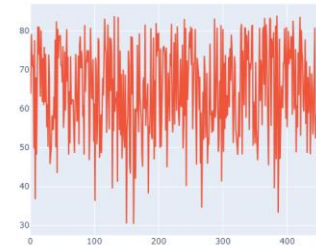
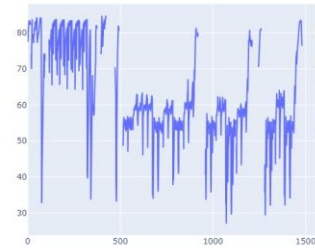
내부온도관측지



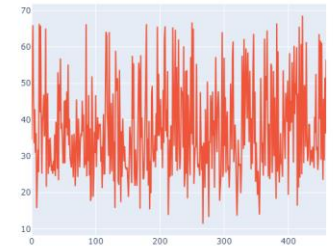
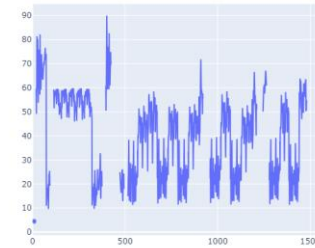
외부온도관측지



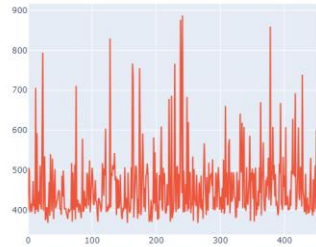
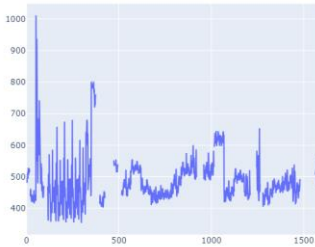
내부습도관측지



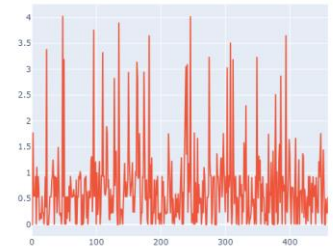
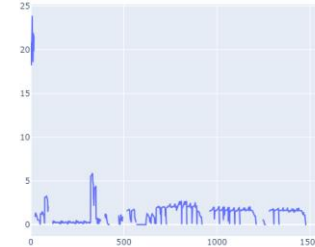
외부습도관측지



CO2관측지



:C관측지



기능

모듈, 함수

파라미터

상관관계 확인

corr

	내부온도관측치	외부온도관측치	내부습도관측치	외부습도관측치	CO2관측치	EC관측치	최근분무량	화이트 LED동작강도	레드 LED동작강도	블루 LED동작강도	냉방온도	냉방부하	난방온도	난방부하	총추정광량	백색광추정광량	적색광추정광량	청색광추정광량	무게	비율
내부온도관측치	1.000000	0.914724	-0.593916	-0.347328	0.090311	0.228836	-0.259776	-0.026716	-0.366753	-0.412775	0.031750	0.606765	0.302380	-0.250200	-0.345095	-0.094399	-0.366756	-0.412765	-0.099403	-0.126872
외부온도관측치	0.914724	1.000000	-0.497909	-0.275979	-0.027334	0.103698	-0.217581	-0.101537	-0.393903	-0.519573	-0.013787	0.594182	0.231268	-0.270714	-0.440984	-0.155695	-0.393897	-0.519554	-0.010985	-0.036581
내부습도관측치	-0.593916	-0.497909	1.000000	0.749341	-0.032015	-0.511126	0.199216	-0.085434	0.276872	0.194409	-0.045466	-0.184345	-0.346947	0.194310	0.142151	-0.025914	0.276888	0.194416	0.227335	0.273139
외부습도관측치	-0.347328	-0.275979	0.749341	1.000000	-0.104920	-0.500869	-0.031541	-0.221013	0.178282	-0.070088	-0.001313	-0.027708	-0.254321	0.225952	-0.107386	-0.180841	0.178279	-0.070093	0.111783	0.134949
CO2관측치	0.090311	-0.027334	-0.032015	-0.104920	1.000000	0.107233	-0.105626	0.138971	-0.019837	0.072781	-0.167474	0.203515	-0.122112	-0.107056	0.135839	0.151923	-0.019829	0.072792	0.021746	0.008204
EC관측치	0.228836	0.103698	-0.511126	-0.500869	0.107233	1.000000	-0.186181	0.255513	0.137186	0.360218	0.085758	-0.028903	0.254893	-0.059108	0.355251	0.230412	0.137186	0.360228	-0.149281	-0.164157
최근분무량	-0.259776	-0.217581	0.199216	-0.031541	-0.105626	-0.186181	1.000000	0.277216	-0.034867	0.251823	-0.133820	-0.086099	-0.113847	-0.131925	0.288387	0.268771	-0.034861	0.251792	-0.276084	-0.267159
화이트 LED동작강도	-0.026716	-0.101537	-0.085434	-0.221013	0.138971	0.255513	0.277216	1.000000	-0.341661	0.365943	-0.357645	0.019818	-0.003182	-0.381531	0.748282	0.951325	-0.341608	0.365962	-0.328852	-0.342631
레드 LED동작강도	-0.366753	-0.393903	0.276872	0.178282	-0.019837	0.137186	-0.034867	-0.341661	1.000000	0.489848	0.292528	-0.130601	-0.264633	0.594818	0.271502	-0.306347	1.000000	0.489854	0.241593	0.264778
블루 LED동작강도	-0.412775	-0.519573	0.194409	-0.070088	0.072781	0.360218	0.251823	0.365943	0.489848	1.000000	-0.104378	-0.270368	-0.052459	-0.006212	0.819290	0.371250	0.489872	1.000000	0.067044	0.082647
냉방온도	0.031750	-0.013787	-0.045466	-0.001313	-0.167474	0.085758	-0.133820	-0.357645	0.292528	-0.104378	1.000000	-0.491042	0.584062	0.369729	-0.187875	-0.317848	0.292477	-0.104430	0.095894	0.100295
냉방부하	0.606765	0.594182	-0.184345	-0.027708	0.203515	-0.028903	-0.086099	0.019818	-0.130601	-0.270368	-0.491042	1.000000	-0.535321	0.097171	-0.176282	-0.040681	-0.130560	-0.270315	-0.046680	-0.068669
난방온도	0.302380	0.231268	-0.346947	-0.254321	-0.122112	0.254893	-0.113847	-0.003182	-0.264633	-0.052459	0.584062	-0.535321	1.000000	-0.374795	-0.096977	0.007223	-0.264689	-0.052514	-0.031174	-0.033069
난방부하	-0.250200	-0.270714	0.194310	0.225952	-0.107056	-0.059108	-0.131925	-0.381531	0.594818	-0.006212	0.369729	0.097171	-0.374795	1.000000	-0.088047	-0.360688	0.594802	-0.006215	0.198710	0.204306
총추정광량	-0.345095	-0.440984	0.142151	-0.107386	0.135839	0.355251	0.288387	0.748282	0.271502	0.819290	-0.187875	-0.176282	-0.096977	-0.088047	1.000000	0.797662	0.271548	0.819304	-0.116598	-0.111288
백색광추정광량	-0.094399	-0.155695	-0.025914	-0.180841	0.151923	0.230412	0.268771	0.951325	-0.306347	0.371250	-0.317848	-0.040681	0.007223	-0.360688	0.797662	1.000000	-0.306298	0.371266	-0.290968	-0.301607
적색광추정광량	-0.366756	-0.393897	0.276888	0.178279	-0.019829	0.137186	-0.034861	-0.341608	1.000000	0.489872	0.292477	-0.130560	-0.264689	0.594802	0.271548	-0.306298	1.000000	0.489877	0.241605	0.264791
청색광추정광량	-0.412765	-0.519554	0.194416	-0.070093	0.072792	0.360228	0.251792	0.365962	0.489854	1.000000	-0.104430	-0.270315	-0.052514	-0.006215	0.819304	0.371266	0.489877	1.000000	0.067066	0.082673
무게	-0.099403	-0.010985	0.227335	0.111783	0.021746	-0.149281	-0.276084	-0.328852	0.241593	0.067044	0.095894	-0.046680	-0.031174	0.198710	-0.116598	-0.290968	0.241605	0.067066	1.000000	0.991685
비율	-0.126872	-0.036581	0.273139	0.134949	0.008204	-0.164157	-0.267159	-0.342631	0.264778	0.082647	0.100295	-0.068669	-0.033069	0.204306	-0.111288	-0.301607	0.264791	0.082673	0.991685	1.000000

기능

CATboost 모델 학습

CATboost 모델 검증

CATboost
모델 검증 시각화

CATboost 평가

모듈, 클래스

CatBoostRegressor

fit

CATmodel.predict

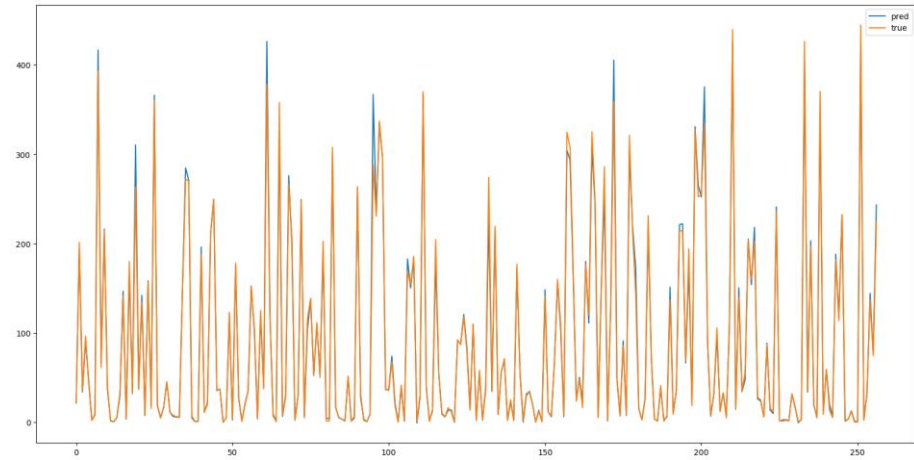
matplotlib

CATmodel.score

파라미터

```
verbose=50,n_estimators=10000,eval_metric='MAE',
early_stopping_rounds=50
```

```
train_x, train_y, eval_set=[(val_x, val_y)], use_best_model=True
```



```
bestTest = 4.685611077 bestIteration = 5912
```




기능	모듈, 클래스	파라미터
ANN 모델 정의	Sequential MLP정의	Dense(100, kernel_initializer='normal', activation='relu'), Dense(50, kernel_initializer='normal', activation='relu'), Dense(25, kernel_initializer='normal', activation='relu'), Dense(1, kernel_initializer='normal', activation='linear')
ANN 모델 compile	ANNmodel.compile	loss=mae, optimizer=Adam(learning_rate=0.001), metrics=[mae]
ANN 학습	ANNmodel.fit	epochs=1000, batch_size=32, verbose=2
ANN 예측	ANNmodel.predict	
Ensemble		CNN, Catboost, ANN score 가중치 적용하여 계산 (가중치의 합 = 1)