

재활용품 분리기

김남규⁰, 김성빈⁰, 임세은, 임지윤
 세종대학교 컴퓨터공학과, 스마트기기공학과
antonio629@naver.com, zxas14@naver.com,
jessie0207@naver.com, ssen.lim@gmail.com

요 약

이 논문은 분리배출 과정에서의 사회적 문제 해결에 도움을 줄 수 있는 기초적인 인공지능 분리수거기의 개발 과정과 결과를 보이는 글이다.

I. 서론

현재 우리나라는 잘못된 분리배출로 인해 골머리를 앓고 있다. 분리배출에 대한 사람들의 인식 부족과 분리배출을 하더라도 보편화되지 않은 기준으로 인해 재활용품으로 잘못 분류되고 있는 쓰레기의 비중이 40%를 육박하고 있다. 이를 처리하는 데에는 매년 수억 원의 지원금이 필요하지만 지원금 부족으로 인건비와 처리 비용을 사비로 충당하고 있는 재활용 처리 업체도 있는 등 여러 문제가 발생하고 있다. 상황이 악화됨에 따라 재활용 처리 업체와 여러 환경운동연합은 무능력한 정부의 대책을 비판하며 불만을 토로하고 있다. 사회의 문제점을 해결하고 실생활에 도움이 되는 것을 목표로 하고 있는 리사이클조는 주제 선정 당시 현재 어린이 대공원 역에 설치된 '슈퍼빈(SuperBin)' 기업에서 개발한 분리수거기가 전국적으로 설치 된다는 소식을 접했다. 이 네프론은 빈 병, 페트, 캔의 3 가지 재활용품을 분류하고 적립금을 적립해주는 제도로 운영되고 있다. 우리 조는 이 기술의 발전과 확대를 3 가지가 아닌 더 많은 가짓수의 쓰레기를 재활용해주는 시스템이 개발되었으면 하는 바람과 조금이나마 사회에 기여를 할 수 있기를 바라는 마음으로 텀프로젝트 주제를 재활용품 분리기로 선정하였다. 훗날 재활용품 분리수거기가 각 가정과 기업에 보급화 되어 위와 같은 사회적 문제점들이 해결되기를 바라며 다음은 본 논문에서 제시하는 프로젝트의 진행 과정이다.

2 절에서는 데이터셋 수집과 처리 과정에 대하여, 3 절에서는 학습 방법에 대하여, 4 절에서는 결론을 서술한다. 다음은 '슈퍼빈'의 네프론 모습이다.

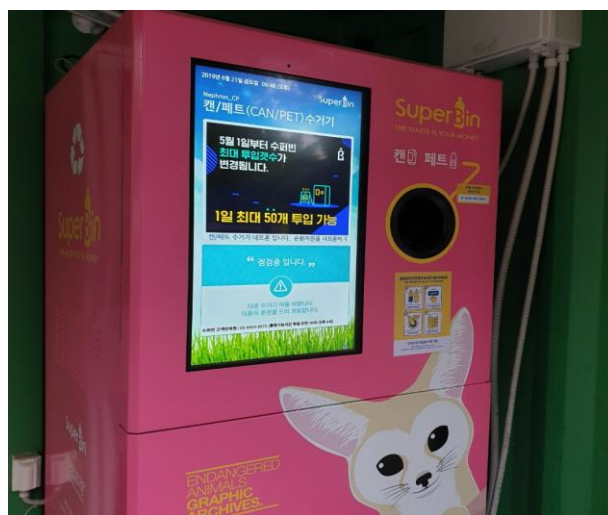


사진 1. 어린이 대공원 분리수거 기계: 네프론

II. 데이터셋

2.1 데이터 수집 및 분류

데이터 수집의 경우, trash net[1]의 기존셋과 Google Images Download[2]를 이용한 이미지 크롤링(crawling)을 사용했다. 데이터의 클래스분류는 한국의 분리수거 기준[3]을 참고하여 정했다. 분류 기준 중 비닐류(vinyl)는 데이터셋이 구하기가 어렵고 학습이 쉽지 않아 제외했다. 상자(cardboard)의 경우 분류기준에 포함했지만 성능 향상을 위해 종이(paper)에 포함시켰다. 이와 관련된 자세한 설명은 밑에서 보이겠다. 따라서 데이터 셋은 종이(paper), 금속(metal), 플라스틱(plastic), 유리(glass)의 4 가지이다.

종이류  신문, 공책, 책, 상자 플라스틱 표지, 스트링, 비닐, 테이프, 택배엽수종 등 이물질 제거, 반듯하게 펴서 자국자국 찢은 후 묶어서 배출 종이류와 종이배류는 연연의 구분	종이배류  우유배, 종이컵 내용물을 비우고 물로 헹군 후 배출 감열지(영수증), 금박지, 부직포는 일반 쓰레기	유리병류  에너지드링크병, 콜라병, 소주병 재질 다른 병뚜껑 분리, 내용물을 비운 후 배출 도자기, 거울, 식기는 불연성폐기물 전용봉투에 따로 배출
금속캔류  참치캔, 음료수캔 재질 다른 뚜껑 분리, 내용물을 비운 후 배출 부탄가스, 살충제용기는 구멍 뚫어 가스 배출 후 배출	플라스틱류  페트병, 세제병 내용물을 비우고, 부직포면 후 압착해 배출 빨대: 일반 쓰레기	비닐류  라면 비닐, 커피믹스 봉투, 입회용 봉투 음식물이 묻었으면 헹군 후 배출, 이물질 제거가 어려우면 일반 쓰레기 스티로폼, 합판면 용기: 가전제품 포장에 사용된 스티로폼은 구입처에 반납, 지자체에 재활용 여부 확인 필요

사진 2. 한국 분리수거 기준[3]

표 1. 데이터 셋 이미지 샘플





















				
(a)metal	(b)paper	(c) plastic	(d) metal	(e)glass
				
(f)paper	(g)metal	(i)glass	(j)paper	(k)paper

표 2. 이미지 데이터셋 정보

Recyclable material types	Number of material	training	test
paper	1039	784	255
glass	626	468	158
plastic	554	455	99
metal	549	449	100
TOTAL	2779	2158	621

2.2 데이터 전처리

표 3. 오답 데이터 셋 예시

				
(a)plastic	(b)paper	(c)glass	(d)metal	(e)paper
				
(f)plastic	(g)glass	(h)plastic	(i)glass	(j)glass

오답 데이터셋을 보여주는 코드[4]를 활용하여 오답을 확인해보았다. 위 사진과 같이 내용물이 경우 오답이 나왔다. 또한 물체가 여러 개인 경우나 라벨이나 워터마크가 있는 경우도 오답이 나왔다. 이

러한 오답 데이터셋은 학습이나 테스트 데이터로 쓰기에 맞지 않다고 판단하여 제외했다.

III. 재활용품 분류를 위한 모델 학습

3.1 모델 선정

본 연구에서는 자주 쓰이는 두가지 모델을 사용하였다.

VGGNet: VGGNet 은 2014 년 ILSVRC 에서 GoogLeNet 에 밀려 2 위를 했지만, 훨씬 간단한 구조로 이루어져 있고, 변형이 쉽다는 장점이 있어 많이 응용된 모델이다.

ResNet: ResNet 은 2015 년 ILSVRC 대회에서 우승한 모델이다. top-5 error 가 겨우 3.6%로 사람의 분류 수준인 5% 내외를 뛰어 넘었다. 이 모델은 데이터들의 흐름에 지름길과 비슷한 Residual 을 만들어 깊은 망을 사용할 때 학습이 되지 않는 문제점을 해결하였다. 본 연구에서 사용한 ResNet50 은 가장 일반적으로 쓰이는 모델 중 하나이며 컨볼루션 연산과 fully connected layer 만 계산 했을 때, 총 레이어 갯수가 50 개인 ResNet 모델을 뜻한다.

두 가지 모델의 성능 비교를 위해 통제변인을 고정시키고 간단한 조건 하에 학습을 시켰다.

표 1. VGG16 과 ResNet50 성능 비교

Model	Test accuracy	Epochs
VGG16	33%	20
VGG16	33%	35
VGG16	53%	50
ResNet50	37%	20
ResNet50	40%	35
ResNet50	60%	50

간단한 모델 테스트 결과 Test accuracy 가 비교적 높은 ResNet50 을 선정하였다.

3.2 모델 성능 개선

과적합된 모델의 성능을 개선하기 위한 여러 가지 방법을 ResNet50 에 적용시켜 보았다.

1) ImageAugmentation: 케라스의 ImageDataGenerator를 이용하여 트레이닝 셋 증강.

2) L1, L2 정규화: L1, L2 정규화를 동시에 사용했을 때와 L1 정규화만 적용시켰을 때는 학습이 되지 않고 local minima에 빠진 듯 보였다. 하지만 L2 정규화만 적용했을 때는 정확도가 4%가량 증가했다.

3) 데이터셋 재분류: 예러 데이터를 분석한 결과, L2 정규화를 적용하기 이전에는 Cardboard를 metal로 착각한 예러가 가장 많은 반면, L2 정규화를 적용시키자 카드보드의 FN 43 개중 38 개를 종이류로 착각한 것으로 나타났다. 또한 한국 재활용품 분류 기준을 보면 판지 또한 종이류에 포함된다. 따라서 데이터셋의 Cardboard, Paper를 Paper로 통합하였다.

4) Dropout: 유닛들간의 연결을 강제로 끊어, 일정 비율의 유닛들만 학습이 되게 함으로써 과적합을 방지하는 방법이다. 랜덤서치 범위를 (0.2, 0.7) 정도로 설정하고 돌려보았지만 학습속도 저하에 비해 효과가 미미했다. Batch normalization이 ResNet50 코드에 들어가 있어서 그런 듯 하다.

5) 하이퍼 파라미터 튜닝: Andrew Ng의 딥러닝 강좌에 따르면 하이퍼 파라미터 튜닝 시 중요도는 아래와 같다고 한다.

표 2. 하이퍼 파라미터 중요도

중요도	하이퍼 파라미터
1	learning rate
2	mini batch size
2	momentum term beta
기타	레이어 수, beta1, beta2, 등

momentum term beta는 momentum을 사용하는 경우만 해당되므로 중요도가 높은 mini batch size와 learning rate를 중점적으로 튜닝할 것이다.

Bergstra and Bengio가 쓴 논문 Random search for Hyper-Parameter Optimization에 따르면 그리드 서치(사람이 직접 제안한 값들을 생성하여 순차적으로 찾는 방법)보다는 랜덤 서치(값을 임의의 값으로 자동생성하여 찾는 방법)가 더 중요한 하이퍼파라미터 발견을 용이하게 하고 효율적이라고 한다.

Bayesian Hyperparameter Optimization 또한 하이퍼 파라미터 탐색을 효율적으로 하기 위해 고안된 방법이다. 하지만 ConvNet 관련된 실전 세팅에서는 아직 조심스레 선택된 구간에서의 랜덤서치가 상대

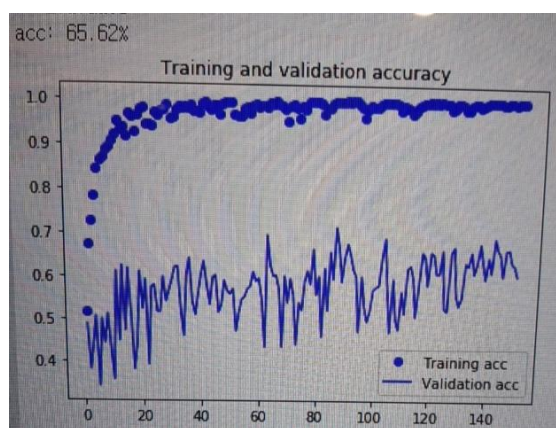
적으로 더 뛰어나다.

따라서 본 연구에서는 랜덤서치를 통해 learning rate를 튜닝하였다.

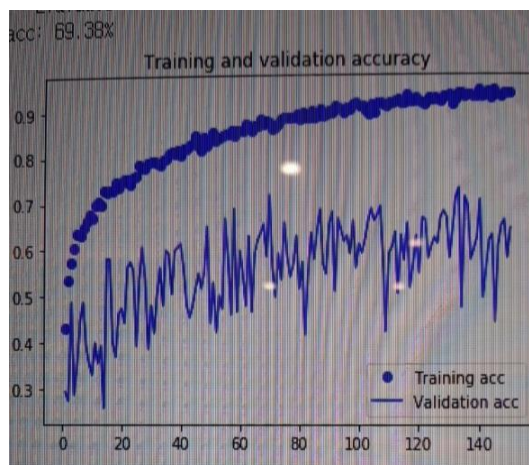
optimizer는 가장 안정적이고 성능이 좋다고 알려진 Adam optimizer를 이용했고 Adam에서 좋은 성능을 낸다는 (0.0005~0.00146)의 범위에서 랜덤서치를 했다.

IV. 실험 결과 및 분석

4.1 L2 정규화 전 후 비교 (클래스 5종류)



L2 정규화 적용 전 acc : 65.62%



L2 정규화 적용 후 acc = 69.38%

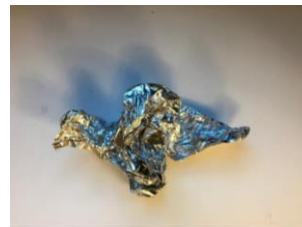
L2 정규화의 유무에 따라 accuracy가 3~5% 정도 변화하였다.

4.2 에러 이미지 분석 및 데이터셋 수정

클래스	예측값	정규화 전 FN 개수	정규화 후 FN 개수
Cardboard	Glass	8	1
	Metal	16	2
	Paper	14	38
	Plastic	9	1
Glass	Cardboard	5	4
	Metal	21	18
	Paper	7	15
	Plastic	22	24
Metal	Cardboard	5	5
	Glass	20	12
	Paper	8	13
	Plastic	13	6
Paper	Cardboard	6	5
	Glass	13	3
	Metal	11	5
	Plastic	4	1

Plastic	Cardboard	7	5
	Glass	30	22
	Metal	20	17
	Paper	8	20

오차행렬에서 FN 으로 판정된 이미지 데이터를 분석한 결과, 학습에 적용시키기 어려울 것이라고 생각되는 데이터가 많았다.(흑백으로 나누어진 배경, 2개 이상의 객체, 배경에 붙어있는 라벨, 클래스 분류가 잘못 된 사진 등). 또한 대부분의 이미지 배경이 흰색이라 그런지 test 데이터셋의 검은색 배경 이미지는 전부 에러데이터에 포함되어 있었다. 따라서 이처럼 학습에 적합하지 않다고 판정되는 데이터들을 Training set 과 Test set 에서 전부 제거하였다.



Metal



Cardboard

이와 같은 경계선이 울퉁불퉁한 데이터(ex 찢겨진 판지)를 Metal 로 판정하는 경우가 많았다. L2 정규화를 적용하고 난 뒤에는 Cardboard 에서의 Metal 판정이 눈에 띄게 줄어들었고 Cardboard 의 FN 중 가장 많은 예측값이 Paper 이고 반대로 Paper 의 경우에는 Cardboard 였다. 재활용품 분류기준에도 Paper 에 Cardboard 가 포함되기 때문에 두 클래스를 Paper 로 합쳤다.

4.2 Batch size 조정

Udacity deep learning 강좌 중 Jay alammr 의 Hyperparameters 강좌에 따르면 Batch size 의 범위는 32~256 (2 의 n 승)이 대체로 좋은 결과를 낸다고 한다. 따라서 32, 64, 128 순서로 Batch size 를 설정하여 결과를 비교해 보았다.

Batchsize 에 따른 정확도 변화

Batch size	test accuracy
32	69%
64	66%
128	60%

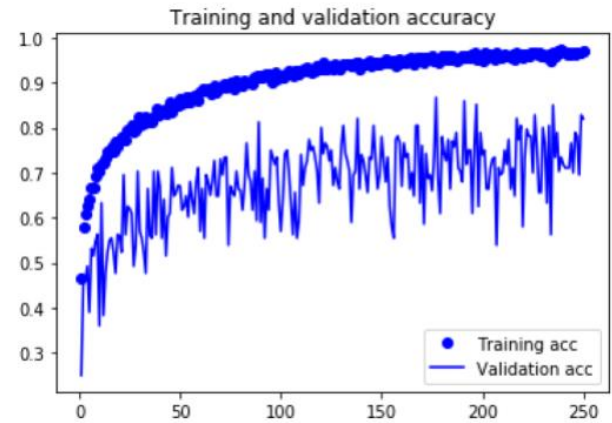
Batch size 가 32 에서 증가할수록 정확도가 낮아지는 추세를 보인다. 따라서 Batch size 값은 32 를 사용했다.

4.3 Learning rate 조정

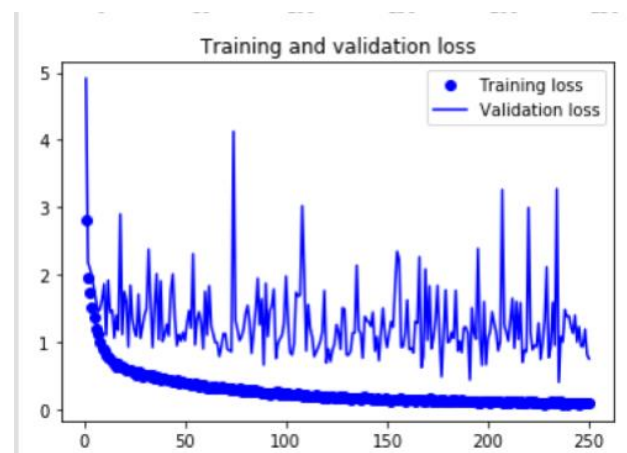
David Mack 의 How to pick the best learning rate for your machine learning project 에 따르면 6 가지 optimizer 중 Adam optimizer 가 가장 안정적이며, Adam 의 learning rate 는 0.0005, 0.001, 0.00146 이 좋은 성능을 보인다고 한다. 랜덤서치 범위는 0.0005 부터 0.00146 사이로 설정하고 후에는 0.0008 부터 0.0014 로 범위를 좁혔다.

Learning rate 에 따른 test accuracy 변화

Learning rate	test accuracy
0.00083	75%
0.00086	71%
0.00091	67%
0.001	69%
0.00113	72%
0.00119	77%
0.00123	78%
0.0013	75%



Best Accuracy



결과적으로 learning rate=0.00123일 때 정확도가 78.91%로 제일 높았다.

4.4 실제 시연



일상에서 보이는 다양한 형태의 재활용품들을 직접 촬영하고 이 이미지들을 학습된 모델에 넣어 시연 해보았다. 아쉽게도 학습할 때에 비하여 정확도가 많이 떨어지는 모습을 보여주었다.

정확도는 46.15%로 학습 대비 32.76%가 떨어졌다.

실제 재활용품은 인터넷에서 얻을 수 있는 데이터

터셋에 비해 물체 표면의 포장지나 형태가 가지각색이므로 정확도가 떨어지는 것 같다.

그리고 error data를 대부분 paper라고 판단했다.

종이에 들어가는 인쇄물 내용이 다양해서 분류기가 error data를 종이라고 판정한 것 같다.

V. 결론

인터넷에서 크롤링을 통해 얻은 데이터셋으로 ResNet50의 모델 성능을 최대한 올린 결과, 비슷한 주제의 기존 논문에서 ResNet을 사용한 accuracy보다 근소하게 높게 나왔다. 하지만 현실의 분리수거 물품과 비슷한 실제 데이터로 시연 시 정확도가 떨어지는 모습을 보여주었다. 그림과 문자가 많은 종이 클래스의 특성때문에 다른 클래스의 겉표면에 다양한 그림이 있는 경우, 인공지능이 종이라고 오판하기 쉬운 듯 하다. 그래서 슈퍼빈의 네프론처럼 캔과 페트병만을 분류하는 것이 아닌 다양한 종류의 재활용품을 정확하게 분류하는 것은 이미지만으로는 한계가 있을 것이다. 이러한 문제상황에서 재활용품에 대한 물리적인 측정 수단이 하나의 해결책이 될 것이라고 생각한다. 예를 들어, 무게 혹은 떨어질 때 소리의 파동등을 함께 측정한다면 더 정확한 분류가 가능하지 않을까 생각한다.

감사의 글

끝까지 열심히 한 우리 조원 모두 서로와 최유경 교수님께 감사드립니다.

참고문헌

- [1] Mindy Yang, Stanford's CS229: Machine Learning Summer 2019
- [2] Hardikvasa, google-images-download in github
- [3] 이시연, 조선일보 뉴스 Q(스티로폼과 생선뼈, 재활용 쓰레기? 일반 쓰레기?)
- [4] Reddit MLQuestions: Evaluate a multiclass classification problem confusion matrix
- [5] 파이썬으로 배우는 머신러닝의 교과서
- [6] 파이썬, 케라스로 배우는 대소나의 딥러닝 기초
- [7] 케라스 창시자에게 배우는 딥러닝
- [8] 머신러닝 실무 프로젝트
- [9] RecycleNet: Intelligent Waste Solting Using Deep Neural Networks
- [10] Random Search for Hyper parameter optimization