

재활용품 분류하기



소프트웨어융합대학원
진혜진

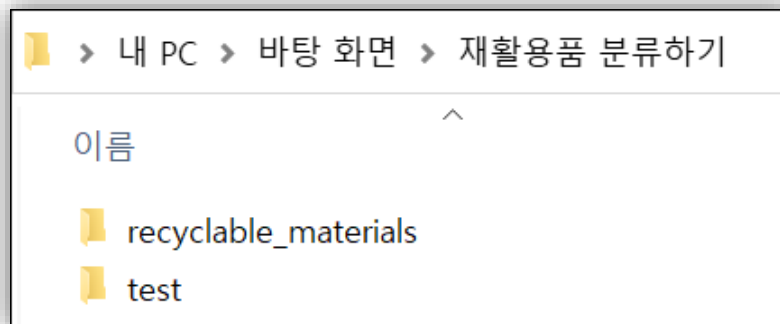
■ 해결해야 할 문제

- 환경부 조사에 따르면 우리나라 국민 한 사람이 70년간 배출하는 생활 쓰레기는 무려 55톤에 달한다고 한다.
- 올바른 재활용 분리 배출은 문제해결의 첫 걸음이 될 수 있을 뿐만 아니라 지속 가능한 자원 순환의 시작이 될 것이다.
- 인공지능을 활용하여 효율적으로 분리수거를 할 수는 없을까?
- 재활용품 이미지 데이터를 분석하고 학습한 후, 효율적으로 재활용품을 분리할 수 있는 인공지능 모델을 만들어 보자.

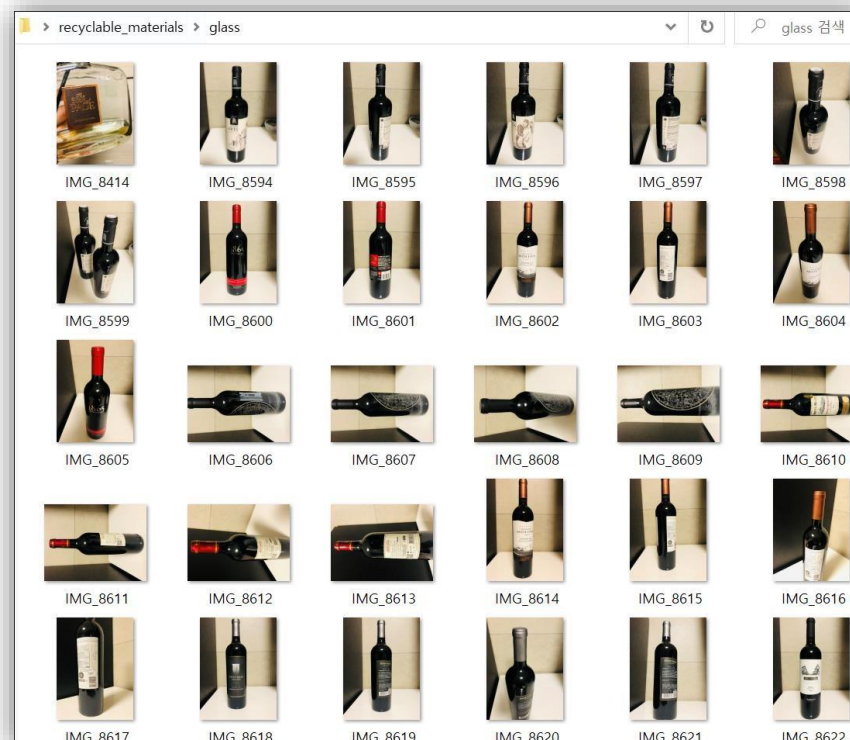
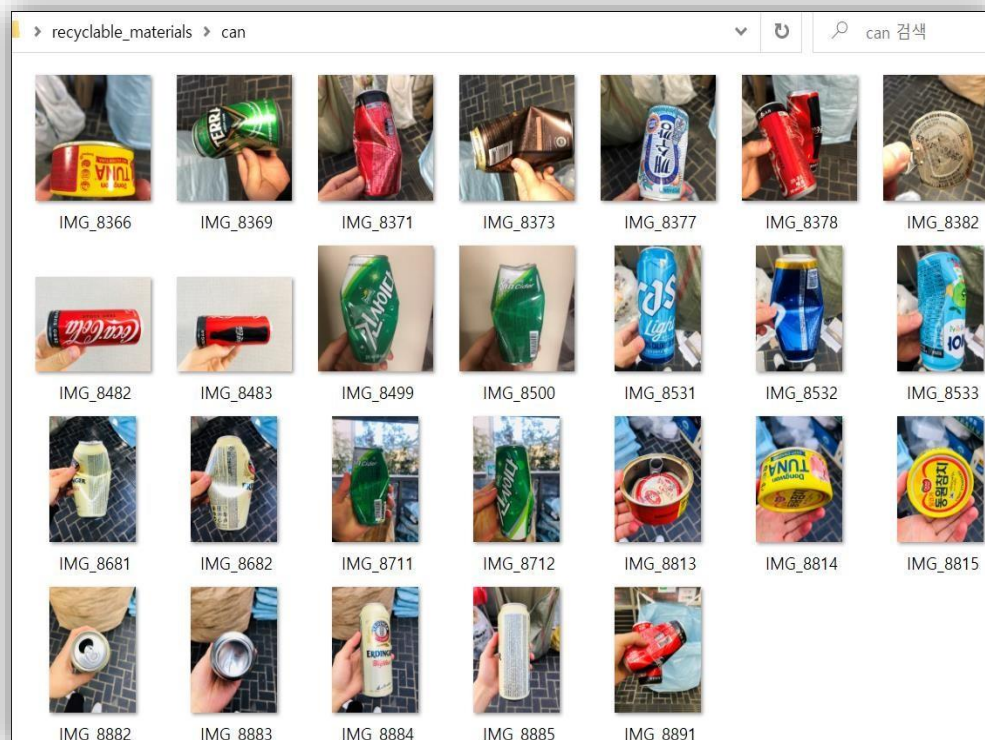
■ 데이터 준비하기

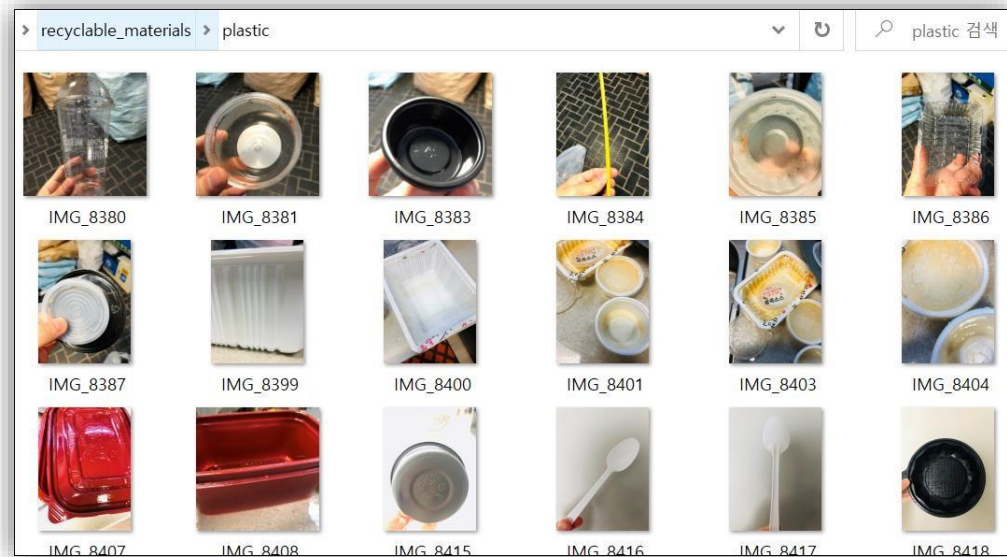
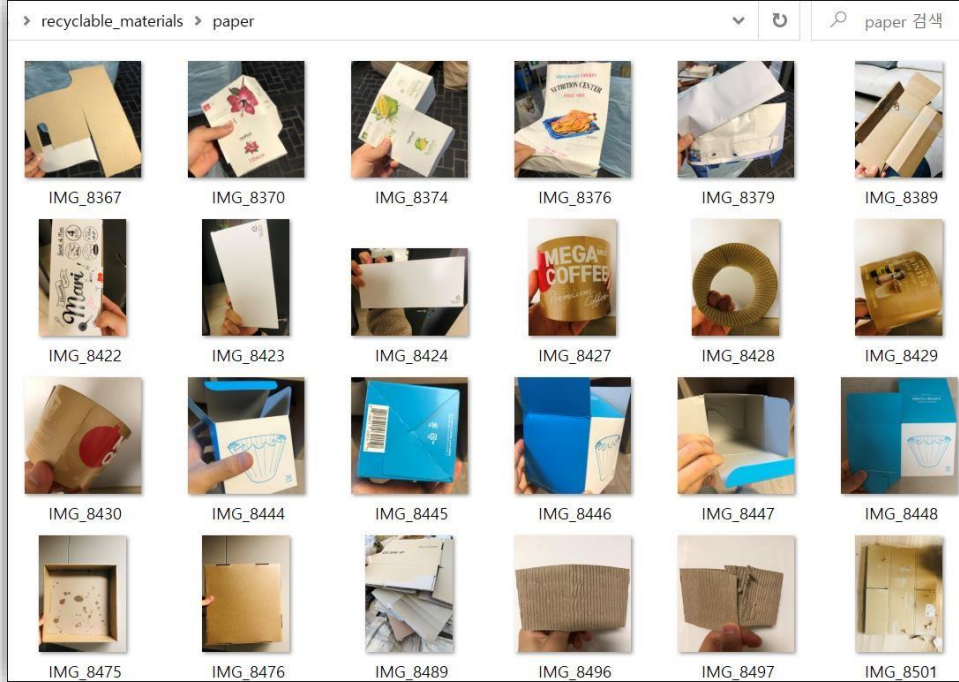
■ 데이터

- 재활용품 분류하기.zip 파일 다운로드



- Orange3에서 이미지 데이터를 모델에 학습시켜 분류나 예측에 활용할 때 폴더명이 데이터의 레이블, 즉 정답이 된다.

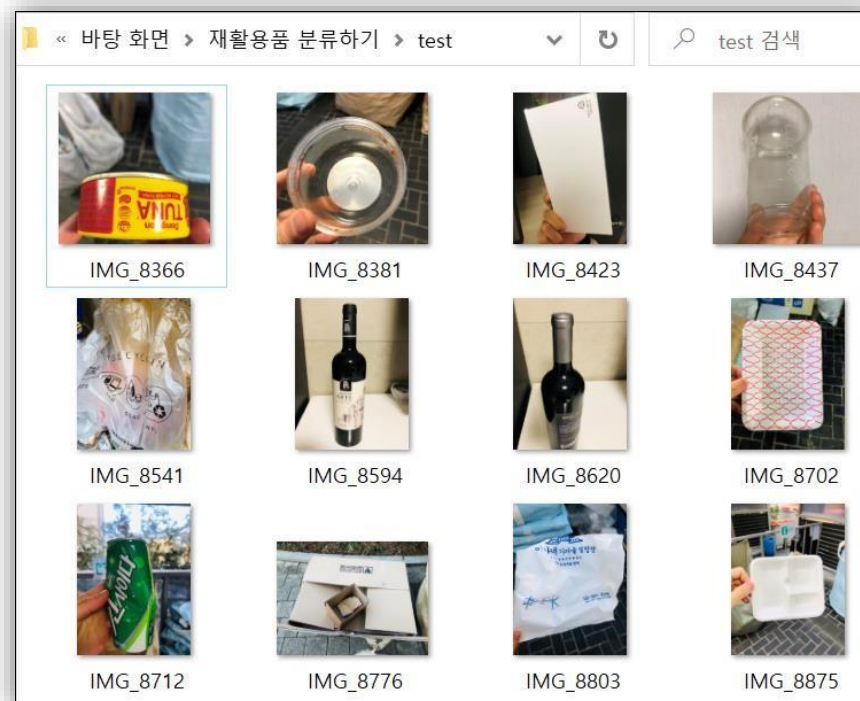
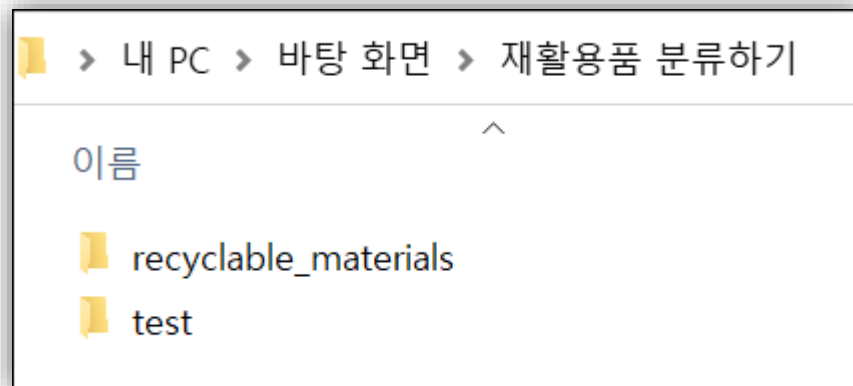




- 이미지 개수(총 282개)
 - can : 26개
 - glass : 60개
 - paper : 55개
 - plastic : 85개
 - plastic_bag : 37개
 - Styrofoam : 19개

■ 훈련 데이터와 테스트 데이터 나누기

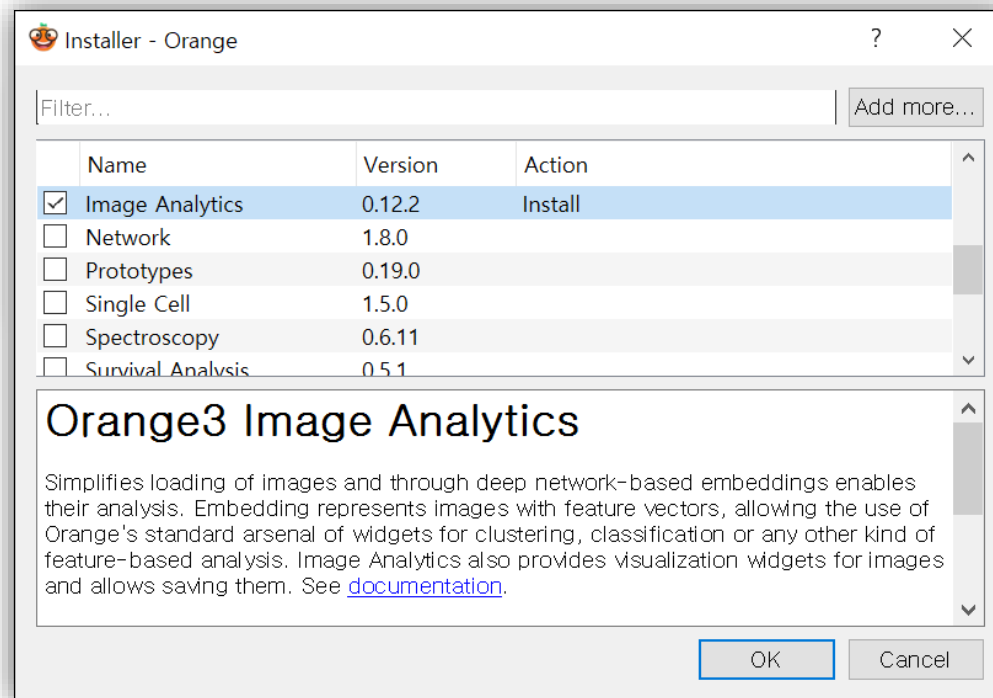
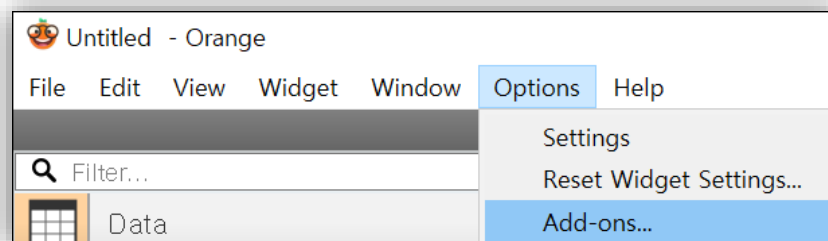
- 이미지 학습을 마친 인공지능 모델을 테스트에 활용하기 위해 test 폴더를 새로 만들고
- 다운로드한 이미지 파일의 일부를 임의로 몇 가지 선택하여 test 폴더로 옮긴다.



■ 데이터 불러오기

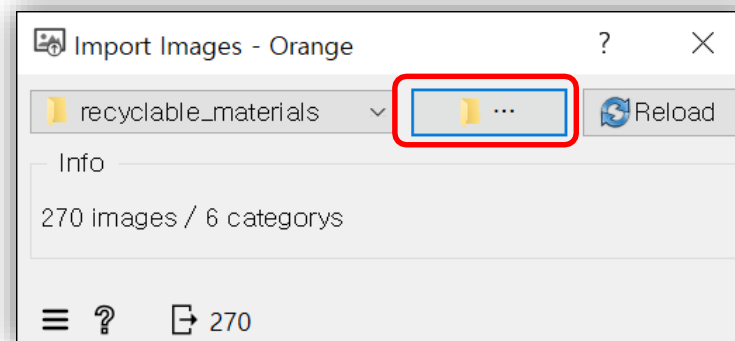
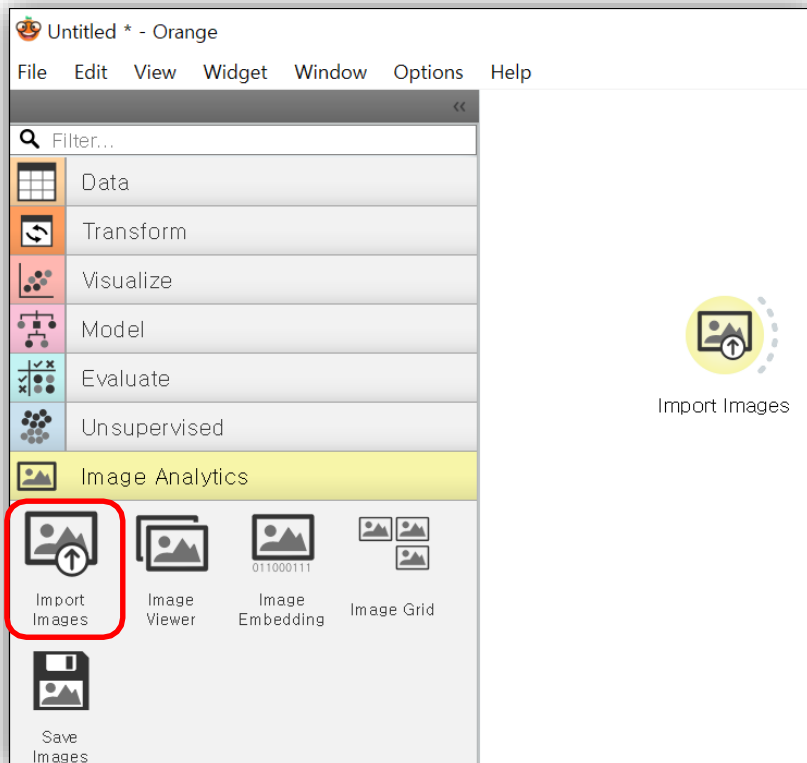
■ 카테고리 기능 추가하기

- 이미지를 분석하기 위해 이미지 분석과 관련된 위젯으로 구성되어 있는 Image Analytics 카테고리를 설치한다.

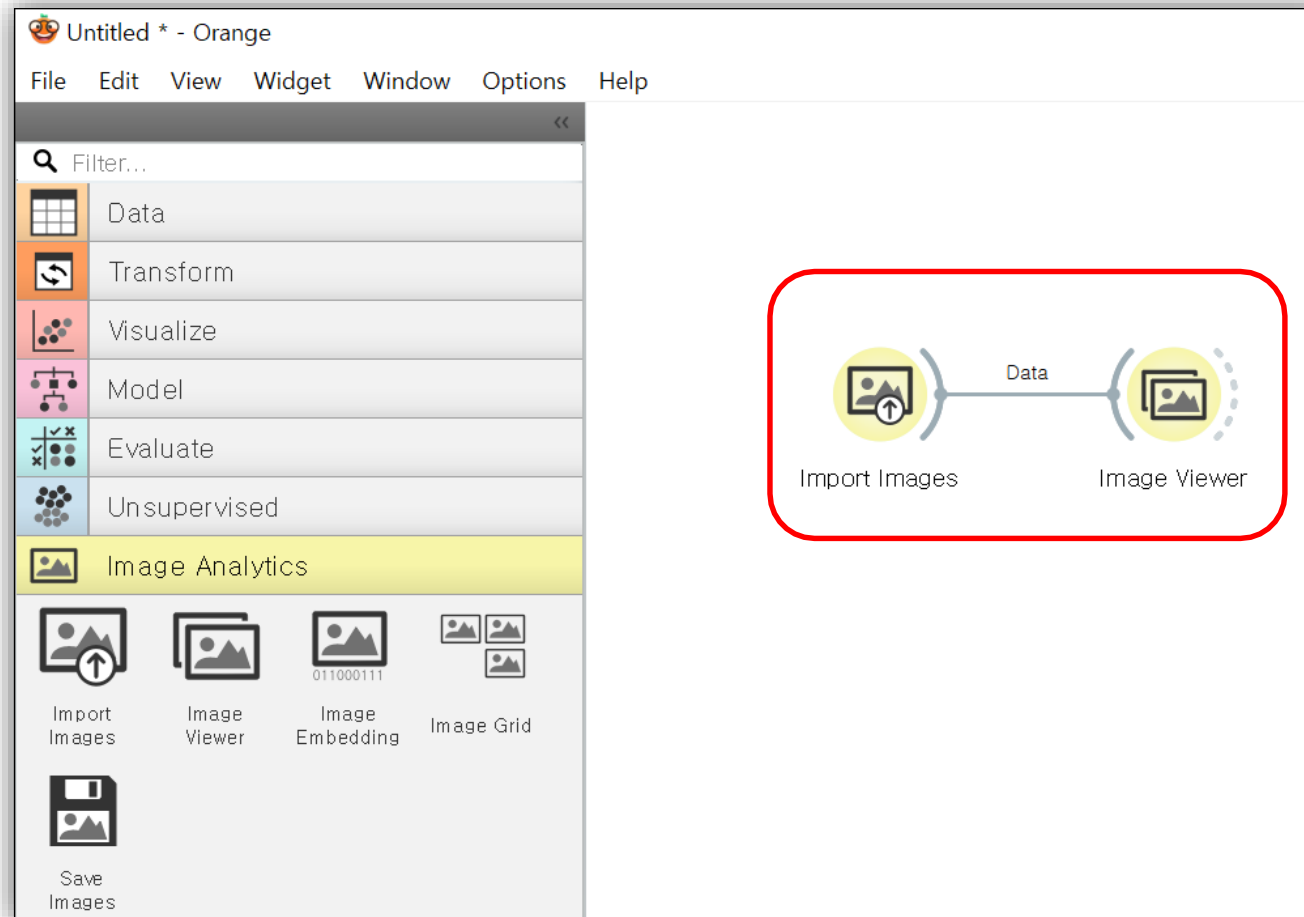


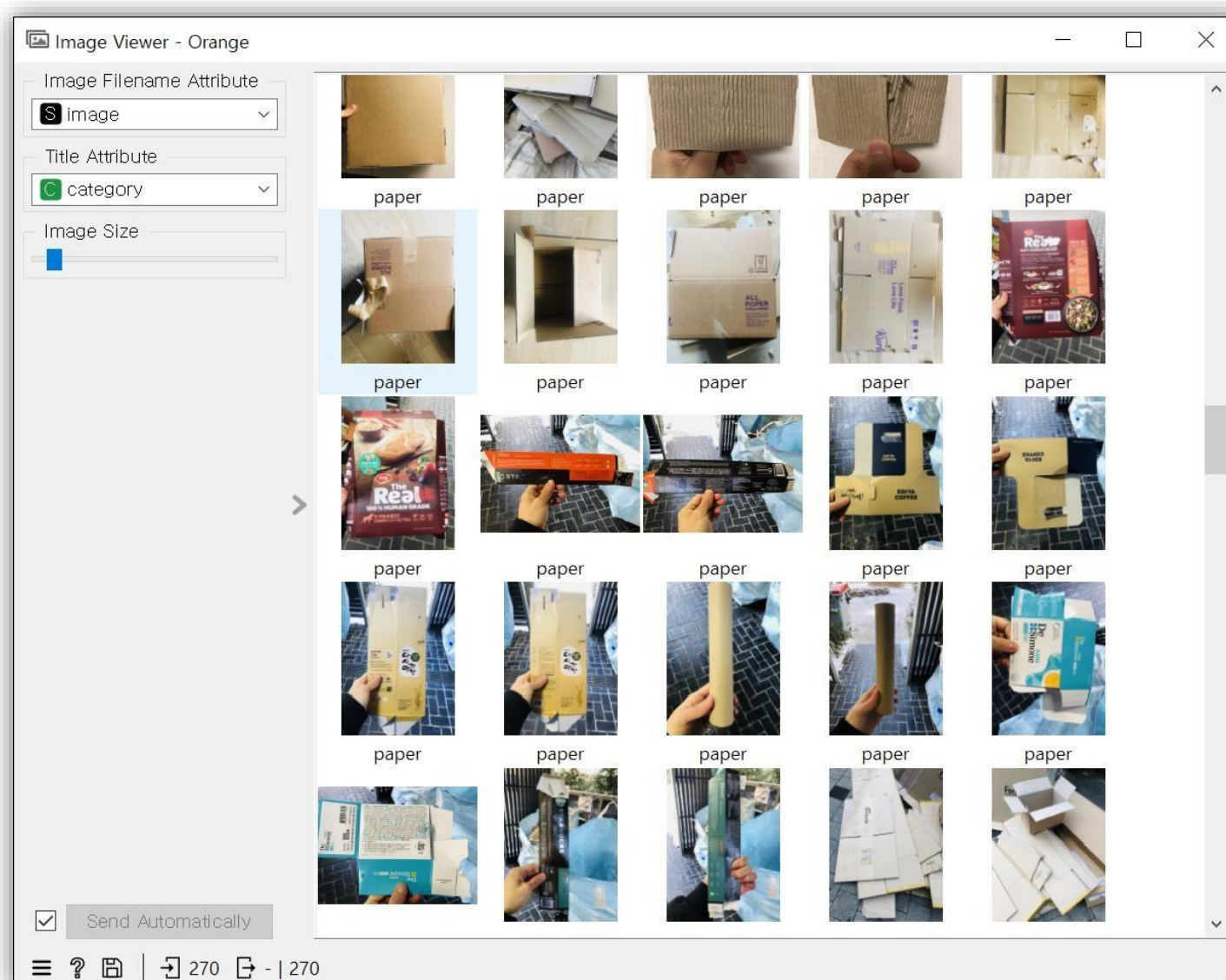


- Image Analytics 카테고리의 [Import Images] 위젯을 가져와서 더블 클릭한 후, recyclable_materials 폴더를 선택한다.
- [Import Images] 위젯은 폴더 안의 이미지 데이터를 한꺼번에 가져올 수 있는 기능을 제공한다.



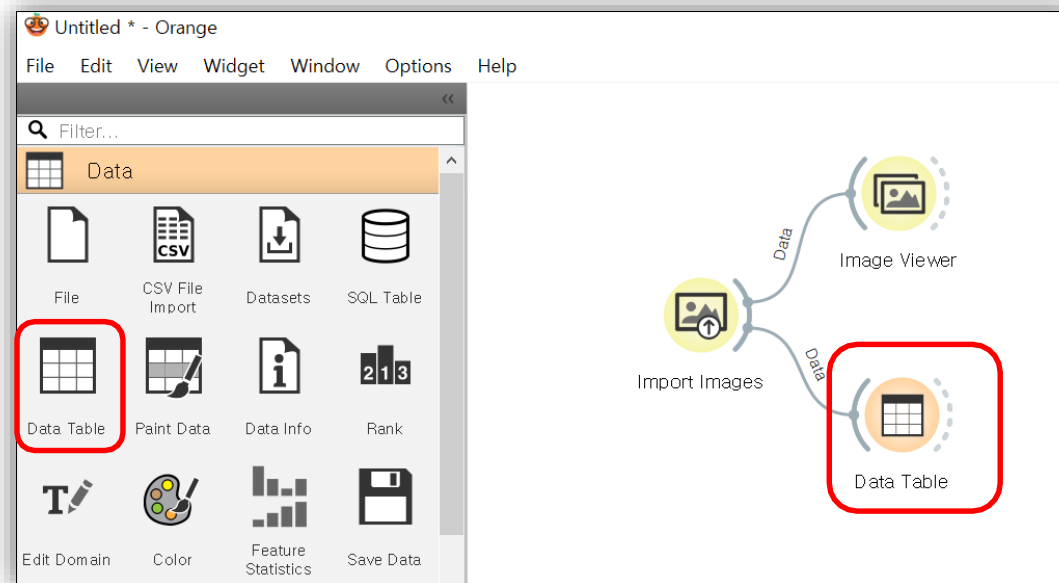
- [Image Viewer] 위젯을 가져와서 [Import Images] 위젯과 연결하면 이
미지를 확인할 수 있다.





■ 데이터 속성 확인하기

- Data 카테고리의 [Data Table] 위젯을 가져와서 [Import Images] 위젯과 연결한다.



- 이미지의 category는 타겟으로 사용되고, 파일명, 파일 경로, 크기, 너비, 높이는 이미지에 대한 사전 정보이다.
- 이미지 데이터를 불러왔지만 속성을 파악할 수 없기 때문에 이미지의 특징을 기반으로 분류 작업을 수행할 수 없다.

Data Table - Orange

Info

270 instances (no missing data)

0 features

Target with 6 values

5 meta attributes

Variables

☒ Show variable labels (if present)

☐ Visualize numeric values

☒ Color by instance classes

Selection

☒ Select full rows

Restore Original Order

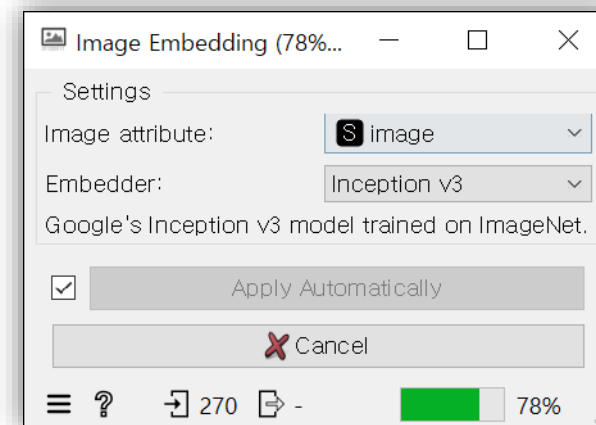
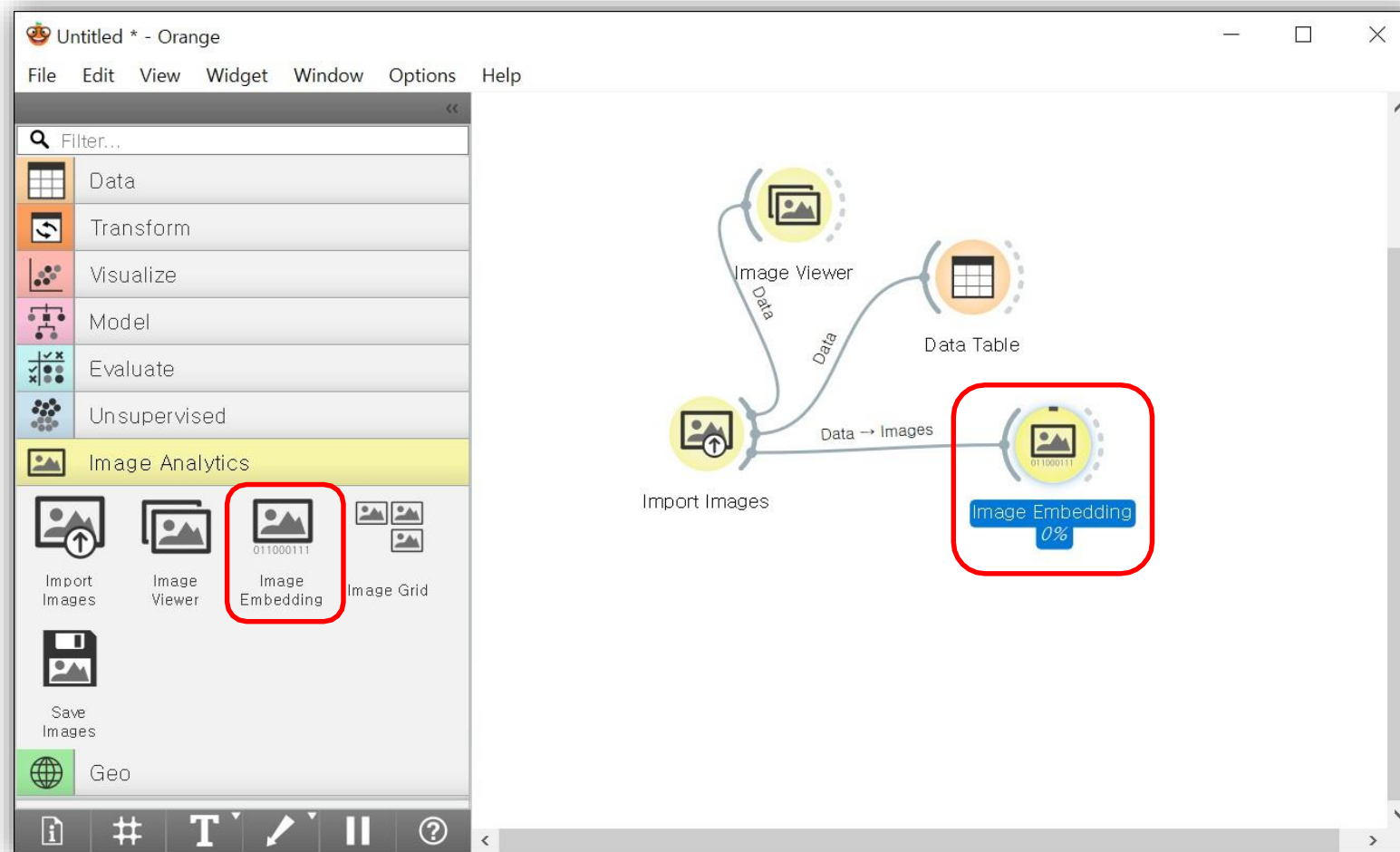
☒ Send Automatically

| origin | category | image name | image I-당 화면/재활용품 분류 image | size | width | height |
|--------|----------|------------|----------------------------------|---------|-------|--------|
| 1 | can | IMG_8369 | can/IMG_8369.jpeg | 1576770 | 3024 | 3024 |
| 2 | can | IMG_8371 | can/IMG_8371.jpeg | 1919971 | 3024 | 3024 |
| 3 | can | IMG_8373 | can/IMG_8373.jpeg | 1902636 | 3024 | 3024 |
| 4 | can | IMG_8377 | can/IMG_8377.jpeg | 2240434 | 3024 | 3024 |
| 5 | can | IMG_8378 | can/IMG_8378.jpeg | 1615744 | 3024 | 3024 |
| 6 | can | IMG_8382 | can/IMG_8382.jpeg | 2157664 | 3024 | 3024 |
| 7 | can | IMG_8482 | can/IMG_8482.jpeg | 1937148 | 4032 | 3024 |
| 8 | can | IMG_8483 | can/IMG_8483.jpeg | 2056189 | 4032 | 3024 |
| 9 | can | IMG_8499 | can/IMG_8499.jpeg | 1804697 | 3024 | 3024 |
| 10 | can | IMG_8500 | can/IMG_8500.jpeg | 1817021 | 3024 | 3024 |
| 11 | can | IMG_8531 | can/IMG_8531.jpeg | 2391319 | 3024 | 4032 |
| 12 | can | IMG_8532 | can/IMG_8532.jpeg | 2126186 | 3024 | 4032 |
| 13 | can | IMG_8533 | can/IMG_8533.jpeg | 2670794 | 3024 | 4032 |
| 14 | can | IMG_8681 | can/IMG_8681.jpeg | 3053591 | 3024 | 4032 |
| 15 | can | IMG_8682 | can/IMG_8682.jpeg | 3187521 | 3024 | 4032 |
| 16 | can | IMG_8711 | can/IMG_8711.jpeg | 2743896 | 3024 | 4032 |
| 17 | can | IMG_8813 | can/IMG_8813.jpeg | 2082354 | 3024 | 4032 |
| 18 | can | IMG_8814 | can/IMG_8814.jpeg | 2074447 | 3024 | 4032 |
| 19 | can | IMG_8815 | can/IMG_8815.jpeg | 2010842 | 3024 | 4032 |
| 20 | can | IMG_8882 | can/IMG_8882.jpeg | 1777759 | 3024 | 4032 |
| 21 | can | IMG_8883 | can/IMG_8883.jpeg | 1800457 | 3024 | 4032 |

270 | 270 | 270

■ 데이터 전처리

- 이미지 데이터는 비정형 데이터이기 때문에 기계학습에 적합한 형태로 바꿔야 한다.
- 이 과정을 이미지 임베딩이라고 하며 Orange3에서는 [Image Embedding] 위젯을 사용한다.



■ 임베딩한 이미지 데이터 속성 확인하기

- Data 카테고리의 [Data Table] 위젯을 가져와서 [Image Embedding] 위젯과 연결하여 데이터 속성을 확인해보자.
- 임베딩하기 전과 다르게 n0에서 n2047까지 총 2048개의 features가 추출되었으며, 추출된 features에 의해 기계학습이 가능해졌다.

Data Table (1) - Orange

Info
270 instances (no missing data)
2048 features
Target with 6 values
5 meta attributes

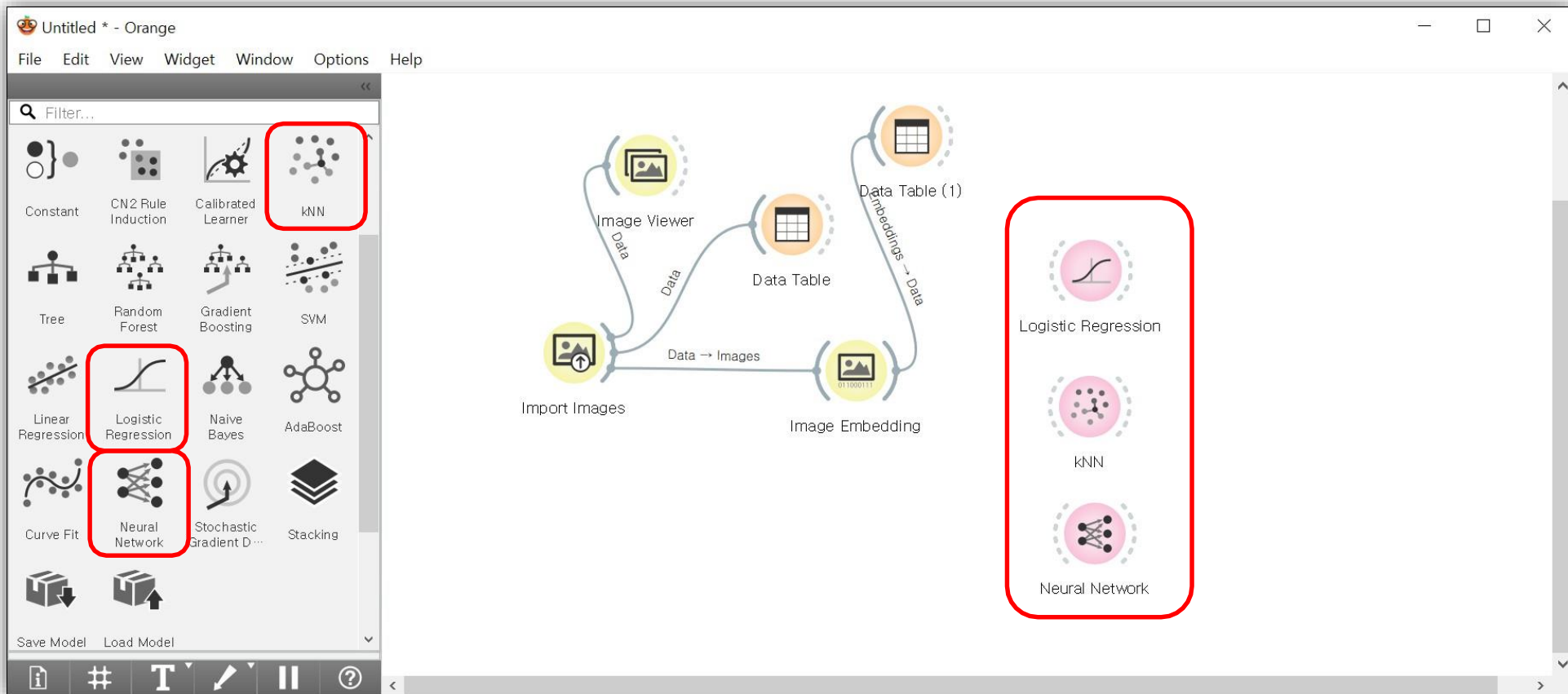
Variables
☒ Show variable labels (if present)
☐ Visualize numeric values
☒ Color by instance classes

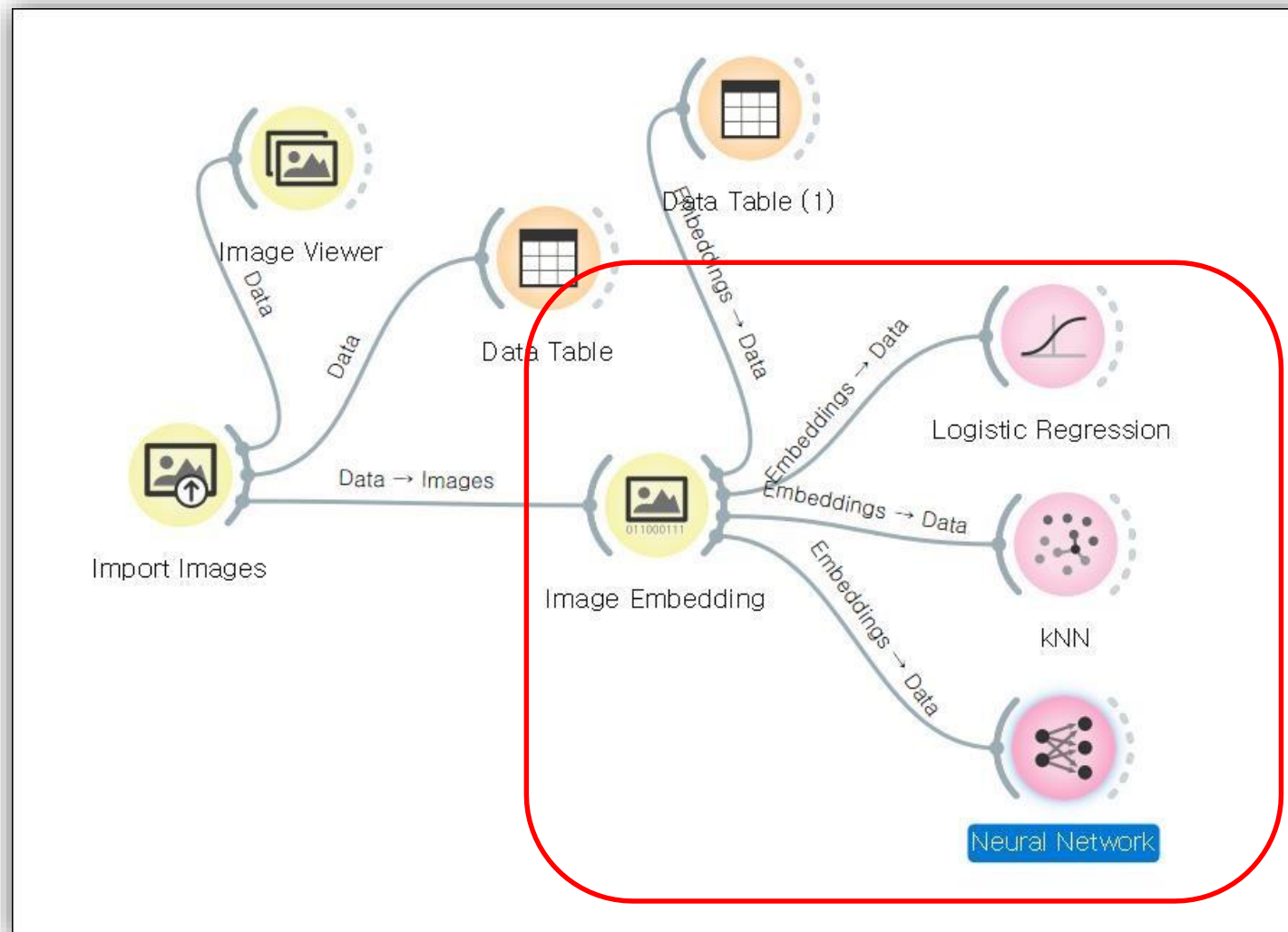
Selection
☒ Select full rows

| hidden origin | category | image name | image | size | width | height | n0 True | n1 True | n2 True | n3 True |
|---------------|----------|------------|-----------|--------|-------|--------|------------|------------|------------|------------|
| 1 | can | IMG_8369 | can/IM... | 157... | 3024 | 3024 | 0.0154644 | 0.0364313 | 0.012875 | 0.0609232 |
| 2 | can | IMG_8371 | can/IM... | 191... | 3024 | 3024 | 0.281874 | 0.396972 | 0.101616 | 0.467086 |
| 3 | can | IMG_8373 | can/IM... | 190... | 3024 | 3024 | 0.229289 | 0.441259 | 0.0846761 | 0.137239 |
| 4 | can | IMG_8377 | can/IM... | 224... | 3024 | 3024 | 0.191763 | 0.277441 | 0.0791516 | 0.764751 |
| 5 | can | IMG_8378 | can/IM... | 161... | 3024 | 3024 | 0.0936743 | 0.156965 | 0.32661 | 0.332358 |
| 6 | can | IMG_8382 | can/IM... | 215... | 3024 | 3024 | 0.21238 | 0.3587 | 0.744584 | 0.426189 |
| 7 | can | IMG_8482 | can/IM... | 193... | 4032 | 3024 | 0.349499 | 0.24808 | 0.0289729 | 0.313097 |
| 8 | can | IMG_8483 | can/IM... | 205... | 4032 | 3024 | 0.165807 | 0.160798 | 0.0230801 | 0.402359 |
| 9 | can | IMG_8499 | can/IM... | 180... | 3024 | 3024 | 0.23319 | 0.155284 | 0.0180272 | 0.154346 |
| 10 | can | IMG_8500 | can/IM... | 181... | 3024 | 3024 | 0.212621 | 0.127727 | 0.0143548 | 0.420538 |
| 11 | can | IMG_8531 | can/IM... | 239... | 3024 | 4032 | 0.128347 | 0.242689 | 0.158449 | 0.190803 |
| 12 | can | IMG_8532 | can/IM... | 212... | 3024 | 4032 | 0.320586 | 0.240914 | 0.0837321 | 0.268665 |
| 13 | can | IMG_8533 | can/IM... | 267... | 3024 | 4032 | 0.299443 | 0.520747 | 0.0317826 | 0.641825 |
| 14 | can | IMG_8681 | can/IM... | 305... | 3024 | 4032 | 0.296801 | 0.530492 | 0.14444 | 0.567239 |
| 15 | can | IMG_8682 | can/IM... | 318... | 3024 | 4032 | 0.630467 | 0.501656 | 0.0404201 | 0.650124 |
| 16 | can | IMG_8711 | can/IM... | 274... | 3024 | 4032 | 0.108399 | 0.433638 | 0.131286 | 0.327397 |
| 17 | can | IMG_8813 | can/IM... | 208... | 3024 | 4032 | 0.210809 | 0.14953 | 0.550386 | 0.138647 |
| 18 | can | IMG_8814 | can/IM... | 207... | 3024 | 4032 | 0.146971 | 0.358813 | 0.0287029 | 0.220863 |
| 19 | can | IMG_8815 | can/IM... | 201... | 3024 | 4032 | 0.11518 | 0.322851 | 0.103483 | 0.0791973 |
| 20 | can | IMG_8882 | can/IM... | 177... | 3024 | 4032 | 0.185155 | 0.281781 | 0.226727 | 0.513736 |

■ 학습 모델 선택하기

- 모델의 성능을 분석 및 비교하기 위해 여러 가지 모델 위젯을 동시에 연결하여 학습시킨다.
- 재활용품을 분류하는 인공지능을 만들기 위해 분류에 좋은 성능을 보이는 로지스틱 회귀, k-최근접 이웃, 인공 신경망 모델을 활용한다.

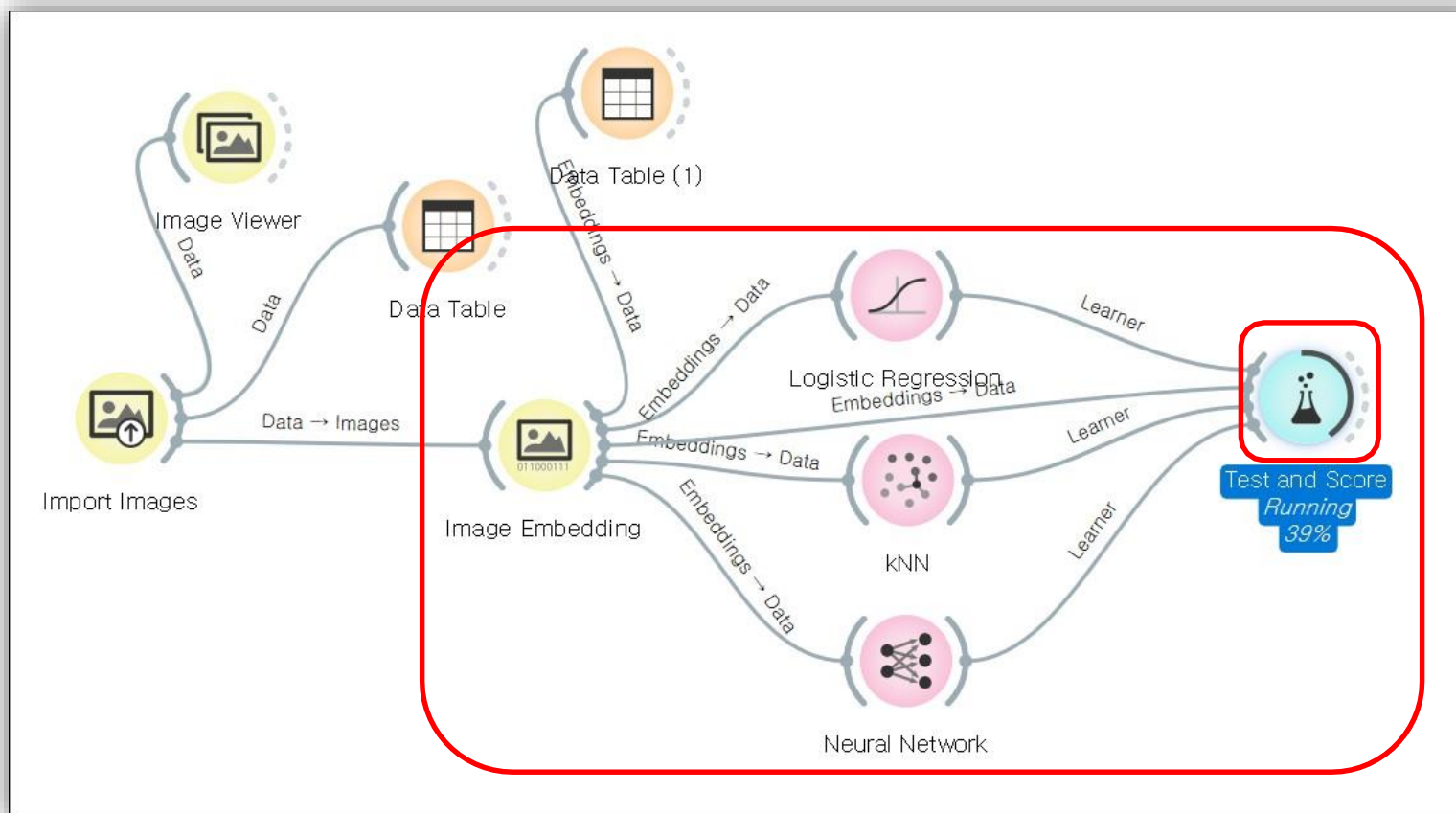




■ 모델 성능 확인하기

■ 학습 결과 확인하기

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



- [Test and Score] 위젯을 더블 클릭하여 각 모델의 성능을 확인한다.
- [Test and Score] : 여러 모델 중 최적의 모델을 확인하는 위젯

Test and Score - Orange

☒ Cross validation
Number of folds: 5
☒ Stratified
☐ Cross validation by feature
☐ Random sampling
Repeat train/test: 10
Training set size: 66 %
☒ Stratified
☐ Leave one out
☐ Test on train data
☐ Test on test data

Evaluation results for target (None, show average over classes)

| Model | AUC | CA | F1 | Prec | Recall | MCC |
|---------------------|-------|-------|-------|-------|--------|-------|
| Neural Network | 0.989 | 0.900 | 0.898 | 0.900 | 0.900 | 0.874 |
| Logistic Regression | 0.991 | 0.896 | 0.891 | 0.899 | 0.896 | 0.870 |
| kNN | 0.968 | 0.793 | 0.788 | 0.798 | 0.793 | 0.742 |

Compare models by: Area under ROC curve
☐ Negligible diff.: 0.1

| | Logistic Regression | kNN | Neural Network |
|---------------------|---------------------|-------|----------------|
| Logistic Regression | | 0.992 | 0.942 |
| kNN | 0.008 | | 0.012 |
| Neural Network | 0.058 | 0.988 | |

Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.