

시각장애인을 위한 캔음료, 컵라면 인식 시스템

팀 도르마무

2022-12-15

구성원



함형빈

팀장

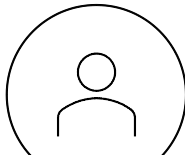
- 데이터 수집
- 데이터 전처리
- 탐색적 분석
- 모델링
- 대시보드
- 자료조사



서태웅

팀원

- 데이터 수집
- 데이터 전처리
- 탐색적 분석
- 모델링
- 자료조사



조재혁

팀원

- 데이터 수집
- 데이터 전처리
- 모델링
- TTS
- 자료조사



전동환

팀원

- 데이터 수집
- 데이터 전처리
- TTS
- 대시보드
- 자료조사

Table of contents

- 1** 프로젝트 목표
- 2** 실증 연구 모델 - 1
- 3** 실증 연구 모델 - 2
- 4** 모델 개선
- 5** 프로젝트 결과



Part 1

프로젝트 목표



시각장애인을 위한 캔음료, 컵라면 인식 시스템

- 시각장애인을 위해 **점자가 없는 제품**(캔음료, 컵라면)을 **이미지 인식**과 **음성 출력**으로 어떤 제품인지 안내하는 **AI 서비스**를 만들고자 한다.
- 캔음료와 컵라면을 대상으로 영상 이미지 인식을 통한 음성 서비스를 제공하는 시제품을 제작하지만, **다양한 제품에 적용 시킨다면 시각장애인의 삶의 질이 향상될 것으로 기대한다.**



Part 2

실증 연구 모델 - 1

이미지 검색



이미지 수집



Fatkun

이미지 저장



1. Chrome 수집 대상 이미지 검색
2. Fatkun 확장 프로그램 활용 이미지 수집
3. 각 Class 당 100장씩 해당하는 구글 드라이브 폴더에 저장
4. 테스트 데이터는 직접 물품을 찍은 이미지로 평가했다.

Cocacola

Gatorade

jin

nuguri

Poweradecan

Pepsi

udong

uk

ZeroPepsi

zza

Train Data



Test Data



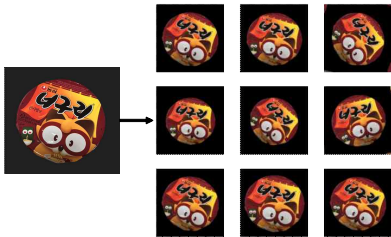
배경 삭제

- 객체 탐지 알고리즘이 아닌 단순 분류 모델에서 객체 외 배경을 제거하여 학습한다.
- OpenCV grabcut 을 활용했다.



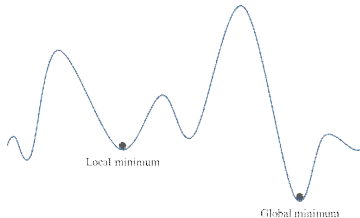
이미지 증강

- 가지고 있는 이미지 데이터의 수가 적고 다양성이 적을 때 이미지를 다양하게 만들어 주어 학습 이미지의 양을 늘리는 기법이다.



정규화

- 학습이 더 빨라지고 local minimum에 빠지는 것을 방지합니다. global minimum에 다가가는 것이 목표이기 때문이다.

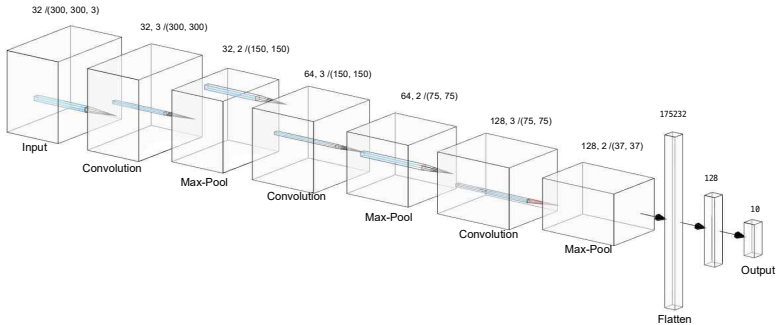


이미지 사이즈 조정

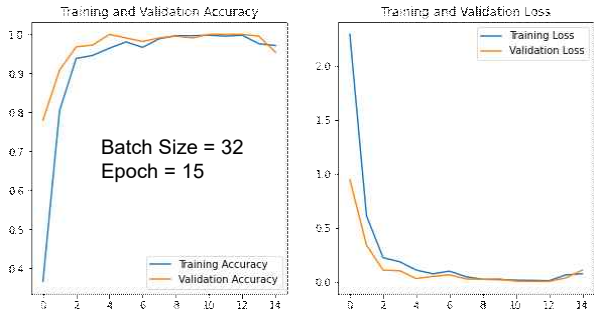
- 수집한 이미지는 다양한 크기를 가지고 있어 모델에 들어가기 전 특정 크기로 맞춰줘야 한다. 이미지 크기는 너무 작게 조정할 경우 이미지 손실이 일어날 수 있고 크게 조정할 경우 학습 시 시간이 오래 걸린다.
- (300 x 300 x 3) 으로 조정했다.



기본적인 CNN 구조를 채택



학습 곡선



- 불안전 하지만 Train data와 Validation data의 학습과 성능 향상 추세를 보인다.

평가

/	Accuracy	F1-score
Test data	19%	0.12

	precision	recall	f1-score	support										
					0	1	2	3	4	5	6	7	8	9
cocacola	0.00	0.00	0.00	31	0	3	0	0	0	0	0	0	28	8
gatorade	0.15	0.65	0.24	23	0	15	0	0	0	0	0	0	0	8
jin	0.00	0.00	0.00	24	0	2	0	0	0	0	6	0	0	16
nuguri	0.00	0.00	0.00	21	0	1	0	0	0	0	18	0	0	2
pepsi	0.77	0.46	0.58	37	0	16	0	0	17	0	0	0	0	4
powerade	0.00	0.00	0.00	56	0	47	0	0	5	0	0	0	0	4
udong	0.11	0.43	0.18	7	0	4	0	0	0	0	3	0	0	0
uk	0.00	0.00	0.00	6	0	4	0	0	0	0	0	0	0	0
zero	0.00	0.00	0.00	12	0	4	0	0	0	0	0	0	1	1
zza	0.10	0.88	0.18	8	0	10	0	0	0	0	0	0	0	2
accuracy			0.19	225	0	1	0	0	0	0	0	0	0	7
macro avg	0.11	0.24	0.12	225	0	1	0	0	0	0	0	0	0	7
weighted avg	0.15	0.19	0.13	225	0	1	2	3	4	5	6	7	8	9

Classification Report

Confusion Matrix

- 게토레이와 짜파게티에 치중되서 학습된 것을 확인할 수 있다.
 - Test data에서 **Overfitting** 된 것을 확인할 수 있었다.



1. train data와 test data가 상이하다. (데이터 문제)
2. 실제 우리가 사진 찍을 때와 비슷한 데이터를 수집해야 한다.
3. 처음부터 복잡한 이미지를 훈련시키지 말아야 한다.
4. GrabCut을 이용한 배경삭제는 물품을 제외한 배경도 포함을 시키는 경우가 많아 배경삭제에 의미가 없어졌다.



Part 3

실증 연구 모델 - 2



이미지 저장



[실증 연구 모델 - 1]의 피드백을 수용하여
크롤링을 통해 실제와 비슷한 사진과 클래스에 해당
물품을 구매하여 OpenCV를 활용해 카메라로 직접
촬영한 사진 섞어서 훈련 데이터로 활용했다.

Train Data

jin



zero



udong



udong



cocacola



cocacola



powerade



uk



jin



Test Data

pepsi



jin



getorade



jin



powerade



powerade



zero



powerade

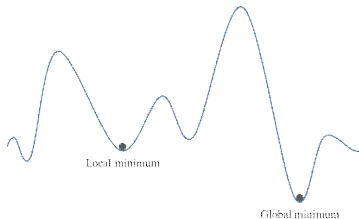


udong



정규화

- 실증 연구 모델 - 1 과 동일하게 시행했다.



이미지 사이즈 조정

- 실증 연구 모델 - 1 과 동일하게 시행했다.



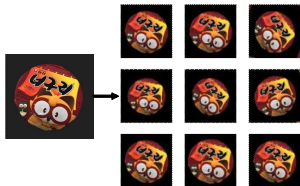
배경 삭제

- 객체 탐지 알고리즘이 아닌 단순 분류 모델에서 객체 외 배경을 제거하여 학습한다.
- Python **rembg** 라이브러리를 활용해 더 정교히 제거했다.

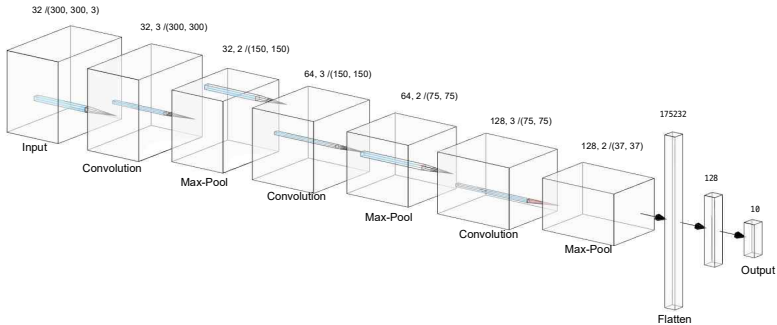


이미지 증강

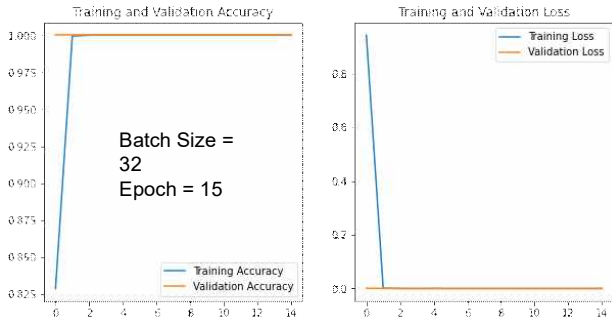
- 가지고 있는 이미지 데이터의 수가 적고 다양성이 적을 때 이미지를 다양하게 만들어 주어 학습 이미지의 양을 늘리는 기법이다.



[실증 연구 모델 - 1]과 동일한 조건으로 시행했다.



학습 곡선



- Train data를 실제 사건과 유사한 이미지를 사용해서 Validation data는 Accuracy 100%를 보였다.

평가

/	Accuracy	F1-score
Test data	36%	0.14

	precision	recall	f1-score	support		0	1	2	3	4	5	6	7	8	9
coca	1.00	0.20	0.33	30	0	6	24	0	0	0	0	0	0	0	0
getorade	0.35	1.00	0.52	68	1	0	68	0	0	0	0	0	0	0	0
jin	0.50	0.04	0.08	23	2	0	19	1	0	0	0	0	0	0	3
nuguri	0.00	0.00	0.00	24	3	0	22	0	0	0	0	0	0	0	2
pepsi	0.00	0.00	0.00	40	4	0	14	0	0	0	26	0	0	0	0
powerade	0.45	0.49	0.47	51	5	0	16	0	0	0	25	0	10	0	0
udong	0.00	0.00	0.00	8	6	0	6	1	0	0	0	0	0	0	1
uk	0.00	0.00	0.00	9	7	0	9	0	0	0	0	0	0	0	0
zero	0.00	0.00	0.00	12	8	0	6	0	0	0	5	0	0	0	1
zza	0.00	0.00	0.00	9	9	0	9	0	0	0	0	0	0	0	0
accuracy			0.36	274	0	1	2	3	4	5	6	7	8	9	
macro avg	0.23	0.17	0.14	274											
weighted avg	0.32	0.36	0.26	274											

Classification Report

Confusion Matrix

- 이전 모델보다 정확도는 상승했지만 f1-score는 변함이 없었다.
대부분 게토레이로 예측하는 경우가 많았다.

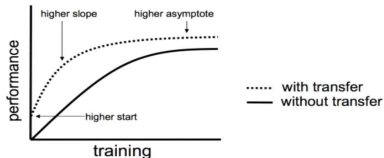
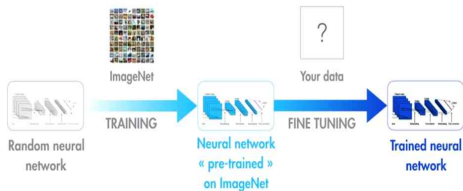


1. 학습 곡선과 테스트 데이터의 평가 지표는 비례하지 않는다.
2. 학습 데이터와 테스트 데이터의 차이가 일부 존재한다.
3. 데이터의 양과 품질을 개선해야 하지만 데이터 수집에 한계가 있으므로 전이학습 기법을 이용하여 이미지에서 특징을 디테일 하게 추출하도록 한다.



Part 4

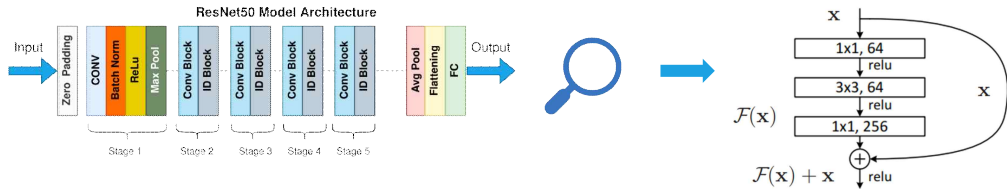
모델 개선



• 전이학습이란?

- ✓ 전이학습은 딥러닝의 학습방법 중 하나로 분류(Classification)나 탐지(Detection), 분할(Segmentation) 업무에 대해 **학습된 딥러닝 모델을 다른 데이터셋에 활용하는 것**을 말한다.
- ✓ 지식전이를 통해 기본성능을 향상시킬 수 있으며, 처음부터 모델을 학습시키는 것 보다 빨라 모델 개발 시간에서도 이득을 볼 수 있고 **데이터의 수가 적더라도 좋은 성능을 보여준다**.

ResNet50



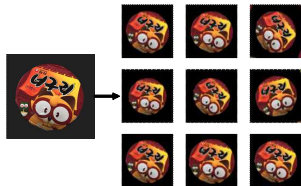
컨볼루션과 활성화 함수를 거친 값과 처음 Input을 합쳐서
 Backpropagation에서 발생하는 깊은 신경망 Vanishing Gradient를 해결한다.
 덧셈 연산만 증가해 매개변수 개수에 영향주지 않고 연산량 증가도 거의 없다.
 따라서 깊은 신경망을 사용할 수 있고 전체적인 훈련 정확성이 올라가게 된다는 장점이 있다.

데이터 수집

- [실증 연구 모델 - 2]과 동일한 데이터 사용.

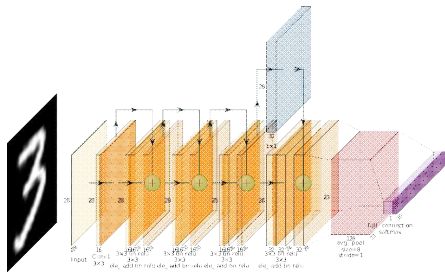
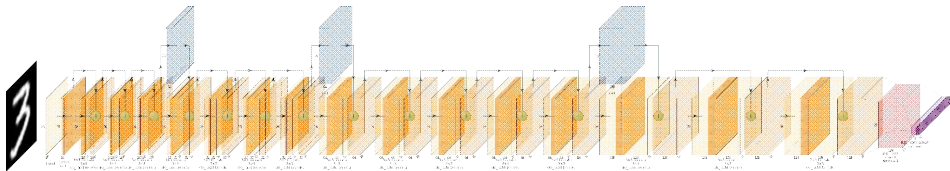
배경삭제, 이미지 증강, 정규화, 이미지 사이즈 조정

- [실증 연구 모델 - 2]과 동일한 조건으로 시행했다.



Part 4

모델 개선 / 모델 구조



- 사전 학습된 ResNet50 모델 생성

ImageNet으로 훈련된 가중치가 저장된 ResNet50 모델을 불러온다.

ResNet은 Imagenet 을 학습한 모델이기 때문에 output 수는 1000개이고 Input Size 는 (224, 224)이다.

우린 10개 클래스를 분류를 하기 때문에Dense layer를 제외한 Conv layer만 가져오고 Input Size는 (300, 300) 으로 정의하고 output 수를 10으로 재정의 한다.

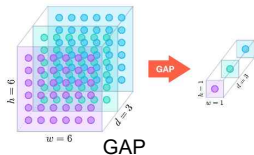
- 특징 추출을 위한 베이스 모델 동결

사전 학습된 모델을 새로운 이미지 데이터에 대해 유의미한 feature을 뽑아내는 용도로 사용한다.

최종 태스크를 수행할 수 있는 classifier만을 더해 사전 학습된 모델이 만든 feature map을 새로운 태스크 수행에 활용한다.

```
[ ] base_model.trainable = False
```

동결(layer.trainable = False로 설정)은 주어진 레이어의 가중치가 훈련 중에 업데이트되는 것을 방지한다.



• 분류층 추가하기

베이스 모델을 가져올 때 분류층을 제외하여 가져왔기 때문에 분류층을 추가하도록 한다.

보통 Conv layer 이후 Flatten을 이용해 Feature Map을 1차원으로 늘린다.

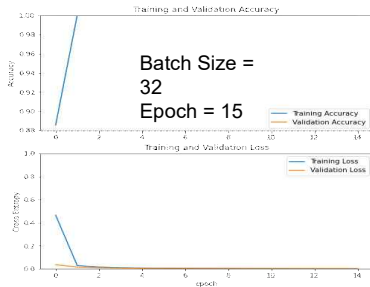
하지만 이때 파라미터 수가 증가하는 단점이 있기 때문에 **Global_Average_Pooling**을 사용한다.

GAP는 FC Layer에 비해 location 정보를 상대적으로 덜 잃게 된다.

Output을 10개로 맞춰 준다.

즉 **ResNet50**을 특징 추출기로 사용하고 분류층만 학습하는 것이다.

학습 곡선



- Training data는 학습 처음부터 높은 정확도를 보였고 Validation data 역시 Accuracy 100%를 보였다.

평가

/	Accuracy	F1-score
Test data	73%	0.69

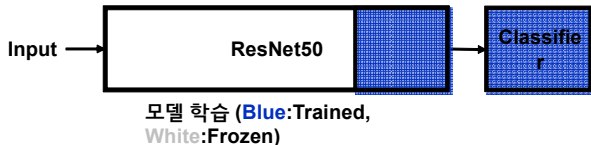
	precision	recall	f1-score	support
cocacola	0.83	1.00	0.91	30
getorade	0.98	0.62	0.76	68
jin	0.48	1.00	0.65	23
nuguri	1.00	0.58	0.74	24
pepsi	0.72	0.65	0.68	40
powerade	0.58	0.90	0.71	51
udong	1.00	0.38	0.55	8
uk	1.00	0.56	0.71	9
zero	1.00	0.67	0.80	12
zza	1.00	0.22	0.36	9
accuracy			0.73	274
macro avg	0.86	0.66	0.69	274
weighted avg	0.81	0.73	0.72	274

Classification Report



Confusion Matrix

- Test data 평가 결과 Accuracy, F1 score 모두 높은 수치를 보였다.
 - 확실히 [실증 연구 모델 - 2] 보다 좋은 결과를 도출했다.
 - 하지만 아직 헛갈려 하는 물품들이 존재한다.



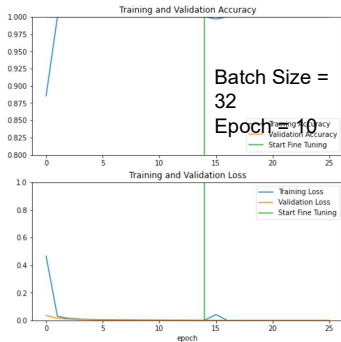
- 미세 조정 후 추가 학습

성능을 더욱 향상시키기 위해 분류기와 사전 훈련된 모델의 일부 최상위 레이어 가중치를 훈련한다. 훈련을 통해 가중치는 개별 데이터셋과 관련된 특징을 추출하도록 조정된다.

ResNet50을 동결 해제하고 입력 층과 하위 층을 훈련 할 수 없도록 설정한다. 그리고 모델을 다시 컴파일하고 훈련을 다시 시작한다.

또 **이전보다 낮은 학습률**을 설정한다.
그렇지 않으면 모델이 매우 빠르게 과대적합 될 수 있기 때문이다.

학습 곡선



- 전이학습 전후 학습 곡선이다.
마찬가지로 학습이 잘된 것을 확인할 수 있다.

평가

/	Accuracy	F1-score
Test data	90%	0.89

	precision	recall	f1-score	support
cocacola	0.97	1.00	0.98	30
getorade	0.83	0.91	0.87	68
jin	1.00	1.00	1.00	23
nuguri	1.00	1.00	1.00	24
pepsi	0.73	0.82	0.78	40
powerade	0.96	0.88	0.92	51
udong	1.00	0.50	0.67	8
uk	1.00	1.00	1.00	9
zero	1.00	0.58	0.74	12
zza	1.00	1.00	1.00	9
accuracy			0.90	274
macro avg	0.95	0.87	0.89	274
weighted avg	0.91	0.90	0.90	274

Classification Report



Confusion Matrix

- Test data 평가 결과 Accuracy, F1 score 모두 높은 수치를 보였다.
미세조정 전보다 더 좋은 결과를 도출했다.



Part 5

프로젝트 결과

Python **tkinter** 로 GUI를 제작해 작동하도록 했다.

캠 영상, 모델 Input 이미지, 확률 대시보드 및 버튼

캠에 물품을 비춘 후 Frame 버튼을 클릭하면 **음성과 확률 값이 나오도록** 했다.

10개 물품을 여러 번 테스트 했을 때 대부분 정확하게 분류했다.

하지만 빛이 물품에 심하게 반사되거나 특정 부분만 보일 경우 종종 분류에 실패한다.



Frame		
num	상품	확률
0	코카콜라	0
1	게토레이	0
2	진라면	0
3	너구리	0
4	엿시	0
5	파워에이드	0
6	우동	0
7	죽가장	0
8	제로엿시	0
9	피파게티	0

프로젝트 결과

훈련 데이터에 따라 성능 차이가 있고 모델 깊이가 깊을 수록 디테일 한 학습이 가능하다.

학습 곡선이 좋은 현상을 보여도 Test Data에서 평가가 좋지 않을 수 있다.

직접 수집한 데이터와 ResNet50 전이 학습 및 미세 조정 모델이 Test Data 평가가 좋았다.

Python Tkinter 를 이용한 GUI 프로그래밍으로 PC에서 작동하도록 했다.

목표론 20개 분류였지만 10개 분류로 축소해 완료했다.

시사점

다양한 물품을 학습시키고 모바일 앱으로 출시할 경우 시각장애인의 삶의 질이 향상될 것이다.

감사합니다.