실습 5

Logistic Regression을 사용한 재활용품 분류

[Lecture] Dr. HeeSuk Kim

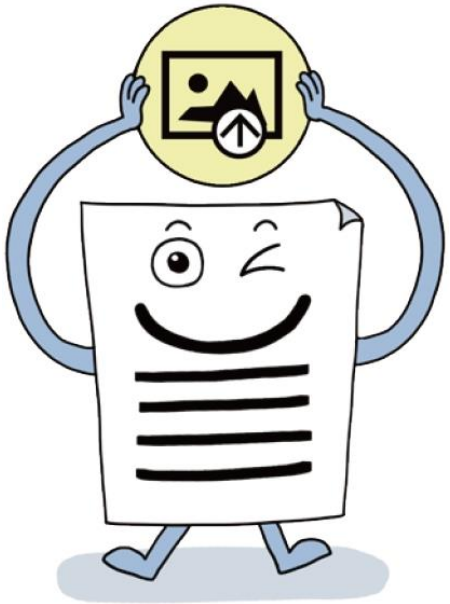
5

똑똑하게
재활용하자!

Logistic Regression을 사용하여
재활용품을 분류해 보자.

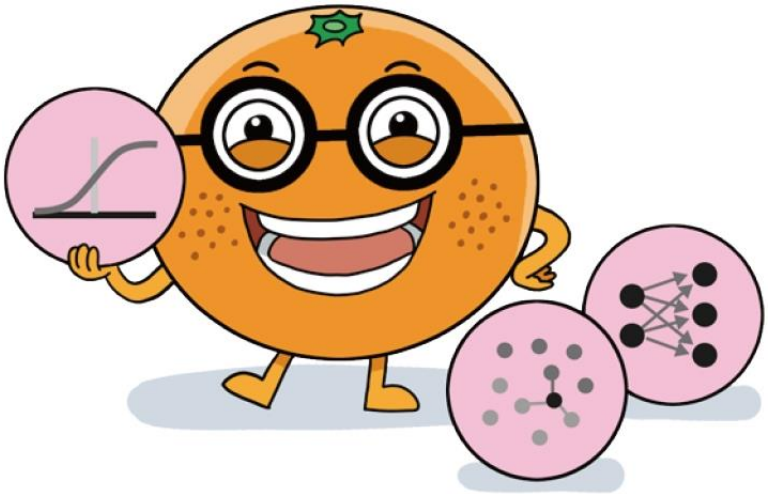
데이터 종류:

비정형 데이터



사용하는 모델:

Logistic Regression



1 해결해야 할 문제는 무엇일까?

문제 상황

환경부 조사에 따르면 우리나라 국민 한 사람이 70년간 배출하는 생활 쓰레기는 무려 55톤에 달한다고 한다. 사람들의 구매 욕구와 생활의 편의로 생활 쓰레기는 급격하게 증가하고 있으며, 이에 따라 여러 가지 환경 문제가 발생하고 있다. 이러한 상황에서 올바른 재활용 분리 배출은 문제 해결의 첫 걸음이 될 수 있을 뿐만 아니라 지속 가능한 자원 순환의 시작이 될 것이다.

하지만 잘못 분리 배출된 재활용 쓰레기는 재사용이 불가능하고, 이를 처리하는 데 더 많은 비용이 발생한다. 인공지능을 활용하여 효율적으로 분리수거를 할 수는 없을까?

재활용품 **이미지** 데이터를 분석하고 학습한 후, **효율적으로 재활용품을 분리**할 수 있는 인공지능 모델을 만들어 보자.

2 데이터를 준비하자!

데이터 다운로드 링크

https://bit.ly/3DnJnWt

recyclable_materials.zip

1 외부 데이터 다운로드

① 캐글 데이터 다운로드하기

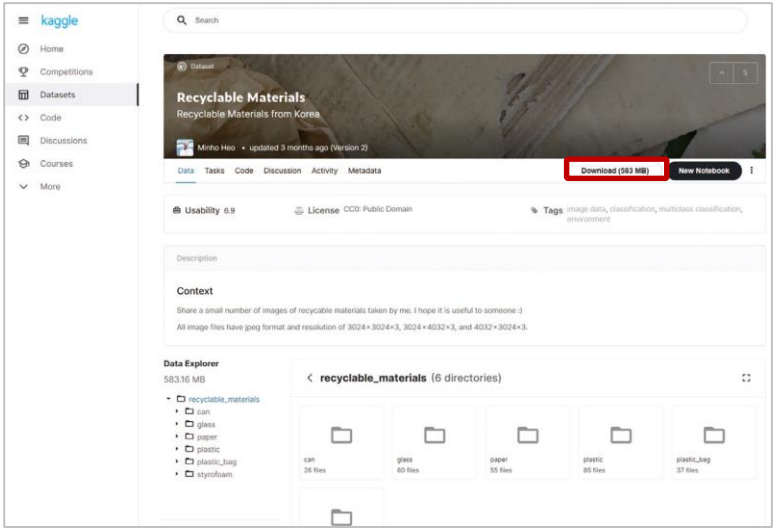
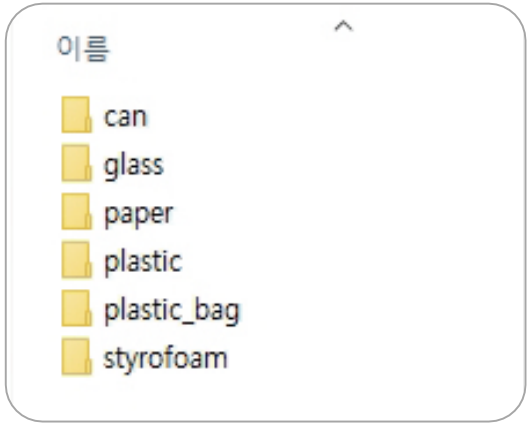


그림 5-1 캐글의 Recyclable Materials 데이터

- 캐글에서 'Recyclable Materials'을 검색하여 데이터 다운로드

② 데이터 살펴보기

- recyclable_materials.zip 파일을 풀어 recyclable_materials 폴더 안에 재활용품 종류에 따라 폴더가 나뉘어져 있는 것을 확인
- Orange3에서 이미지 데이터를 모델에 학습시켜 분류나 예측에 활용할 때 폴더명이 데이터의 레이블, 즉 정답이 됨.
- 데이터는 **can, glass, paper, plastic, plastic_bag, styrofoam** 총 6개의 레이블로 구성



폴더명	이미지 개수
can	24
glass	56
paper	52
plastic	83
plastic_bag	34
styrofoam	18

이미지의 개수는 총 267개이다.

③ 훈련 데이터와 테스트 데이터 나누기

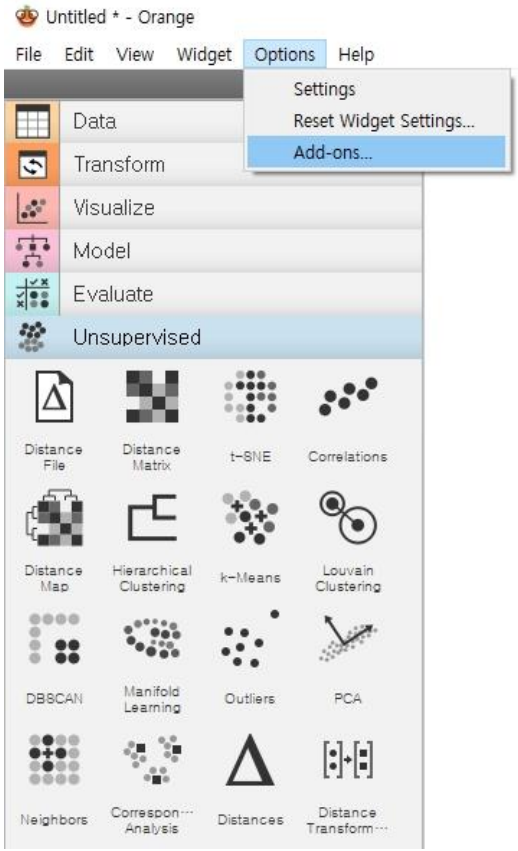
- 이미지 학습을 마친 인공지능 모델을 테스트에 활용하기 위해 다음과 같이 test 폴더를 새로 만들고, 다운로드한 이미지 파일의 일부를 임의로 몇 가지 선택하여 test 폴더로 옮긴다.

2 데이터 불러오기

① 카테고리에 기능 추가하기

- 이미지를 분석하기 위해 이미지 분석과 관련된 위젯으로 구성되어 있는 Image Analytics 카테고리 설치

- [Options] 메뉴에서 [Add-ons...]를 클릭하면 Installer 창이 나타난다.
  이 창에서 기능을 추가로 설치할 수 있으며, 이 중에서 Image Analytics를 체크(☑) 하고 OK를 클릭하여 카테고리에 기능을 추가로 설치한다.



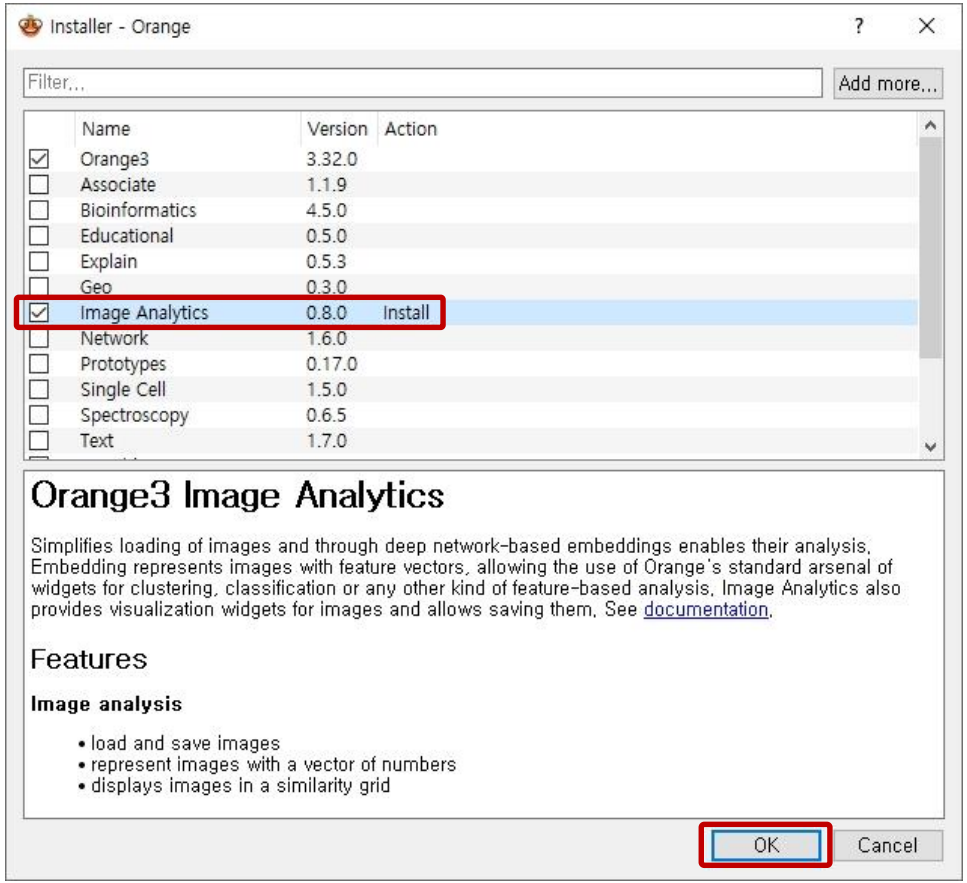


그림 5-2 Image Analytics 추가

- Image Analytics 설치가 완료된 후 Orange3가 재실행되면 [그림 5-3]과 같이 카테고리에 Image Analytics 가 추가된 것을 확인할 수 있다.

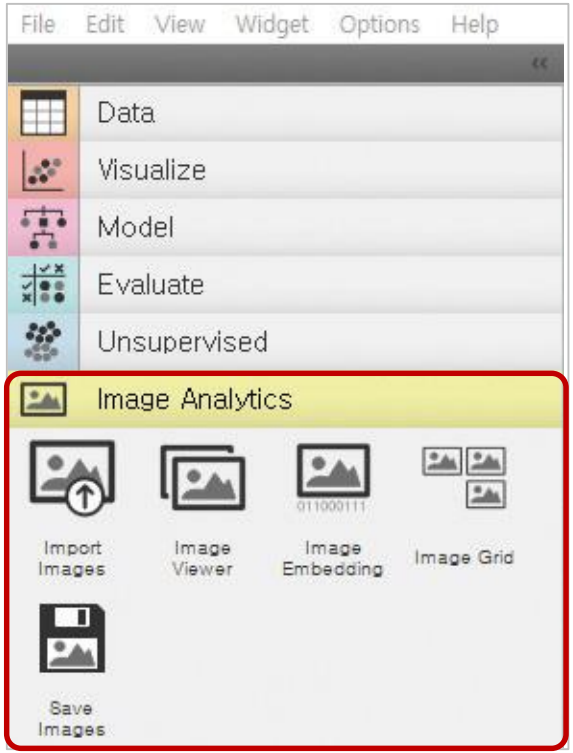
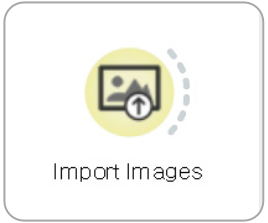
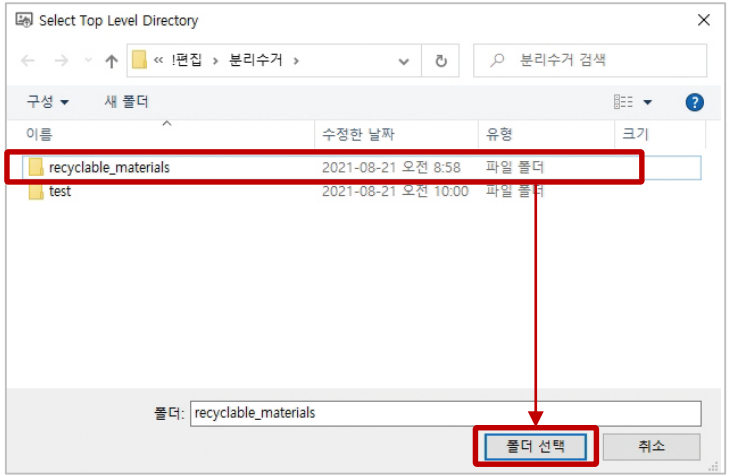


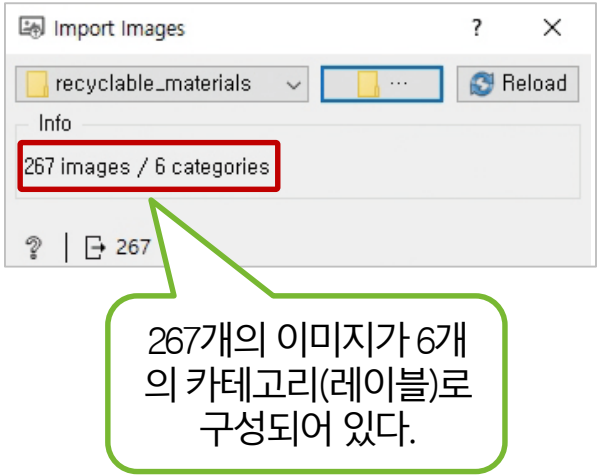
그림 5-3 추가된 Image Analytics

② 데이터 불러오기

- Image Analytics 카테고리의 **[Import Images]** 위젯을 가져와서 더블클릭한 후, recyclable_materials 폴더 선택
- [Import Images] 위젯은 폴더 안의 이미지 데이터를 한꺼번에 가져올 수 있는 기능을 제공함.







- Image Analytics 카테고리의 [Image Viewer] 위젯을 가져와서 [Import Images] 위젯과 연결하여 이미지 확인
- Title Attribute를 category로 설정하면 [그림 5-4]와 같이 **이미지 하단에 레이블(폴더명)**이 **표시**됨

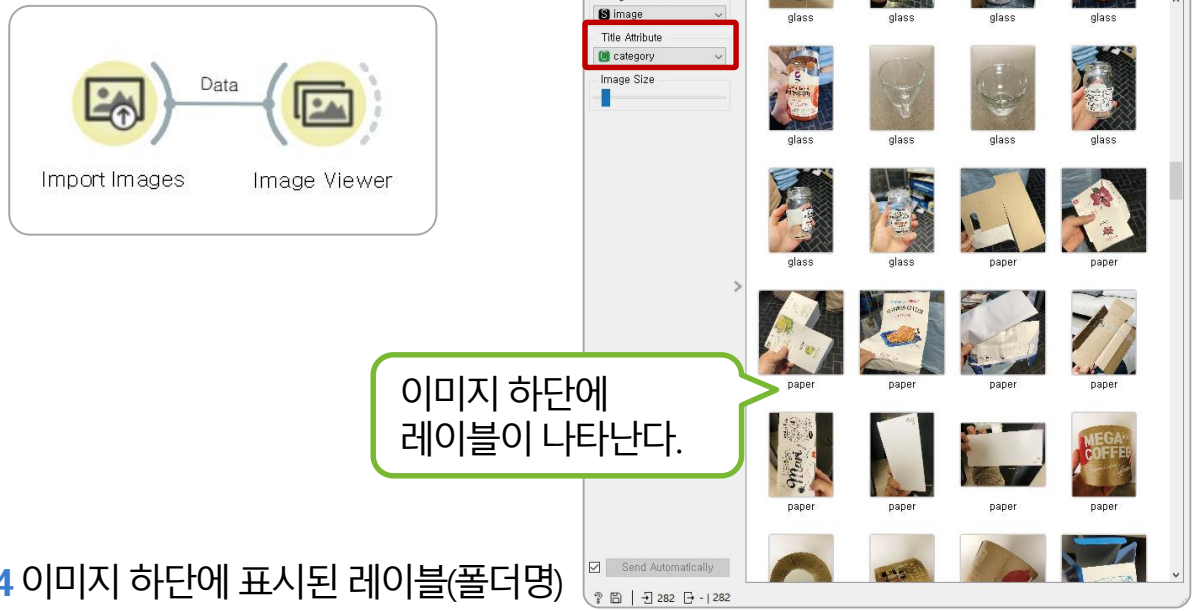
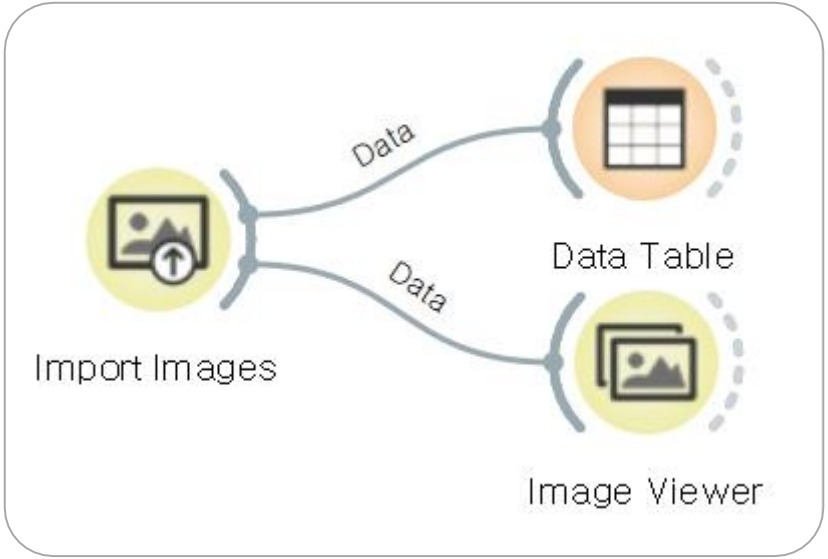


그림 5-4 이미지 하단에 표시된 레이블(폴더명)

③ 데이터 속성 확인하기

- Data 카테고리의 [Data Table] 위젯을 가져와서 [Import Images] 위젯과 연결



- [그림 5-5]를 보면 데이터는 **6개의 category**로 구분되어 있고,
  파일명(image name), 파일 경로, 크기(size), 너비(width), 높이(height)로 구성되어 있는 것을 확인
- 이미지 데이터를 불러왔지만 속성을 파악할 수 없기 때문에 이미지의 특징을 기반으로
  분류 작업을 수행할 수 없음

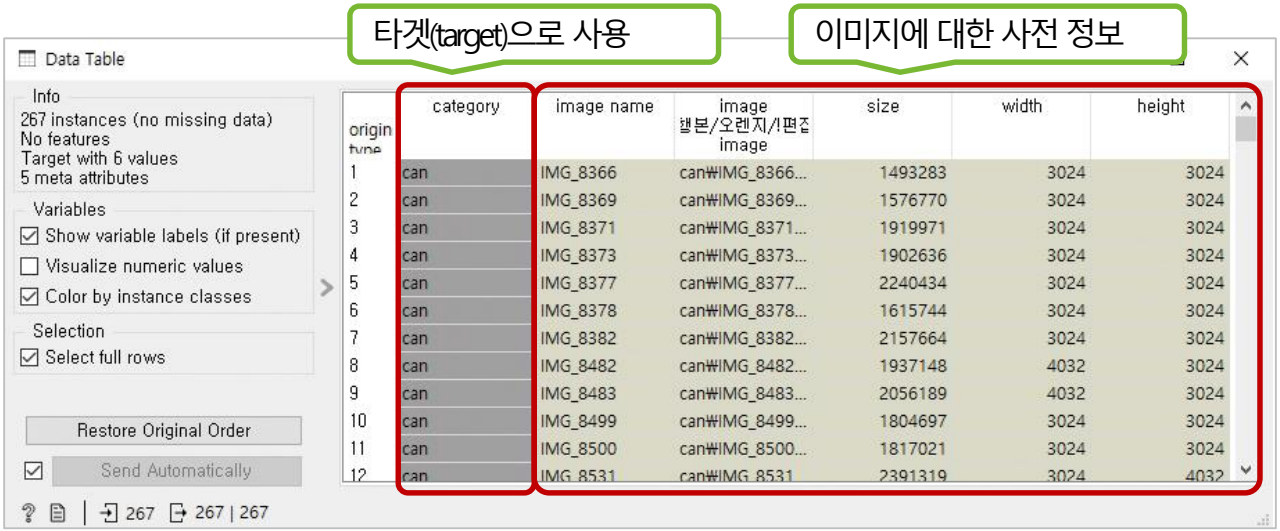
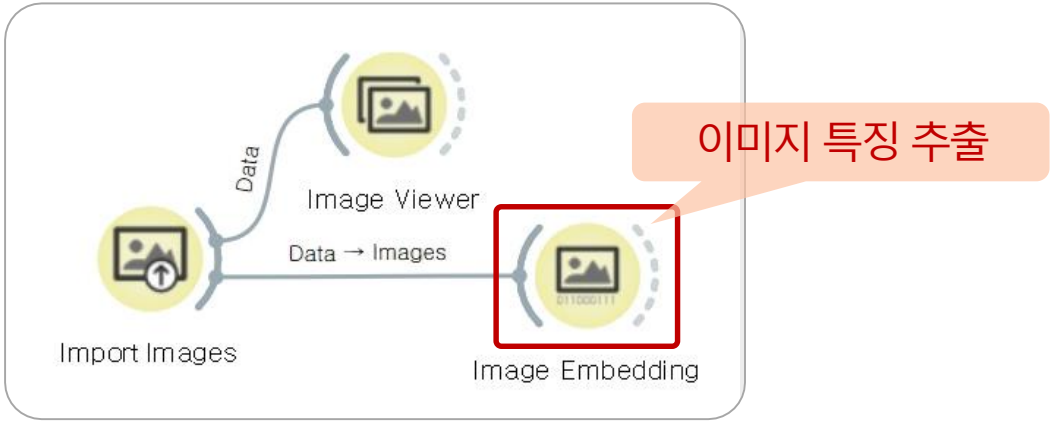


그림 5-5 Data Table 창에서 본 Recydable Materials 데이터

3 데이터 전처리하기

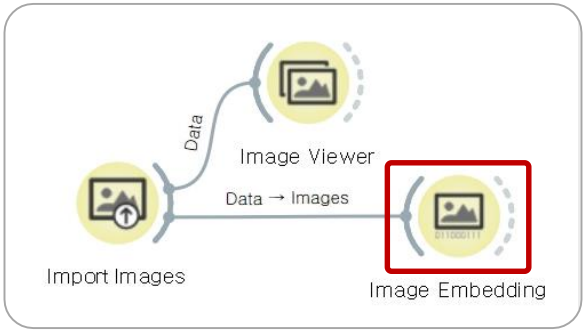
이미지 데이터는 **비정형 데이터**이기 때문에 기계학습에 적합한 형태로 바꾸는 **전처리** 작업이 필요하다. 이 과정을 **이미지 임베딩**이라고 하며 Orange3에서는 [Image Embedding]위젯을 사용한다.

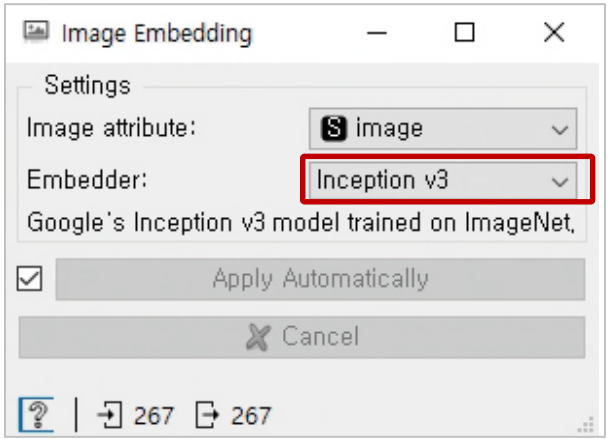


① 이미지 임베딩하기

- Image Analytics 카테고리의 [Image Embedding] 위젯을 가져와서 [Import Images] 위젯과 연결한다.

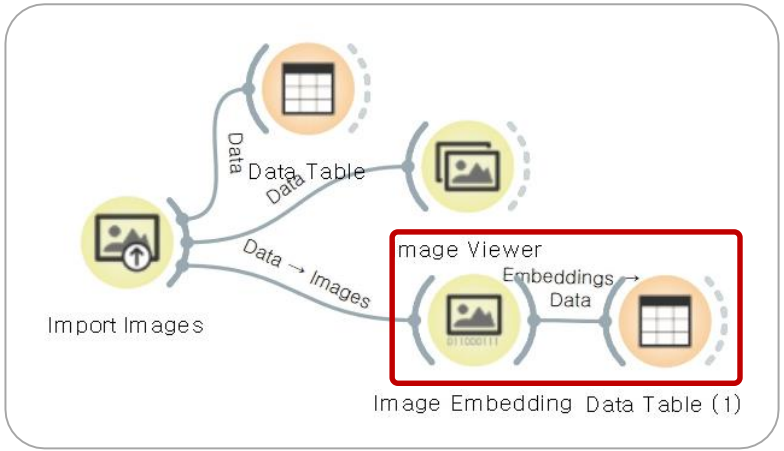
- [Image Embedding] 위젯을 더블 클릭하면 임베더(Embedder)를 선택할 수 있다. 이 활동에서는 기본 설정값인 **Inception v3**를 사용한다.



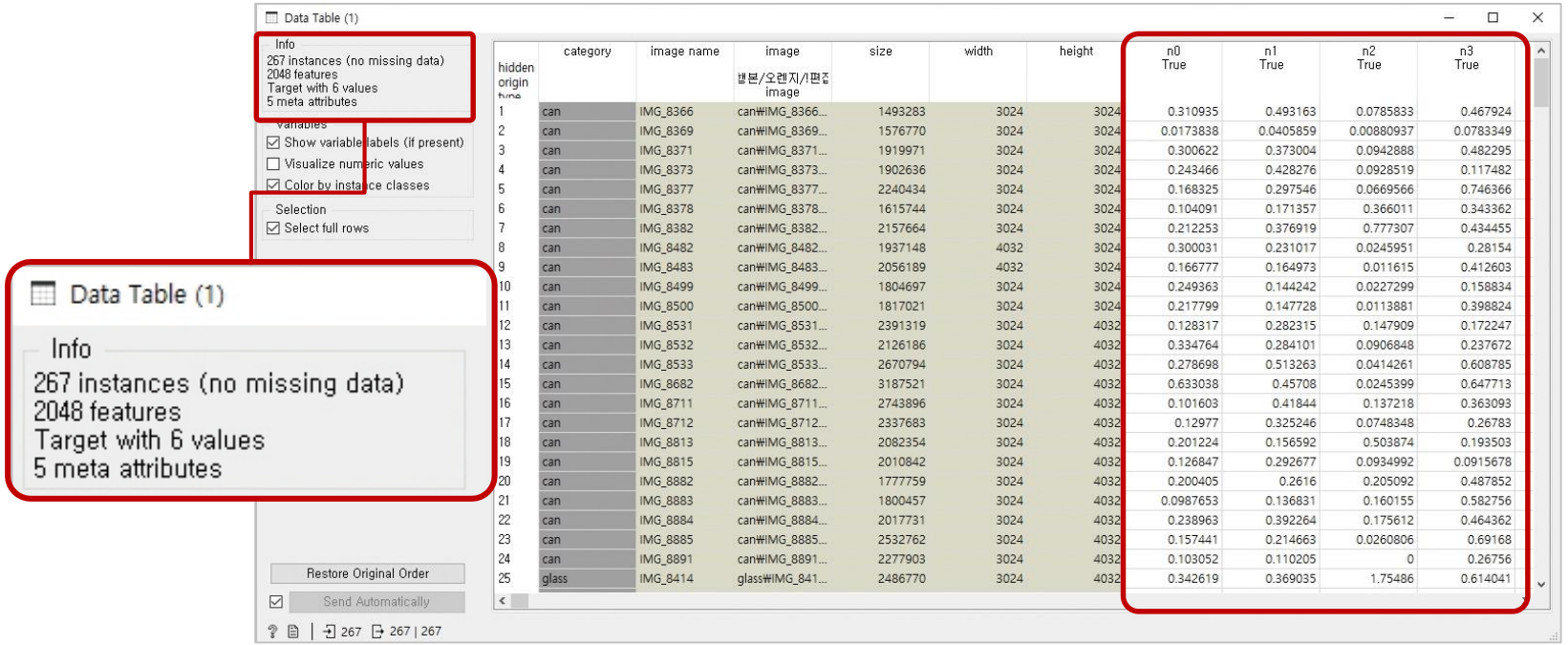


② 임베딩한 이미지 데이터 속성 확인하기

- Data 카테고리의 [Data Table] 위젯을 새로 가져와서 [Image Embedding] 위젯과 연결하여 데이터의 속성을 확인해 보자.



- 임베딩하기 전과 달리 **n0에서 n2047까지** 총 **2048개의 feature**가 추출되었으며, 추출된 feature에 의해 기계학습이 가능해졌다.



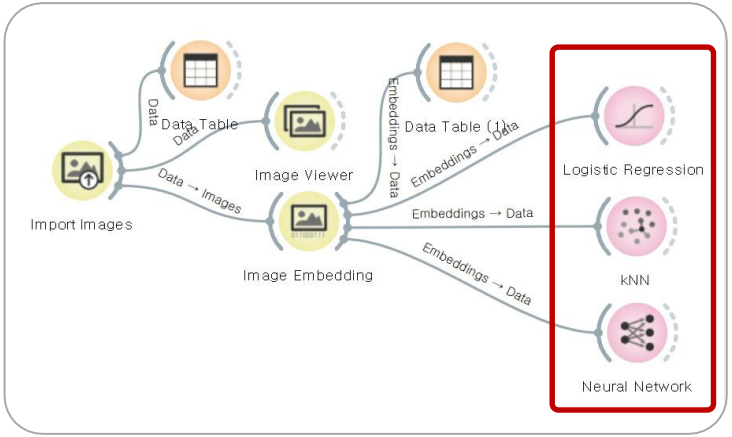
hidden origin type	category	image name	image 밸본/오렌지/1편집 image	size	width	height	n0 True	n1 True	n2 True	n3 True
1	can	IMG_8366	can₩IMG_8366...	1493283	3024	3024	0.310935	0.493163	0.0785833	0.467924
2	can	IMG_8369	can₩IMG_8369...	1576770	3024	3024	0.0173838	0.0405859	0.00880937	0.0783349
3	can	IMG_8371	can₩IMG_8371...	1919971	3024	3024	0.300622	0.373004	0.0942888	0.482295
4	can	IMG_8373	can₩IMG_8373...	1902636	3024	3024	0.243466	0.428276	0.0928519	0.117482
5	can	IMG_8377	can₩IMG_8377...	2240434	3024	3024	0.168325	0.297546	0.0669566	0.746366
6	can	IMG_8378	can₩IMG_8378...	1615744	3024	3024	0.104091	0.171357	0.366011	0.343362
7	can	IMG_8382	can₩IMG_8382...	2157664	3024	3024	0.212253	0.376919	0.777307	0.434455
8	can	IMG_8482	can₩IMG_8482...	1937148	4032	3024	0.300031	0.231017	0.0245951	0.28154
9	can	IMG_8483	can₩IMG_8483...	2056189	4032	3024	0.166777	0.164973	0.011615	0.412603
10	can	IMG_8499	can₩IMG_8499...	1804697	3024	3024	0.249363	0.144242	0.0227299	0.158834
11	can	IMG_8500	can₩IMG_8500...	1817021	3024	3024	0.217799	0.147728	0.0113881	0.398824
12	can	IMG_8531	can₩IMG_8531...	2391319	3024	4032	0.128317	0.282315	0.147909	0.172247
13	can	IMG_8532	can₩IMG_8532...	2126186	3024	4032	0.334764	0.284101	0.0906848	0.237672
14	can	IMG_8533	can₩IMG_8533...	2670794	3024	4032	0.278698	0.513263	0.0414261	0.608785
15	can	IMG_8682	can₩IMG_8682...	3187521	3024	4032	0.633038	0.45708	0.0245399	0.647713
16	can	IMG_8711	can₩IMG_8711...	2743896	3024	4032	0.101603	0.41844	0.137218	0.363093
17	can	IMG_8712	can₩IMG_8712...	2337683	3024	4032	0.12977	0.325246	0.0748348	0.26783
18	can	IMG_8813	can₩IMG_8813...	2082354	3024	4032	0.201224	0.156592	0.503874	0.193503
19	can	IMG_8815	can₩IMG_8815...	2010842	3024	4032	0.126847	0.292677	0.0934992	0.0915678
20	can	IMG_8882	can₩IMG_8882...	1777759	3024	4032	0.200405	0.2616	0.205092	0.487852
21	can	IMG_8883	can₩IMG_8883...	1800457	3024	4032	0.0987653	0.136831	0.160155	0.582756
22	can	IMG_8884	can₩IMG_8884...	2017731	3024	4032	0.238963	0.392264	0.175612	0.464362
23	can	IMG_8885	can₩IMG_8885...	2532762	3024	4032	0.157441	0.214663	0.0260806	0.69168
24	can	IMG_8891	can₩IMG_8891...	2277903	3024	4032	0.103052	0.110205	0	0.26756
25	glass	IMG_8414	glass₩IMG_841...	2486770	3024	4032	0.342619	0.369035	1.75486	0.614041

그림 5-6 이미지 임베딩 후 속성 변화

3 어떤 모델을 선택하고 학습시킬까?

1 학습 모델 선택하기

- 재활용품을 분류하는 인공지능을 만들기 위해 분류에 좋은 성능을 보이는 로지스틱 회귀(Logistic Regression), k-최근접 이웃(kNN, K-Nearest Neighbor), 인공 신경망(Neural Network) 모델을 활용한다.
- Model 카테고리의 **[Logistic Regression], [kNN], [Neural Network]** 위젯을 가져와 **[Image Embedding]** 위젯에 연결한다.



2 모델 학습시키기

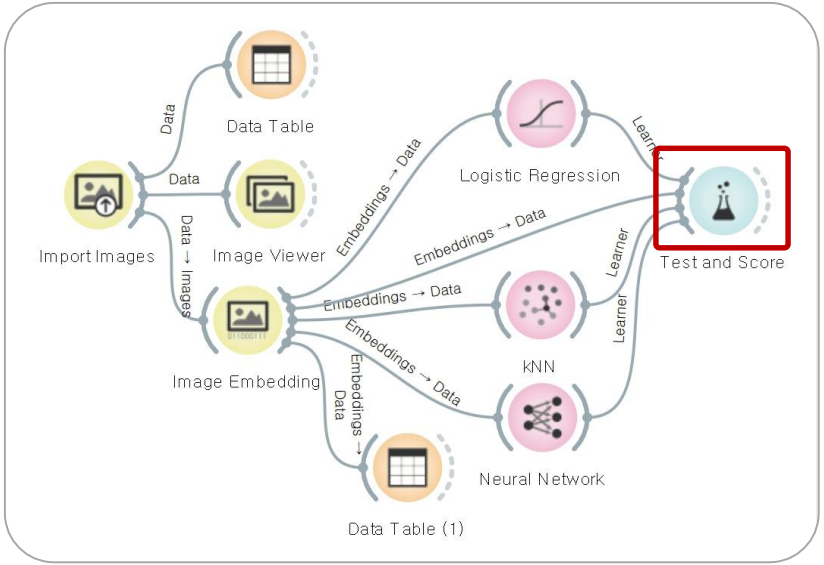
- 별도의 실행 명령을 주지 않아도 위젯을 연결하면 모델 위젯이 자동으로 실행되어 각 모델이 데이터를 학습하게 된다.

4 모델의 성능을 확인해 보자!

1 학습 결과 확인하기

① 성능 확인하기

- Evaluate 카테고리의 **[Test and Score]** 위젯을 가져온 후, 각 모델 위젯과 **[Image Embedding]** 위젯에 연결



- [Test and Score] 위젯을 더블 클릭하여 각 모델의 성능을 확인
- 데이터 샘플링 방법은 교차 검증(Cross validation)을 선택하고, Number of folds는 5로 설정

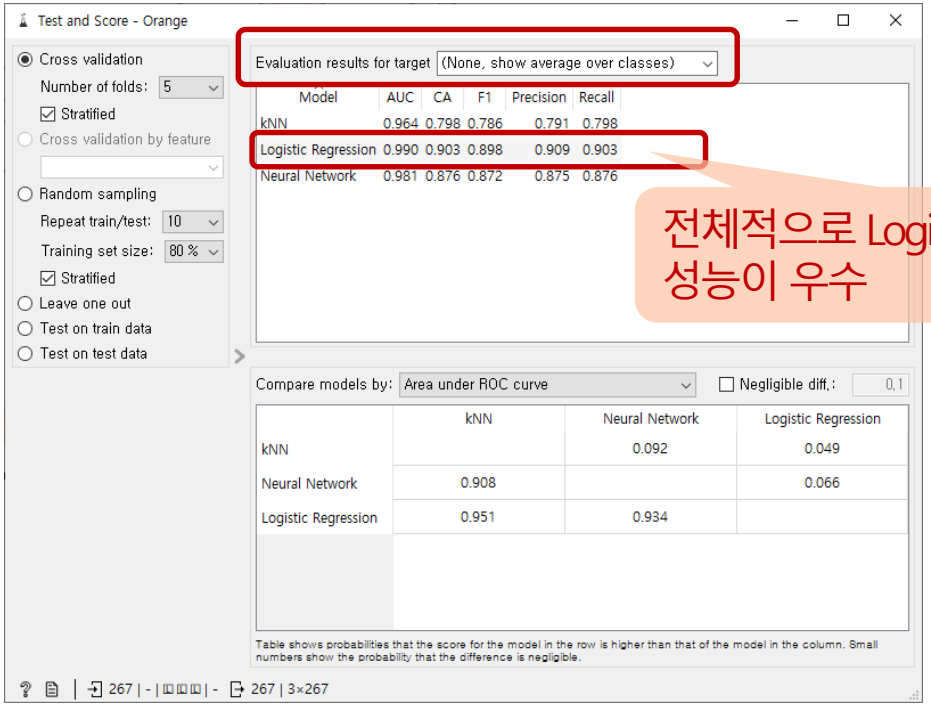


그림 5-7 성능 평가 결과표

② 원하는 통계값 확인하기

- Target Class와 Model Comparison을 변경하여 원하는 통계값을 확인할 수 있다.
- [그림 5-7]의 성능 평가 결과표에서는 전체적으로 Logistic Regression이 우수했으나, [그림 5-8]과 같이 Target Class를 paper로 놓고 보았을 때 Neural Network의 성능이 더 우수하다는 점을 확인할 수 있다.

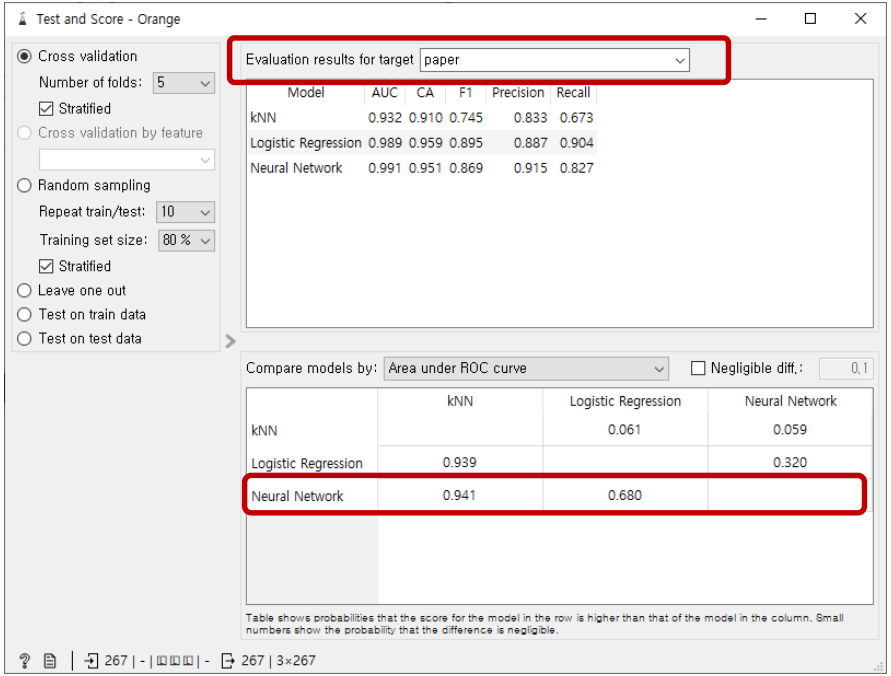
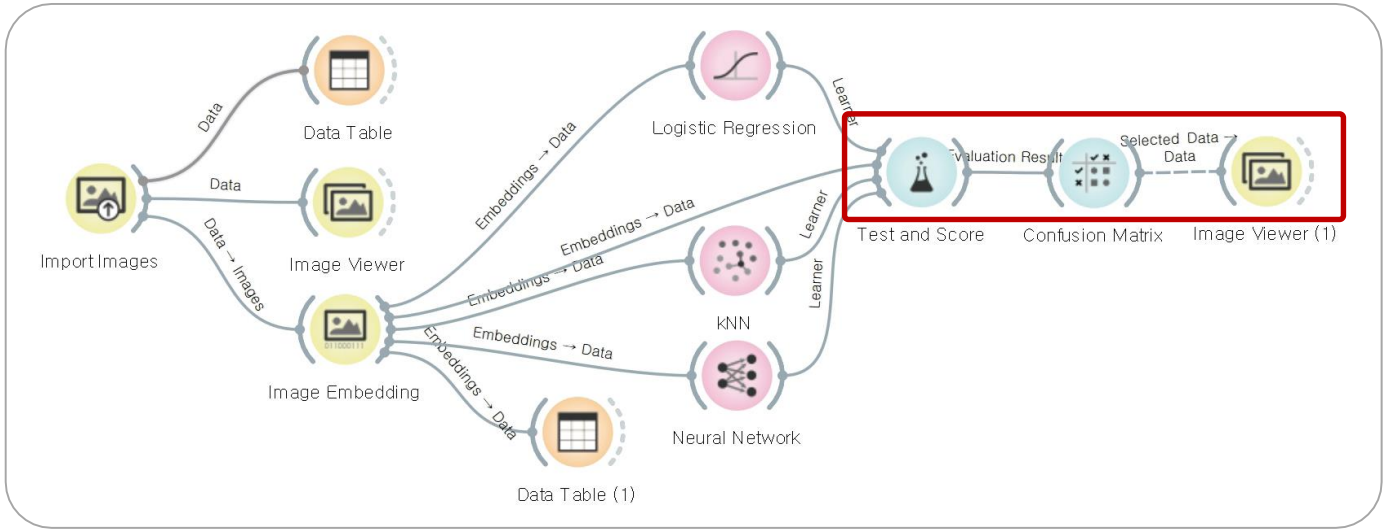


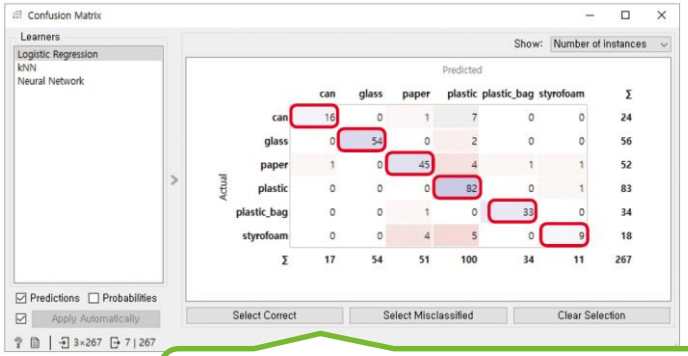
그림 5-8 Target Class와 Model Comparison 변경 후 통계값

③ 혼동 행렬과 Image Viewer로 결과 분석하기

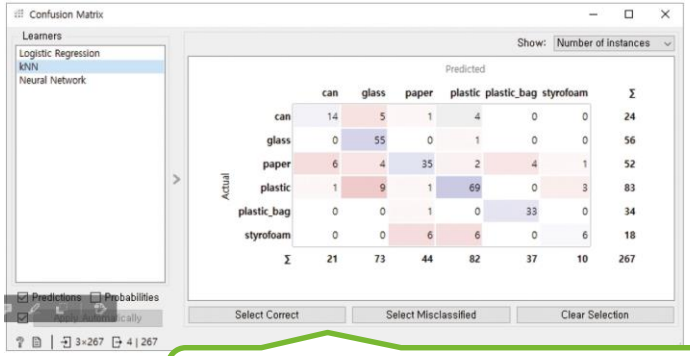
- Evaluate 카테고리의 **[Confusion Matrix]** 위젯을 가져온 후, **[Test and Score]** 위젯에 연결
- Image Analytics 카테고리의 **[Image Viewer]** 위젯을 새로 가져와서 **[Confusion Matrix]** 위젯과 연결



- [Confusion Matrix] 위젯을 더블 클릭하여 3개 모델의 혼동 행렬 비교

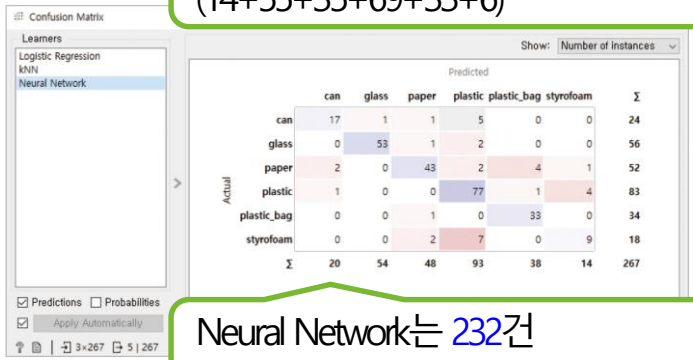


Logistic Regression 모델이 239건
(16+54+45+82+33+9)



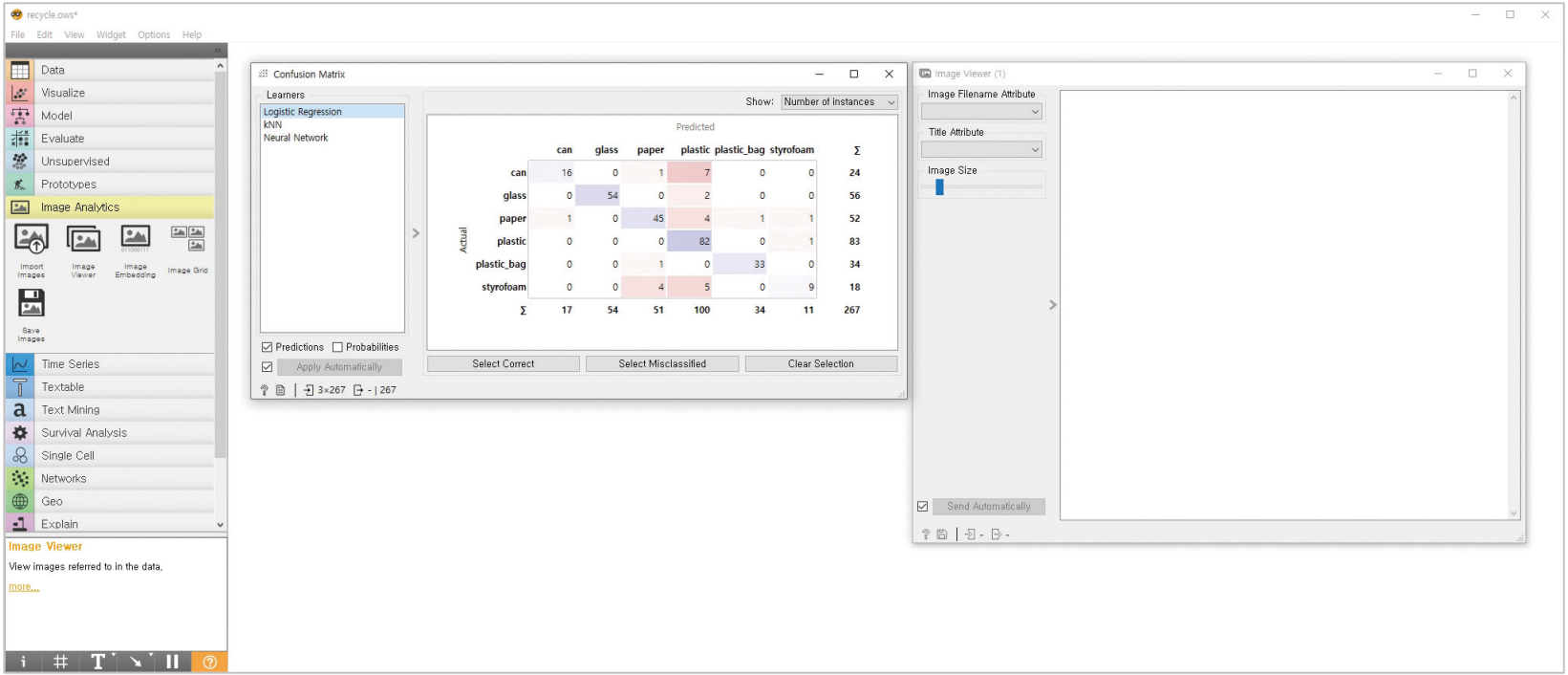
kNN 모델이 212건
(14+55+35+69+33+6)

이 활동에서는 Logistic Regression이 데이터를 가장 잘 분류하였다.

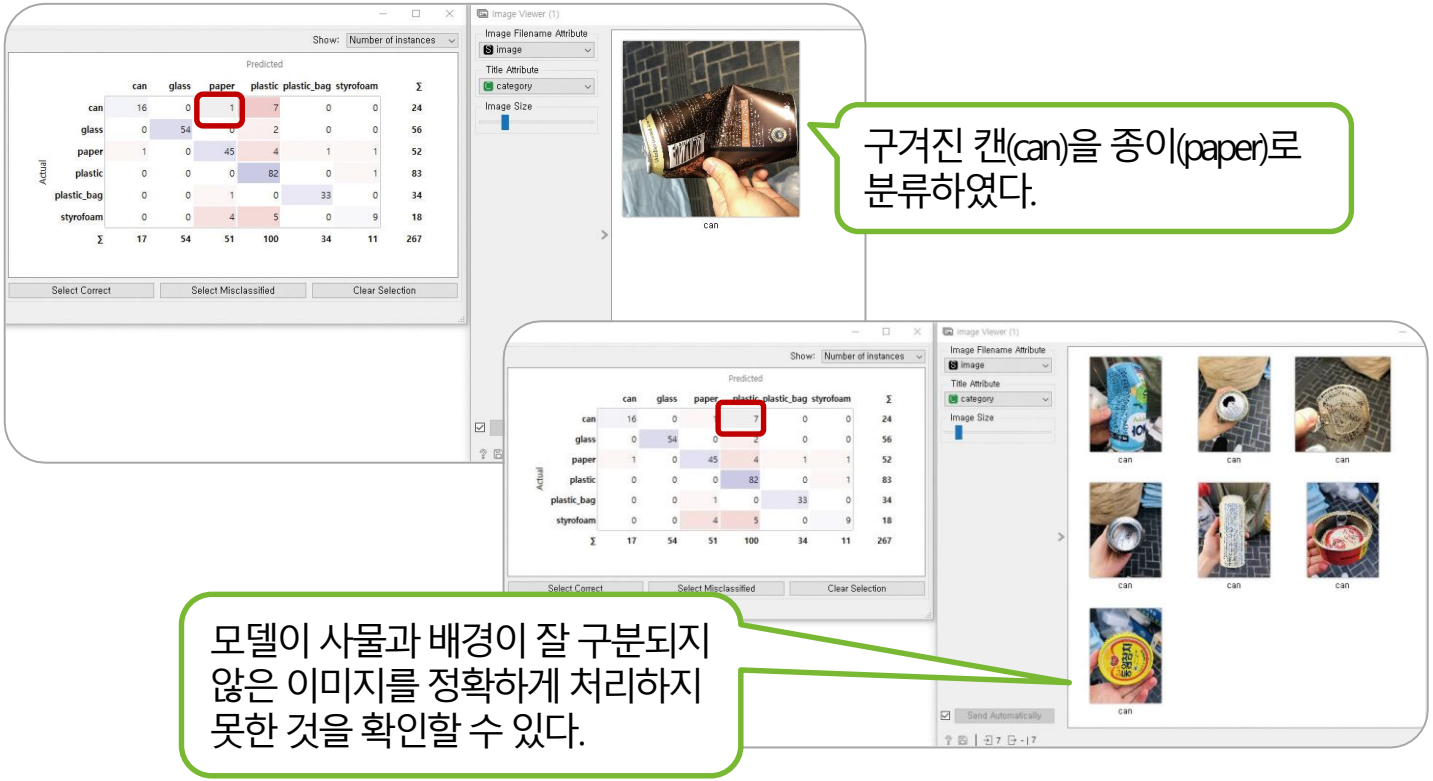


Neural Network는 232건
(17+53+43+77+33+9)

- Confusion Matrix 창을 열어 둔 상태에서 [Image Viewer (1)] 위젯을 더블 클릭하여 다음과 같이 나란히 둔다.



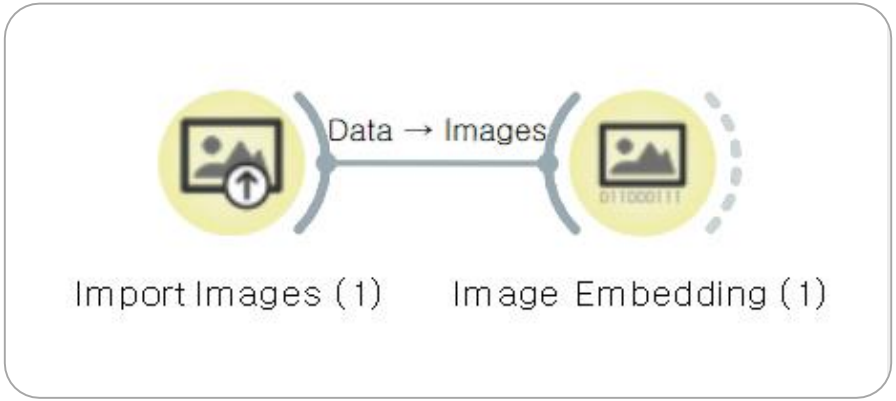
- 혼동 행렬에서 잘못 분류한 경우를 클릭하면 Image Viewer에서 해당 이미지를 볼 수 있다.



2 성능 결과 확인하기

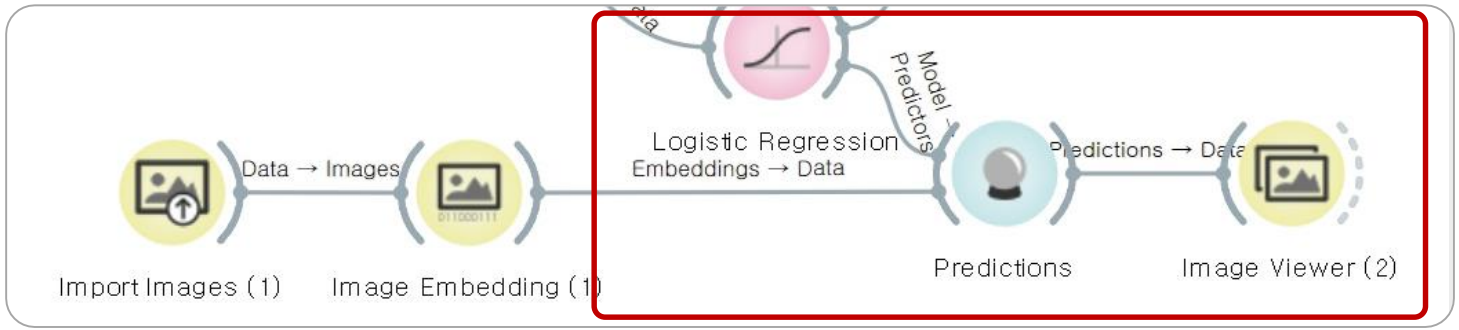
① 테스트 데이터 불러오기

- Image Analytics 카테고리의 [Import Image] 위젯을 새로 가져와 **test** 폴더를 불러오기
- 86~87쪽에서 이미지 데이터를 임베딩한 것과 마찬가지로 test 폴더의 이미지를 임베딩 처리하기



② 분류하기

- Evaluate 카테고리의 [Predictions] 위젯을 가져와 3개의 모델 중 성능이 우수했던 [Logistic Regression] 위젯에 연결
- Image Viewer 창에서 예측한 결과를 확인하기 위해 [Image Viewer] 위젯을 새로 가져와 [Predictions] 위젯에 연결



- Predictions 창과 Image Viewer 창을 나란히 두고 분류 결과 확인
- [그림 5-9]에서 모델이 15개의 이미지 데이터 중 3번 파일(IMG_8429.jpg)을 제외한 14개의 이미지를 정확하게 분류한 것을 확인 가능

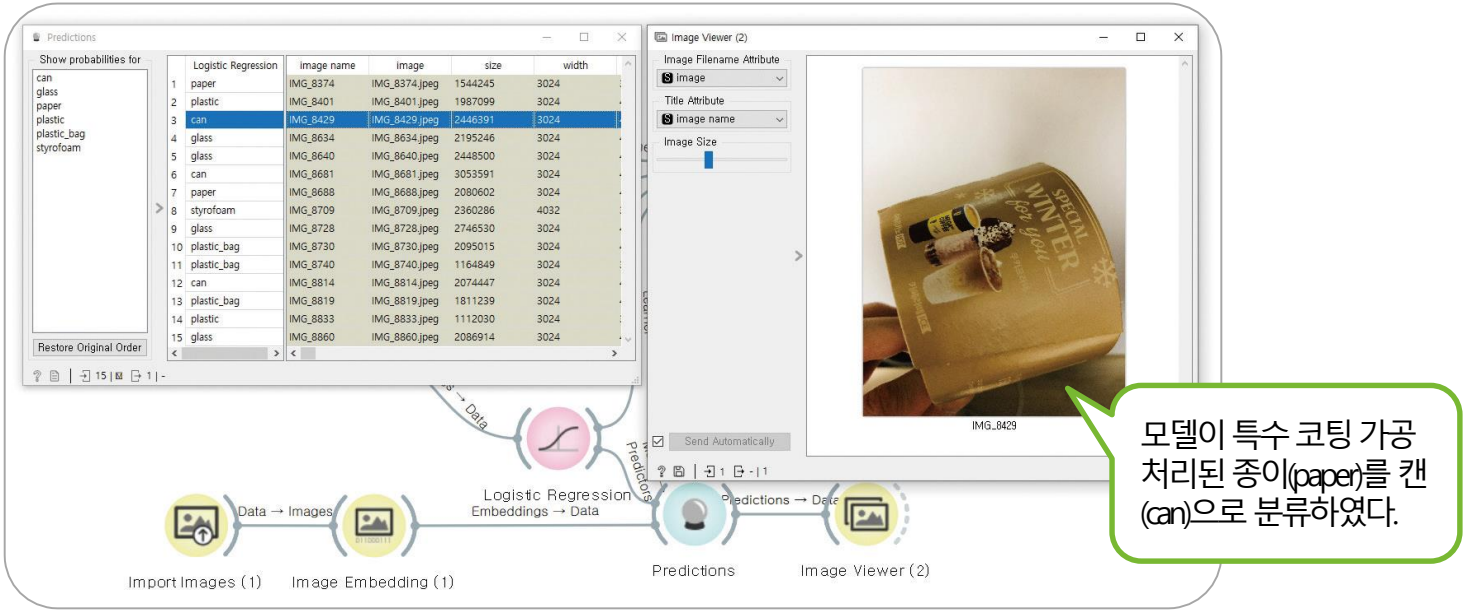


그림 5-9 모델의 테스트 데이터 분류 결과

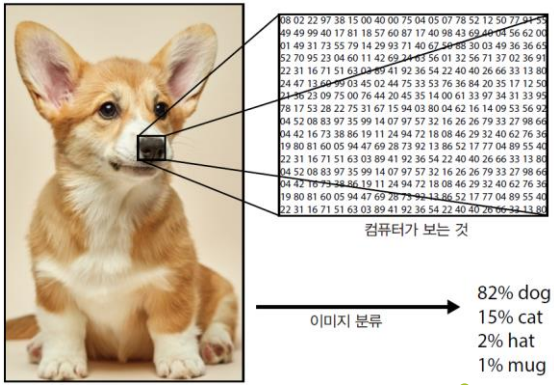
AI 전문가 되기

이미지 분류(Image Classification)

이미지 분류 문제란 주어진 이미지의 레이블이 무엇인지, 각 레이블에 대한 확신도(confidence)의 분포(distribution)가 어떤지를 예측하는 것이다. 이미지는 크기 폭(width)×높이(height)×3에 원소로 0~255 사이 정수를 가지는 3차원 배열이다. 여기서 3은 R, G, B 세 개의 색 채널(channel)을 의미한다.

즉, 우리는 이미지 데이터를 하나의 정지된 그림으로 보지만, 기계는 이를 연산 가능한 형태로 만드는 임베딩 과정을 거쳐서 인식한다.

이미지를 인식하는 것은 사람에게는 쉬운 일이지만 기계에게는 보는 각도, 크기의 변동, 변형, 가려짐, 조명, 배경과 섞임등의 이유로 어렵다.



교재 92페이지 참조

AI랑 친해지기

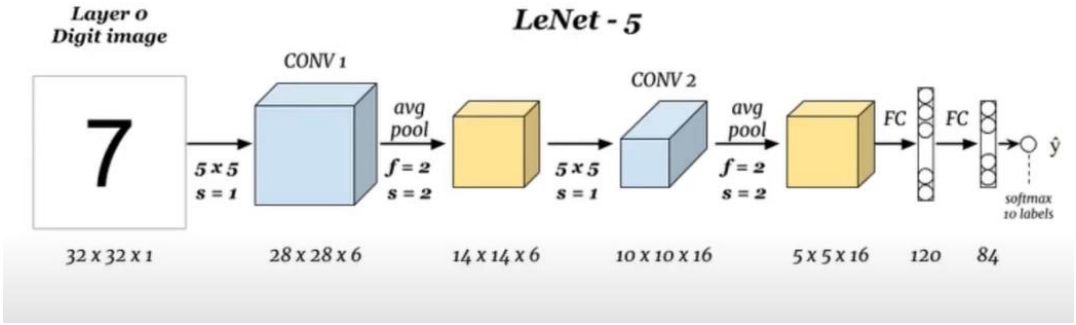
Convolutional Neural Network

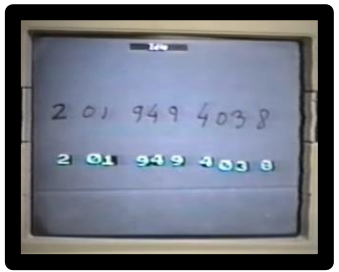
- 합성곱 신경망
  - 다층 퍼셉트론(MLP; Multi Layer Perceptron) 구조로 설계되며 이미지 인식 및 분류, 의료 영상 처리, 자율 주행 자동차 등의 **컴퓨터 비전(Computer Vision)** 분야에 주로 사용

AI랑 친해지기

Convolutional Neural Network

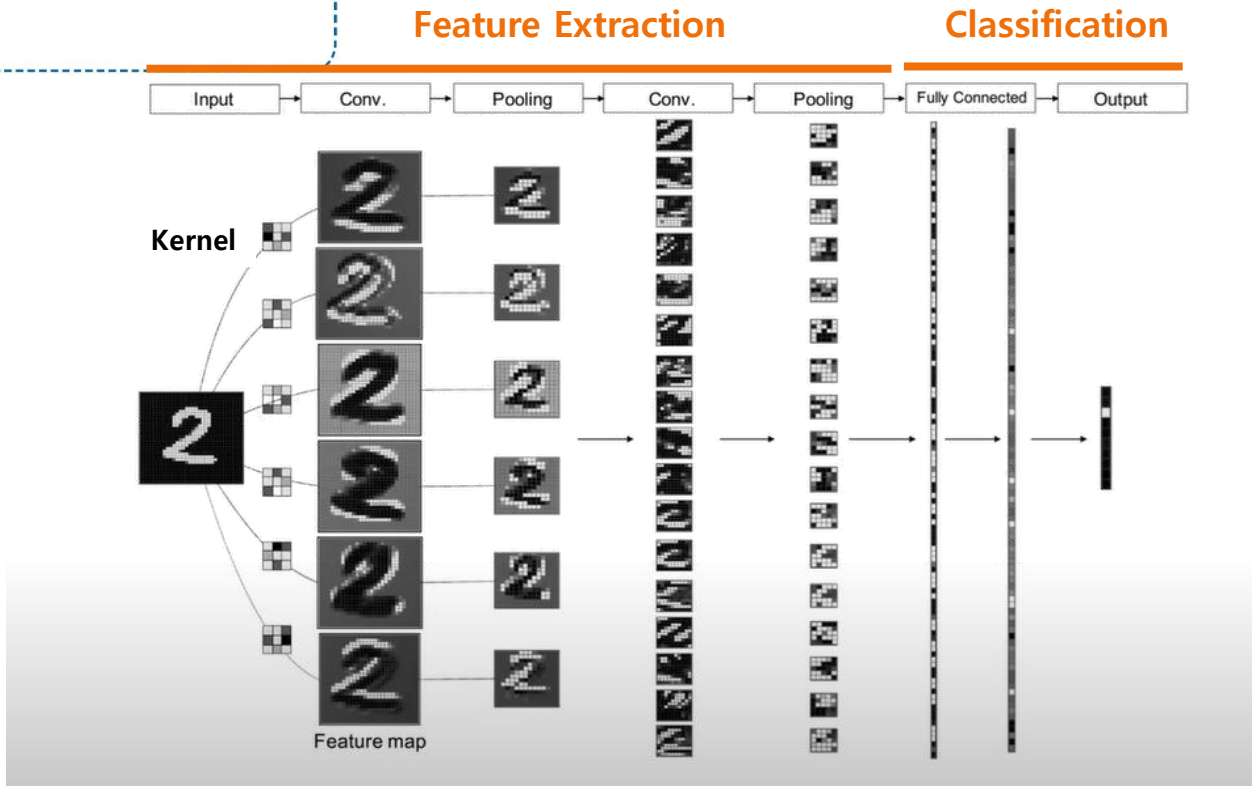
- **Yann LeCun(얀 르쿤)**
  - 1989년, 미국의 통신 회사인 AT&T의 벨 연구소에 있을 때 **Convolutional Neural Network** 발표
  - LeNet5가 은행에서 수표에 적힌 숫자 인식에 사용





AI랑 친해지기

Convolutional Neural Network



Convolutional Neural Network (Feature Extraction)

AI랑 친해지기

1 Convolution Layer

- 입력된 이미지에서 특징을 추출하기 위해 Kernel을 활용

1	0	1	0
0	1	1	0
0	0	1	1
0	0	1	0

4×4 Input Image

예) 2×2 Kernel의 경우

- ✓ 각 셀에는 가중치가 들어있음
- ✓ 가중치를 ×1, ×0 라고 가정

×1	×0
×0	×1

2×2 Kernel

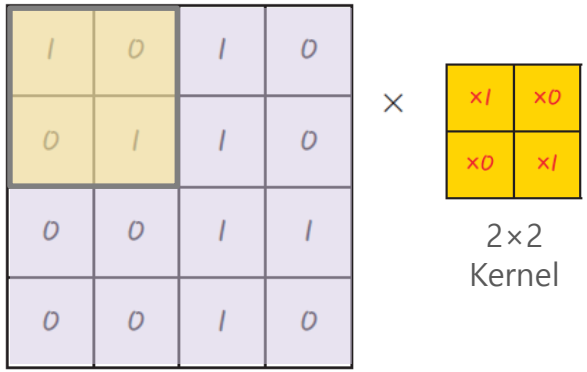
AI랑 친해지기

Convolutional Neural Network (Feature Extraction)

- 4×4 Input Image와 2×2 Kernel을 Convolution

✓ Input Image에 Kernel의 가중치의 값을 각각 곱하고, 곱한 결과를 더하면 2가 됨

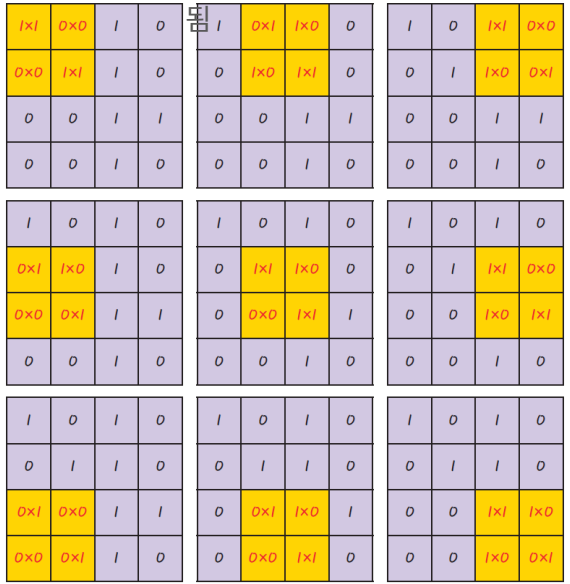
$$(1\times 1)+(0\times 0)+(0\times 0)+(1\times 1)=2$$



4×4 Input Image

stride = 1

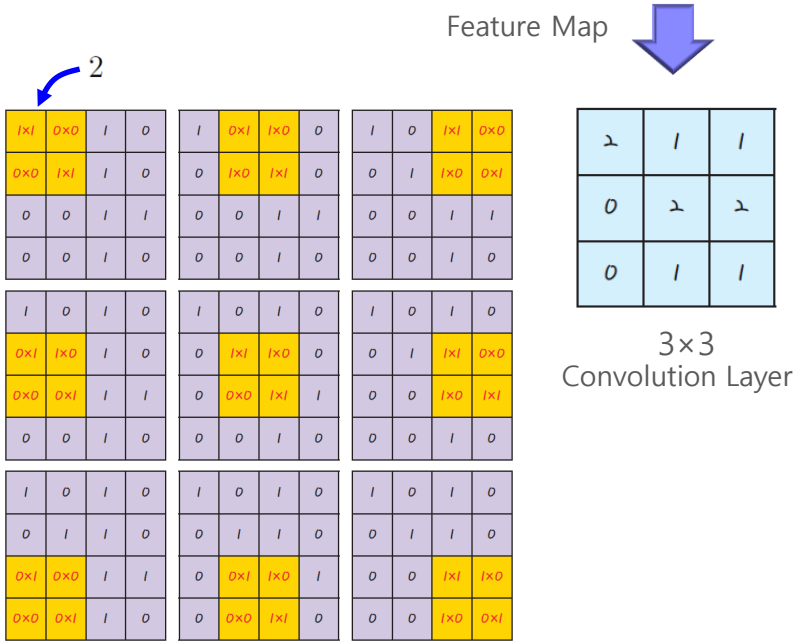
Kernel이 한 칸씩 이동하면서 적용됨



AI랑 친해지기

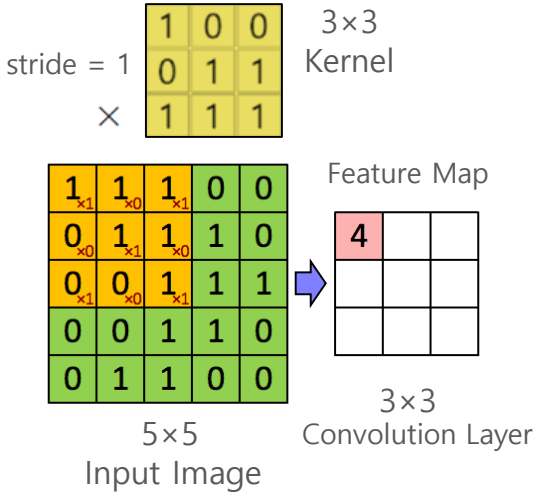
Convolutional Neural Network (Feature Extraction)

- 4×4 Input Image와 2×2 Kernel을 Convolution



<참고>

- 5×5 Input Image와 3×3 Kernel을 Convolution

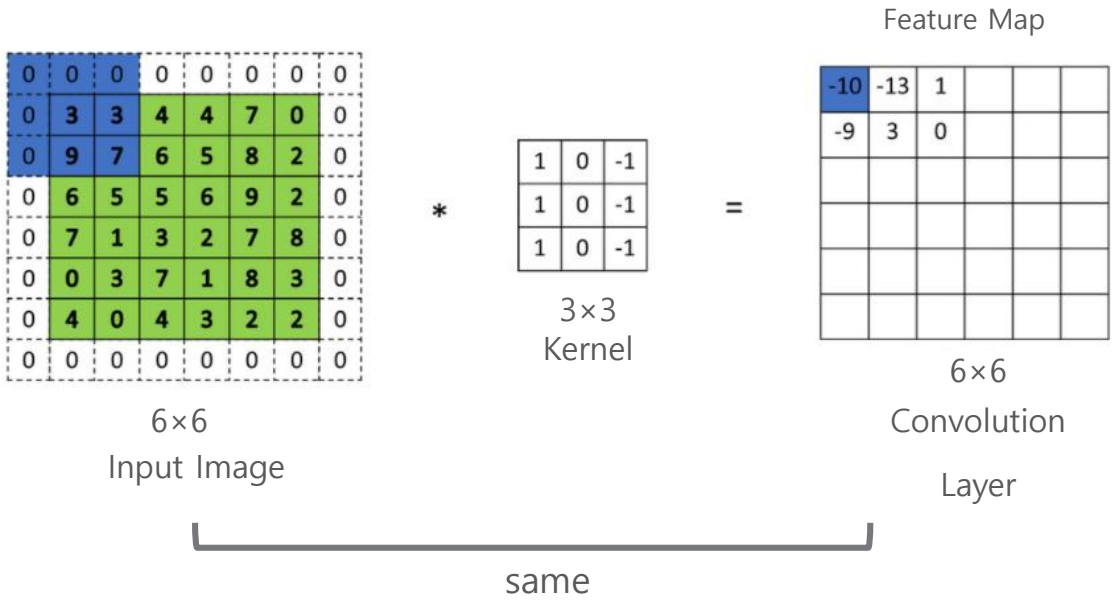


AI랑 친해지기

Convolutional Neural Network (Feature Extraction)

- **Zero Padding**

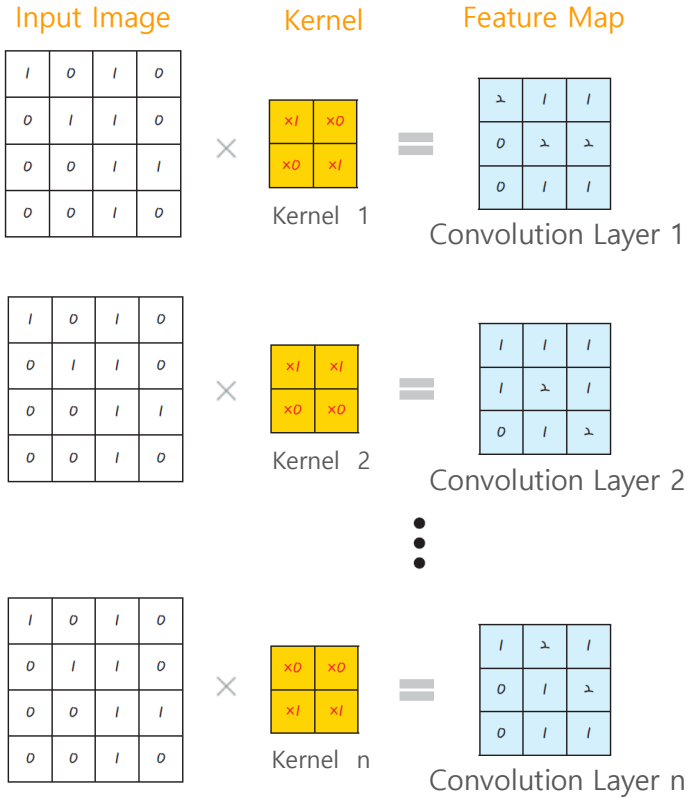
✓ Feature Map의 사이즈를 Input Image의 사이즈와 동일하게 생성



AI랑 친해지기

Convolutional Neural Network (Feature Extraction)

- **다양한 패턴의 Kernel을 여러 개 사용**
  - ✓ 여러 개의 Convolution된 이미지가 만들어져서 **다양한 Feature Map을 획득**할 수 있으므로 더욱 **정교한 특징**을 추출할 수 있다.



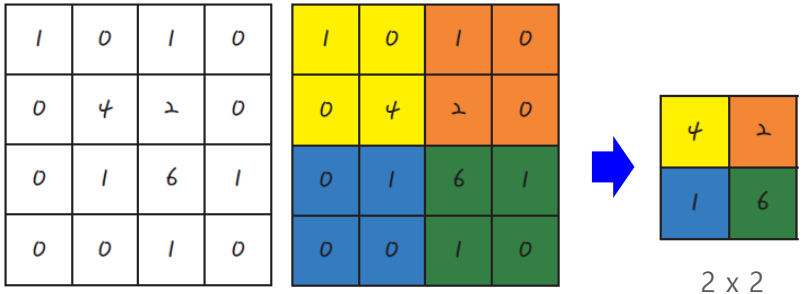
AI랑 친해지기

Convolutional Neural Network (Feature Extraction)

2 Pooling

- Convolution Layer의 **데이터의 크기를 축소**시키면서 이미지의 **핵심 특징**을 찾는 방식
  - ✓ **Max Pooling** : 정해진 구역 안에서 최대값을 추출하는 방식
  - ✓ **Average Pooling** : 정해진 구역 안에서 평균값을 추출하는 방식

예) 2×2 Max Pooling

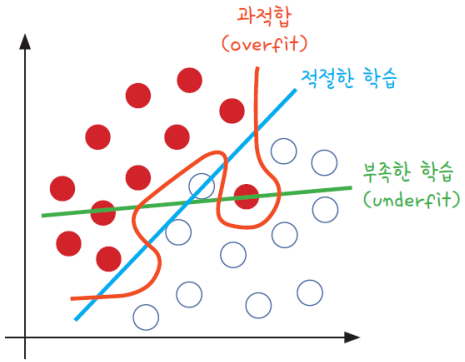


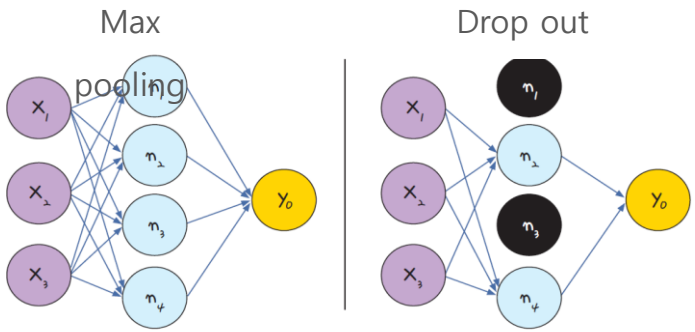
AI랑 친해지기

Convolutional Neural Network (Feature Extraction)

3 Drop out

- 은닉층에 배치된 노드 중 일부의 노드를 신경망에서 차단시키는 것으로 Overfitting 해결하기 위해서 사용



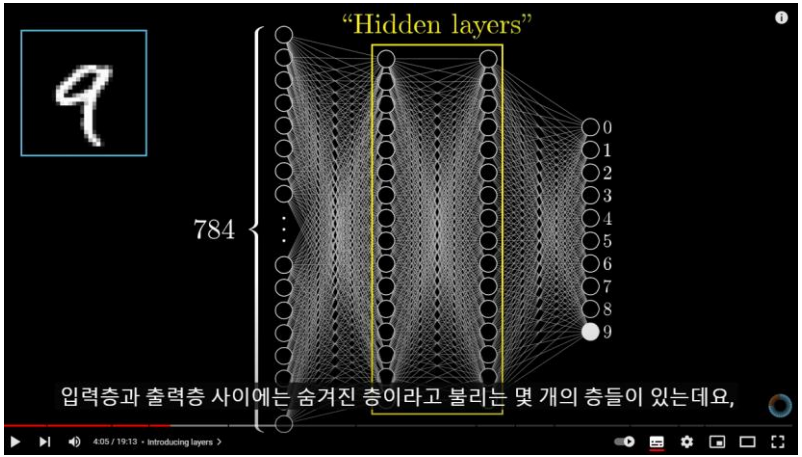


AI랑 친해지기

Convolutional Neural Network

[참고 영상]

https://www.youtube.com/watch?v=aircAruvnKk&t=237s



정리하기

지금까지 재활용품을 분류할 수 있는 인공지능 모델을 만들어 보았다.
이러한 분류 모델을 실생활에 적용한다면 재활용이 번거롭다고 생각하는 사람들의 인식을 바꿈으로써 재활용률이 증가할 것이고, 재활용품을 품목별로 바르게 분류함으로써 재분리하거나 처리하는 데 부담해야 했던 사회적 비용이 감소할 것이다.
하지만 사람들이 버리는 재활용품의 종류는 수천 수만 가지이며,
형태나 질감 또한 다양하기 때문에 우선 방대하고 체계적인 데이터 셋이
구축되어야 할 것이다.

Q & A