

개별 이미지를 통해 가정 쓰레기의 종류 분류하기

(05) 경제학부 응용통계전공 201812799 홍연정

CONTENT



INTRODUCTION

- 주제 선정 배경
- 주제 및 목적



MODELING

- DATA 설명
- 알고리즘 설명
- 이미지 분류 과정



CONCLUSION

- 새로운 이미지 분류 결과
- 결론 및 한계점

INTRODUCTION



주제 선정 배경

경상일보 4일 전

바뀐 분리수거 제도 시행됐지만 정착은 아직

올해
기간

뉴스포스트 | 2021.05.04.

[르포] 빌라 앞 음식물 오물...주택가 쓰레기 분리수거 '영망'

"분리수거? 아파트만 그나마 하는 척하고, 주택이나 빌라는 아예 되지 않는 것 같다. 정확한 분리배출 방법에 대한 홍보가 제대로 이루어지고 있는지 모르겠다." ...



세계일보 PICK | 2021.04.25. | 네이버뉴스

해양 쓰레기 플라스틱 병이 '아귀 배'에서 나왔어요

플라스틱 쓰레기에 대한 해양 오염의 심각성은 국내외에서 알려진 바 있다. 비슷한 시기에 물디브에서는 밧줄에 묶인 고래상어가 발견되는가 하면, 한 어부가 바닷물



연합뉴스 | 1일 전 | 네이버뉴스

광명시, 폐기물 활용 친환경제품 생산 플랫폼 추진

경
통

머니투데이 | 2021.04.22. | 네이버뉴스

'4-22지구의 날' "쓰레기도 자원"...페트병→목도리로 변신

이에 대해 박 대표는 "버려진 쓰레기를 일일이 분류하고, 세척하고, 다시 말리는 과정도 전부 비용"이라며 "천 같은 경우는 다림질도 해야 한다"고 어려움을 전했다....



분리수거가 제대로 이루어지지 않고 있음



육지에서 생산되는 쓰레기가 해양으로
유입됨에 따라 해양쓰레기 증가 및 생태계 위협



많은 기업에서
친환경 제품 생산에 관심 및 노력

[주제 및 목적]

- ✓ 개별 쓰레기 이미지를 12개의 class로 분류함으로써 자동화된 쓰레기 분류 프로세스 기대
- ✓ 쓰레기를 세분화된 등급으로 분류함으로써 재활용 쓰레기의 비율 증가 기대
- ✓ 재활용 쓰레기를 이용한 친환경 제품의 증가 기대



쓰레기 분류 로봇의 기초 알고리즘 작성

MODELING



DATA 설명

- ▶ 출처 : kaggle (<https://www.kaggle.com/mostafaabla/garbage-classification>)



- ▶ 12가지 종류의 가정 쓰레기 (총 12개의 클래스, 15,150개의 이미지 존재)



battery



biological



brown-glass



cardboard



green-glass



clothes



metal



paper



shoes



trash



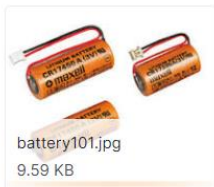
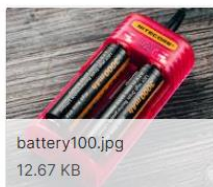
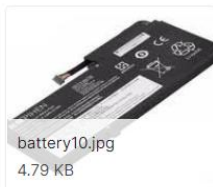
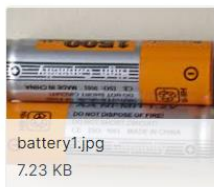
white-glass



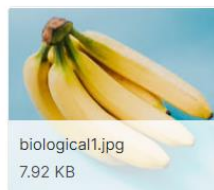
plastic

DATA 설명

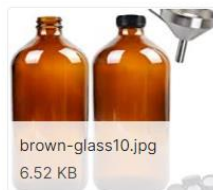
0. Battery 클래스 예시 (총 945개)



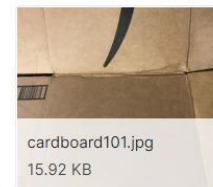
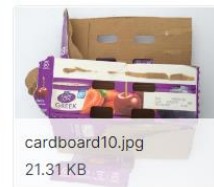
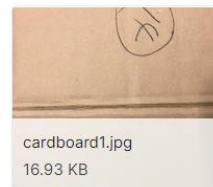
1. Biological 클래스 예시 (총 985개)



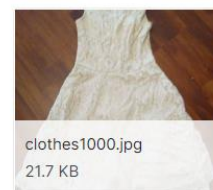
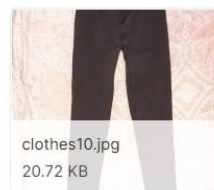
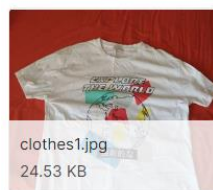
2. Brown-glass 클래스 예시 (총 607개)



3. Cardboard 클래스 예시 (총 891개)



4. Clothes 클래스 예시 (총 5325개)

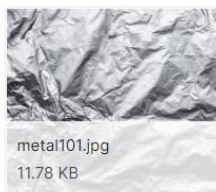
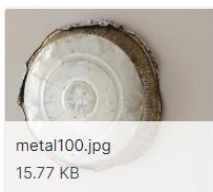
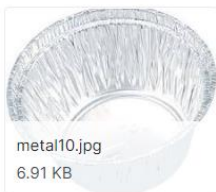
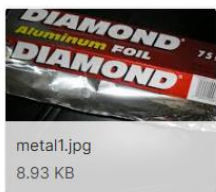


5. Green-glass 클래스 예시 (총 629개)

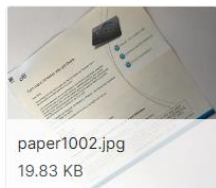
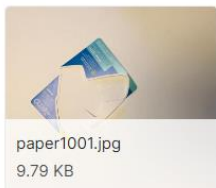


DATA 설명

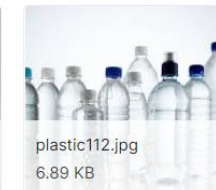
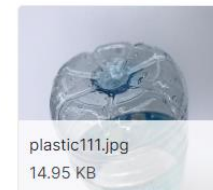
6. Metal 클래스 예시 (총 769개)



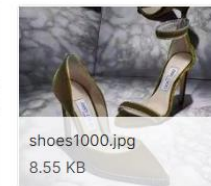
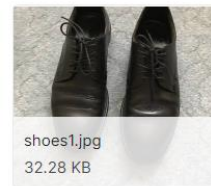
7. Paper 클래스 예시 (총 1050개)



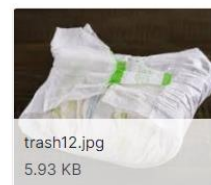
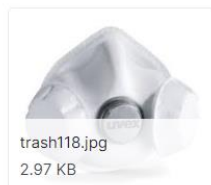
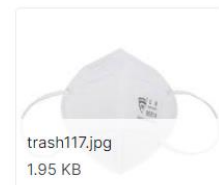
8. Plastic 클래스 예시 (총 865개)



9. Shoes 클래스 예시 (총 1977개)



10. Trash 클래스 예시 (총 697개)



11. White-glass 클래스 예시 (총 775개)



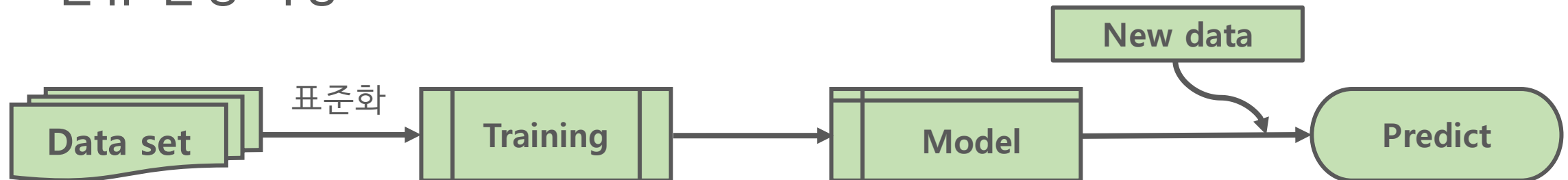
알고리즘 설명

Deep Learning

인간의 뉴런과 비슷한 인공신경망 방식으로 정보를 처리

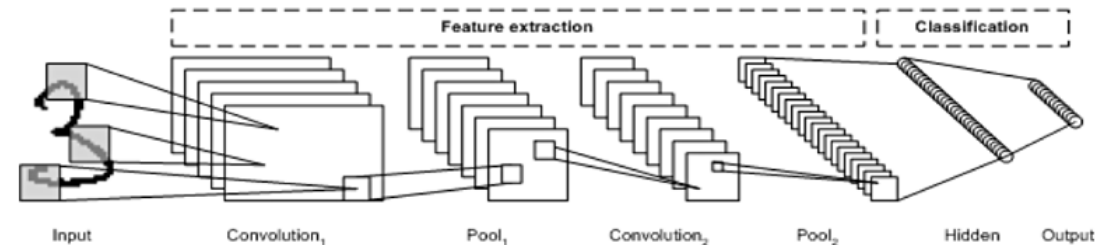


▶ 분류 진행 과정 Workflow



CNN

Convolutional Neural Network, 합성곱 신경망
이미지 인식에 탁월한 효과를 보이는 신경망으로,
하나의 이미지를 다양한 관점으로 변환시켜 종합해 판단

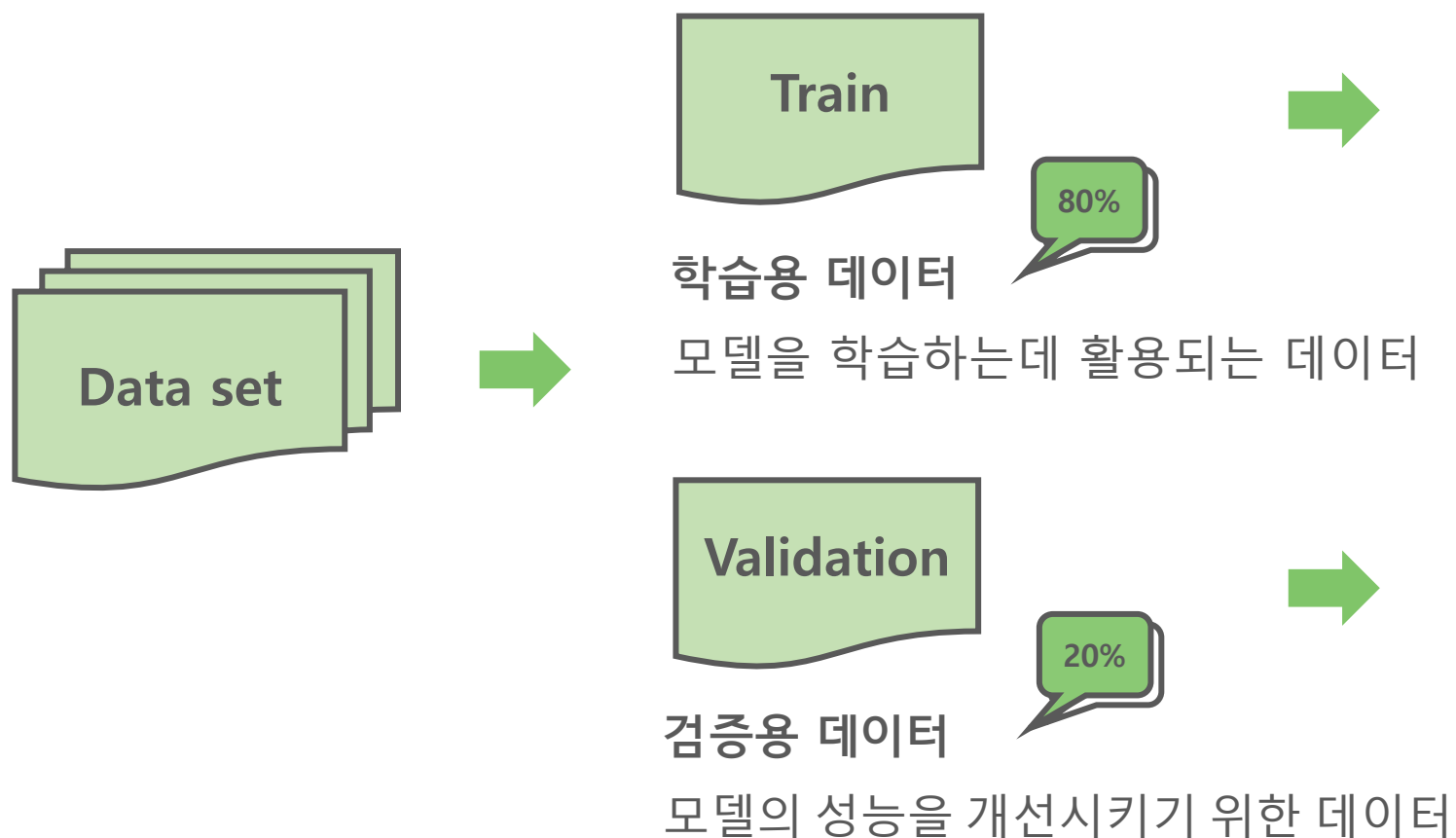


[예시] 숫자 손 글씨 이미지를 분류

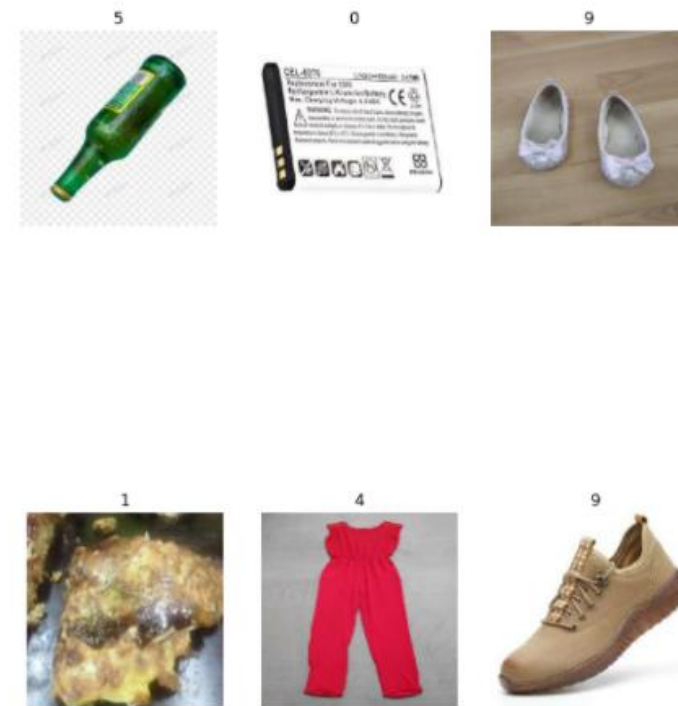
[이미지 분류 과정]

▶ Data set 분할

Data set을 8 : 2로 Train set과 Validation set 분할

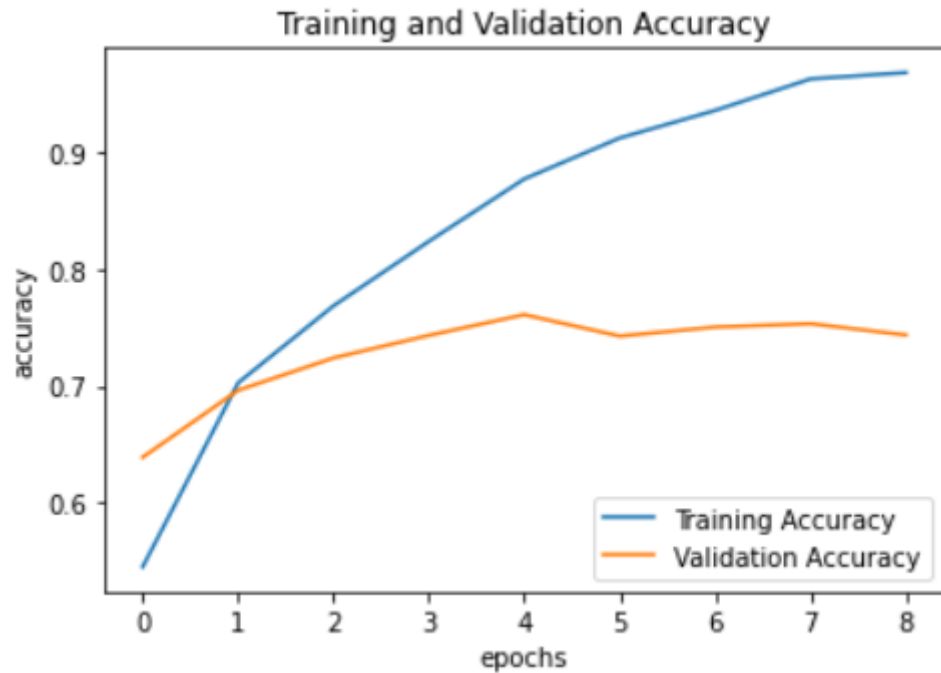


< 예시 >

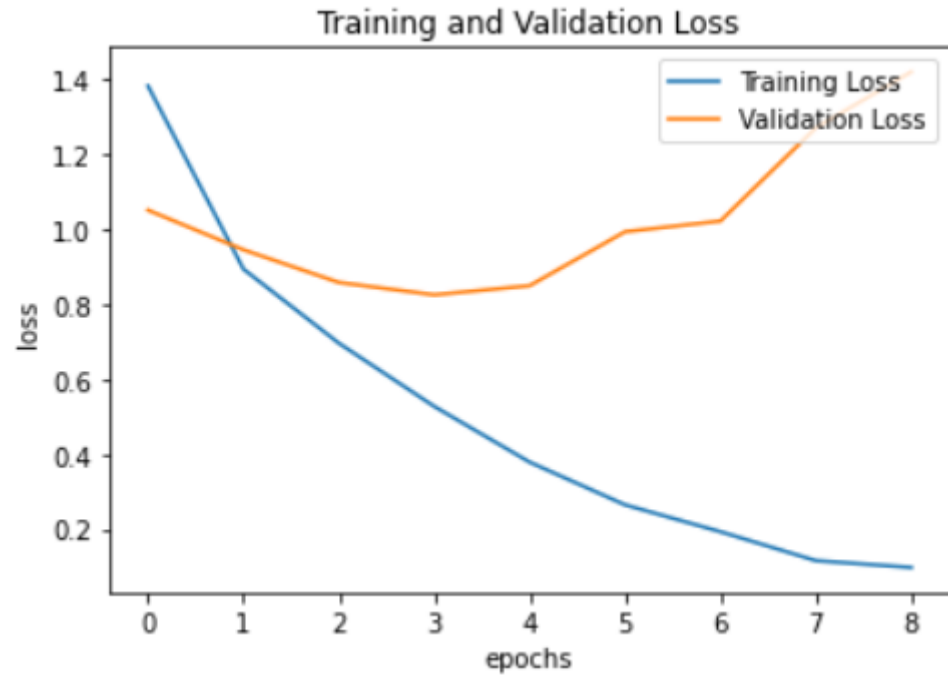


[이미지 분류 과정]

▶ 첫 번째 모델의 훈련 및 검증 세트에 대한 정확성과 손실



정확성



손실

※ epochs?

- 훈련 데이터셋에 포함되어 있는 데이터 값들이 한 번씩 예측 모델에 들어온 뒤 weight 값을 갱신하는 주기
- 여러 번의 epochs 사용하는 이유 : 이후 새로운 데이터에 대한 일반화 능력을 향상시키기 위해
- 횟수가 많아질수록 loss는 줄어들고 accuracy는 증가하게 되어 어느 시점이 되면 일정 값에 수렴하다가 성능이 점점 떨어짐
→ Over-fitting, 과대적합

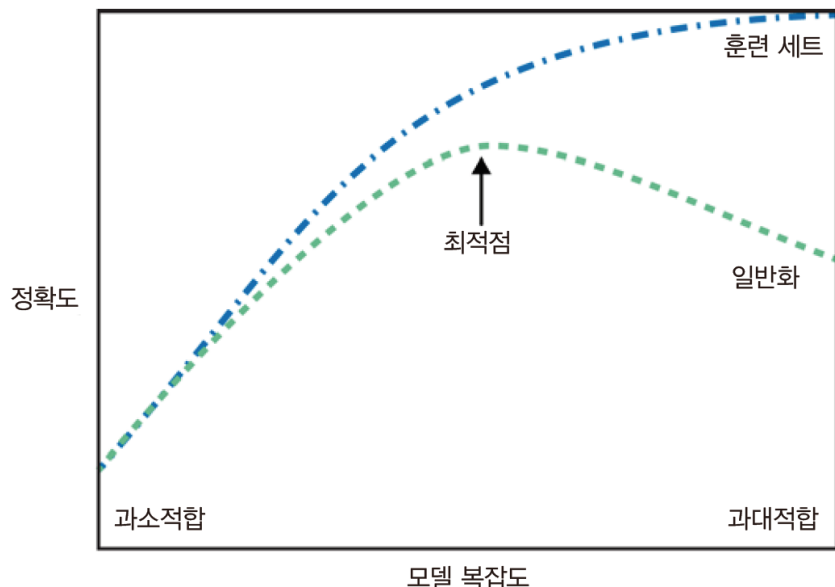
Over-fitting

▶ Over-fitting 이란?

인공지능에서 학습 데이터에 지나치게 맞는 모델을 학습함으로써 일반화 성능이 떨어지는 현상.
즉, 모델 파라미터가 너무 많아 복잡한 모델을 만들기 때문에 새로운 데이터에는 대응하기 힘든 것을 의미.

※ 통계에서의 Overfitting(과다적합) 분석?

모수 절약의 원칙을 위배해 계수 하나를 일정한 규칙이나 임의로 부가해 그 결과가 더 적합한가의 여부를 분석하는 것.



- 훈련 정확성이 시간의 지남에 따라 선형적으로 증가하는 반면 검증 정확성은 훈련 과정에서 약 70%를 벗어나지 못함
- 훈련 정확성과 검증 정확성 간의 차이가 큼

Over-fitting 가능성 존재

Over-fitting

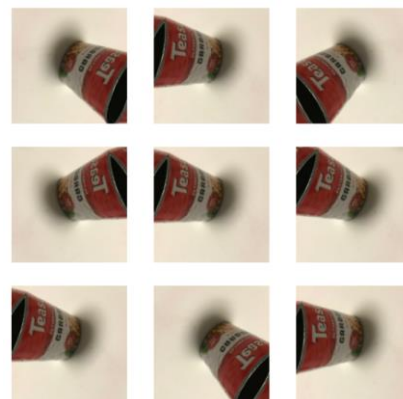
▶ Over-fitting을 방지하기 위한 방법 ①

이미지 증강 (Augmentation)

- 이미지를 생성하는 임의 변환해 기존에서 추가 훈련 데이터를 생성하는 접근법을 취하는 것
- Train set을 더 많이 확보 가능해 overfitting 방지 가능
- 왼쪽에서 오른쪽으로 뒤집은 후, 시계 방향 혹은 시계 반대 방향으로 $\pm 0.1 \times 2\pi$ 무작위 회전 적용



<원래 이미지>



<이미지 증강 적용>

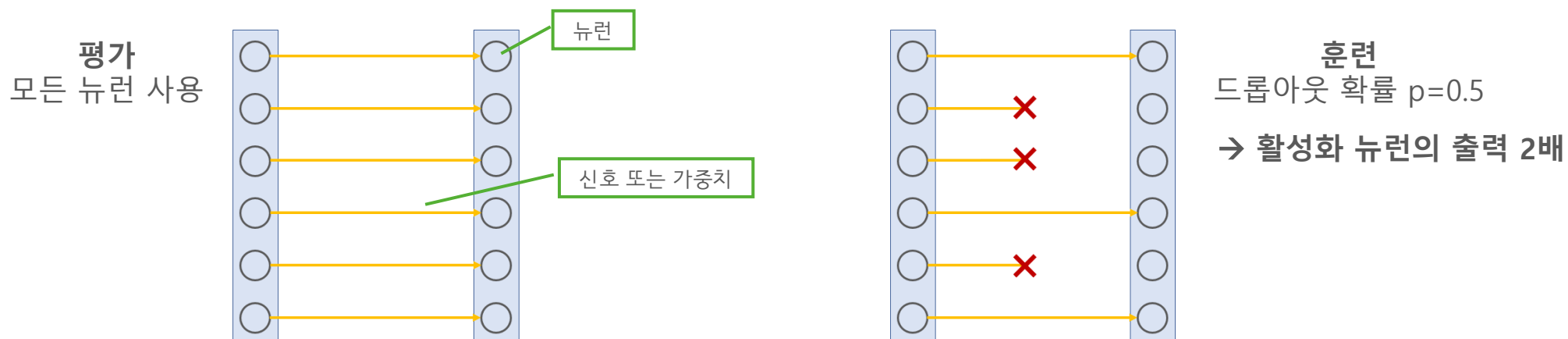
데이터의 더 많은 측면을 파악하게 되므로 예측 범위가 넓어져 일반화가 더 쉬워지는 효과

Over-fitting

▶ Over-fitting을 방지하기 위한 방법 ②

드롭아웃 (Dropout)

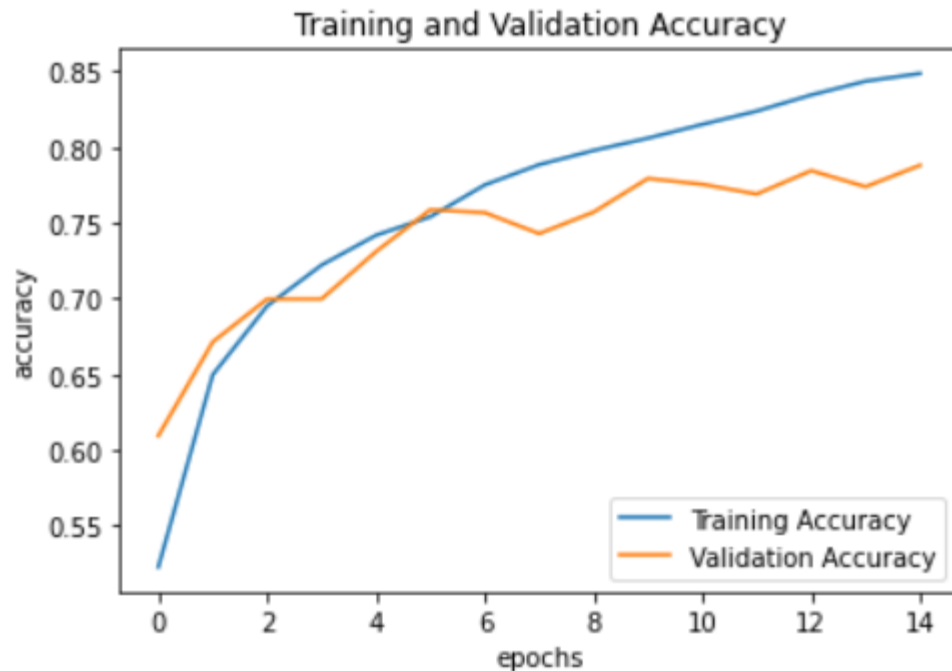
- 훈련할 때 임의의 뉴런을 골라 삭제해 신호를 전달하지 않게 함. (테스트할 때는 모든 뉴런 사용)
- 0.1, 0.2, 0.4 등의 형식의 소수를 입력 값으로 사용(최대 1) → 10%, 20%, 40%를 임의로 제거 의미
- 남은 뉴런에 연결된 가중치가 삭제된 뉴런의 비율만큼 증가



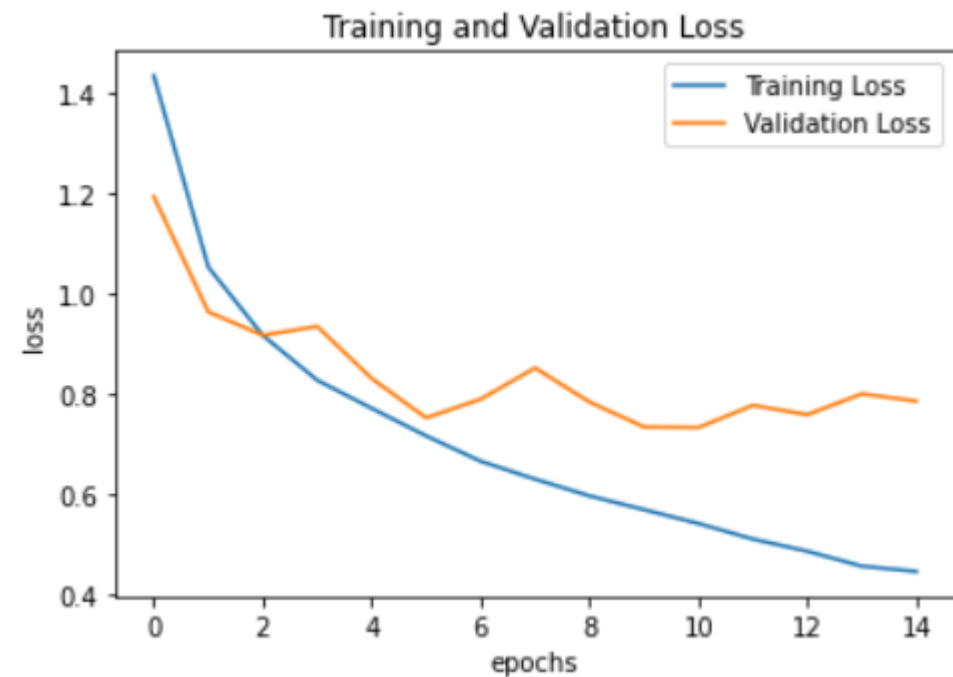
어떤 뉴런이 다른 특정 뉴런에 의존적으로 변하는 것 방지 효과 (상호 적응 문제 회피)

[이미지 분류 과정]

▶ 과대적합 방지 방법을 적용한 모델의 훈련 및 검증 세트에 대한 정확성과 손실



정확성



손실

정확성이 약 75%로 이전의 모델보다 증가 → 과대적합이 조금은 해결되었다고 할 수 있음

CONCLUSION



[새로운 데이터 적용]

▶ 예시 1



```
predictions = model.predict(img_array1)
score = tf.nn.softmax(predictions[0])
```

```
print('분류 결과 : ', class_label[np.argmax(score)])
print('정확성 : {:.2f}%'.format(100 * np.max(score)))
```

분류 결과 : **green-glass**
정확성 : 73.67%

옳은 분류

▶ 예시 2



```
predictions = model.predict(img_array2)
score = tf.nn.softmax(predictions[0])
```

```
print('분류 결과 : ', class_label[np.argmax(score)])
print('정확성 : {:.2f}%'.format(100 * np.max(score)))
```

분류 결과 : **white-glass**
정확성 : 90.50%

옳은 분류

[새로운 데이터 적용]

▶ 예시3



```
predictions = model.predict(img_array4)
score = tf.nn.softmax(predictions[0])

print('분류 결과 : ', class_label[np.argmax(score)])
print('정확성 : {:.2f}%'.format(100 * np.max(score)))
```



옳은 분류

분류 결과 : **biological**
정확성 : 99.99%

▶ 예시4



```
predictions = model.predict(img_array6)
score = tf.nn.softmax(predictions[0])

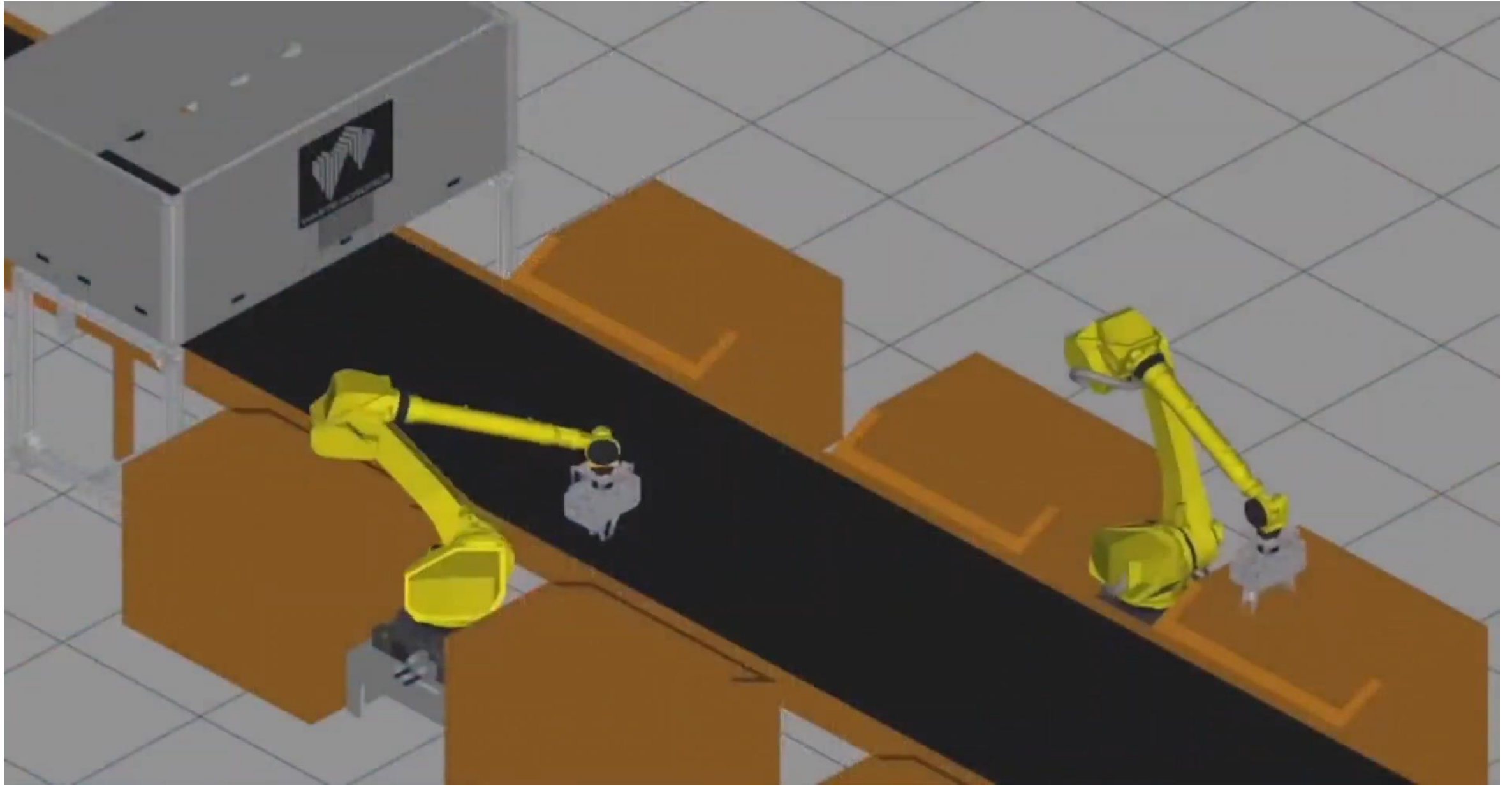
print('분류 결과 : ', class_label[np.argmax(score)])
print('정확성 : {:.2f}%'.format(100 * np.max(score)))
```



옳은 분류

분류 결과 : **battery**
정확성 : 99.76%





[결론 및 한계점]

- ✓ 75 ~ 80%의 정확도를 가지는 모델 구성
- ✓ 새로운 이미지를 넣었을 때 100% 완벽하게 분류하지 못한다는 아쉬운 점 존재
- ✓ 더 좋은 데이터로 모델을 보완한다면 정확성이 더 높아질 것이라고 기대
 - 클래스 별로 데이터셋의 크기가 유사하고 다양한 데이터 등
- ✓ 시스템이 더 개발되어 적은 데이터의 학습으로 더 좋은 성능을 나타내는 알고리즘 개발 기대



하나의 로봇이 하나의 종류를 완벽하게 분류한다면,

여러 개의 로봇이 각각 하나의 종류를 분류하게 함으로써 재활용 프로세스가 개선될 것

THANK YOU

