



딥 러닝을 통한 도로의 점자 블록 판별에 관한 연구

A Study on Braille Block Discrimination of Roads through Deep Learning

우성욱 (Seong Wook Woo)¹

¹ 한국외국어대학교, 컴퓨터·전자시스템공학부; woosw95@hufs.ac.kr


한글 요약: 데이터 분석을 활용하여 다양한 분야에 적용이 가능하다. 최근 사회 전반적으로 데이터 분석이 매우 중요하게 작용 되고 있다. 데이터 분석을 하기 위해서는 데이터를 잘 다루어야 한다. 데이터를 다루는 방법 중 한 가지 방법은 딥 러닝이다. 딥러닝 중에서 합성곱 신경망은 이미지 데이터를 처리하는데 유용하다. 그러므로 합성곱 신경망을 이용하여 도로 분석을 하는 시스템을 제안한다. 도로 분석을 할 때는 점자 블록의 유무로 도로를 분류 해준다. 점자 블록 유무에 따라 도로를 분류하기 위해서 데이터를 학습시킨다. 학습 된 모델을 바탕으로 실제 구글 지도의 로드 이미지를 갖고 와서 어느 카테고리에 속하는지 자동으로 분류를 해주도록 한다.

핵심어: 점자 블록, 인도 분석, 데이터 분석, 합성곱 신경망

영문 요약: Data analysis can be applied to various fields. Recently, data analysis has become very important in society. In order to do data analysis, you need to handle data well. One way to deal with data is deep learning. In deep learning, convolutional neural networks are useful for processing image data. Therefore, we propose a system for road analysis using convolutional neural networks. When analyzing the road, the road is classified according to the presence or absence of Braille blocks. Data is trained to classify roads according to the presence or absence of Braille blocks. Based on the learned model, it brings the load image of the actual Google map and automatically classifies which category it belongs to.

Keywords: braille block, road analysis, CNN

본 논문은 2021학년에 제출된
한국외국어대학교
컴퓨터·전자시스템공학부
졸업논문이다.

지도교수: 최재영 교수님 

서명: 우성욱

참여한 캡스톤 설계

설계명

▶ 스트리트 뷰 이미지와
딥 러닝을 통한 도로 정보 분석

팀원명

- ▶ 우성욱
- ▶ 김동규
- ▶ 굴람무르타자
- ▶ 블라디 신



Copyright: © 2021 by the authors.
Submitted for possible open
access publication under the
terms and conditions of the
Creative Commons Attribution
(CC BY) license
(<http://creativecommons.org/licenses/by/4.0/>).

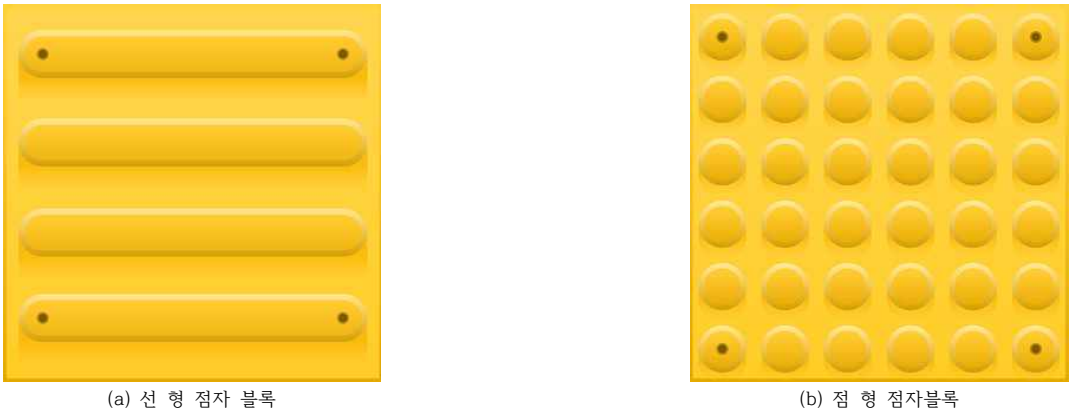
1. 서론 - Introduction

과거에 기업은 사람의 경험과 감에 의지해 비즈니스의 상태를 파악하는 경우가 많다. 하지만, 최근 사회 전반적으로 데이터 분석 분야가 주목 받고 있다. 데이터 분석이란 정형 및 비정형의 데이터를 분석하는 것을 의미할 수 있다. 현황 파악을 하기 위해서 필요한 정보를 데이터 분석을 하여 가시화 톨로 알기 쉽게 정리할 수 있다. 현재 데이터 분석으로 공공, 여행, 교통, 마케팅, 보험, 제약, 농업, 금융, 치안 등 다양한 분야에서 활용이 되고 있다. 데이터 분석이 활용된 사례로는 세계 주요 도시 건강생활지수[1], 인구 이동 현황 분석[2], 걷기 여행[3], 교통 운영 및 관리[4], 교통사고 분석 시스템[5], 카카오 광고 시스템[6], 보험료 할인 및 보험금 지급[7], PDB 서비스[8], 스마트 팜[9], 핀테크[10], 범죄 예측[11] 등이 있다. 또한, 실시간

교통 정보들이 생산되고 있으며 이러한 정보는 내비게이션[12]이나 대중교통 안내 서비스 등에서도 활용되고 있다. 데이터 분석이 중요하게 활용될 수 있는 분야 중 하나가 교통 분야이다. 현재 시점에서도 많은 실시간 교통 정보들이 만들어 지고 있으며, 이러한 정보들을 분석하여 내비게이션이나 대중교통 안내 시스템 등의 서비스 등에 활용되고 있다.

인공신경망 기법은 뇌의 작동 원리를 모방하는 방법으로 데이터의 패턴이나 특징을 인식하는데 필요한 모델을 생성하는 기법이다. 인공신경망기법 중 합성곱 신경망은 학습과정을 통해 분류하려는 데이터에 적합한 특징을 구분하여 자동으로 특징을 알아내고 결과를 분류할 수 있기 때문에 다양한 영상 및 이미지 판독 분야에 사용되고 있다.

점자 블록은 시각 장애인 유도 블록으로 시각 장애인이 안전하게通行 할 수 있도록 바닥에 설치되어 있는 울퉁불퉁한 블록이다. 점자 블록은 보행 분기점, 대기 점, 시발점, 종료지점 등의 위치를 표시하고 위험물이나 위험지역을 알리는 역할을 한다. 크게 <그림 1>와 같이 선 형 점자 블록이랑 같은 점 형 점자 블록이 있다.

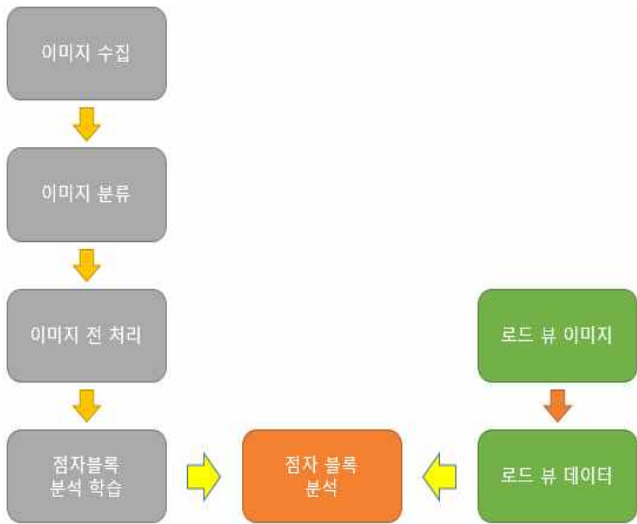


<그림 1> 점자블록 종류

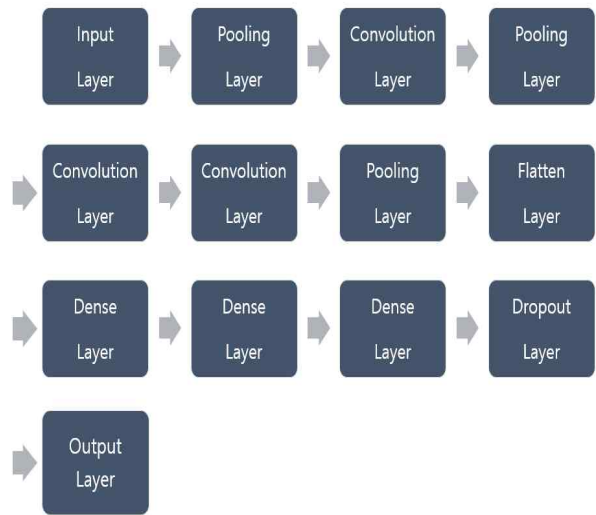
사람이 직접 일일이 직접 다니면서 확인하려면 많은 시간이 걸린다. 그래서 구글 지도를 활용하여 출발하는 지점과 도착하는 지점의 이미지를 취득하고, 딥 러닝 기술 중 합성곱 신경망을 통해 점자 블록이 존재 하는지 또는 존재하지 않는지 분류함으로써 취득된 도로 이미지를 자동 분류할 수 있는 알고리즘을 제시한다. 논문의 구성 방법으로는 2장에서는 구현에 대해서 서술한다. 3장에서는 결과에 대해서 작성하고 4장에서는 분석에 대해서 서술한다. 마지막으로 5장에서는 결론에 대해서 서술하겠다.

2. 구현

제안하는 시스템은 학습 데이터 수집, 데이터 학습, 실습 데이터 수집, 학습된 데이터로 실습 데이터 분석으로 구성된다. 먼저 학습 시킬 데이터를 수집한다. 그리고 나서 수집된 이미지를 전 처리 과정을 거쳐서 더 다양한 경우로 만든다. 전 처리 과정을 거친 데이터들을 학습을 시킨다. 그 다음으로 실습할 이미지를 수집한다. 수집된 이미지를 데이터화 해 가지고 기존에 학습 시킨 모델에 적용을 해서 실습 할 데이터가 어느 그룹에 속하게 되는지 분류시킨다. <그림 2>은 제안하는 시스템 원리를 보여준다.



<그림 2> 시스템 구성도



<그림 3> CNN 구조

2-1 이미지 수집

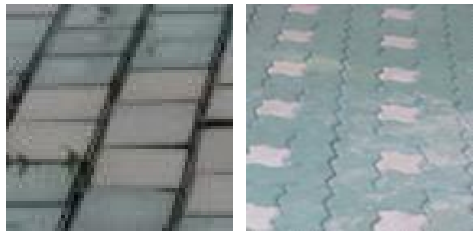
2021년 3월부터 약 1개월간 거리를 직접 돌아다니면서 사진을 찍은 이미지와 구글 이미지, 구글 스트리트 뷰 이미지를 본 연구에 활용하였다. 수집된 이미지는 점 형 점자블록 1,500장, 선형 점자블록 508장, 점자블록이 아닌 이미지 5,294장이다. 총 7,302장을 수집했다. 학습 데이터의 구성은 점 모양의 점자 블록, 선 모양의 점자 블록, 점자 블록이 아닌 데이터로 나누어 각각 구축하였다.



(a) 점 형 점자 블록



(b) 선 형 점자 블록



(a) 점자 블록이 아닌 도로

<그림 4> 수집된 이미지의 일부

2-2 이미지 전 처리

2-1에서 수집한 이미지를 토대로 다양한 이미지 데이터를 얻기 위해서 90° 회전, 180° 회전, 270° 회전, 사이즈 재설정, 밝기 조절, 명암 조절, 흑백 화, 노이즈 제거 등 다양한 전 처리 과정을 했다.

resize_dot (1494)
resize_dot (1495)
resize_dot (1496)
resize_dot (1497)
resize_dot (1498)
resize_dot (1499)
resize_dot (1500)
rotate180_dot (1)
rotate180_dot (11)
rotate180_dot (751)
rotate180_dot (752)
rotate180_dot (753)
rotate180_dot (754)
rotate180_dot (755)
rotate180_dot (756)
rotate180_dot (757)
rotate180_dot (758)
rotate180_dot (759)
rotate180_dot (760)

(a) 점 형 점자 블록

rotate180_line (505)
rotate180_line (506)
rotate180_line (507)
rotate180_line (508)
rotate270_line (1)
rotate270_line (2)
rotate270_line (3)
rotate270_line (4)
rotate270_line (5)
rotate270_line (6)
rotate270_line (7)
rotate270_line (8)
rotate270_line (9)
rotate270_line (10)
rotate270_line (11)
rotate270_line (12)
rotate270_line (13)
rotate270_line (14)

(b) 선형 점자 블록

grayscale_nb (529...)
grayscale_nb (529...)
grayscale_nb (529...)
grayscale_nb (529...)
invert_nb (1)
invert_nb (2597)
invert_nb (2598)
invert_nb (2599)
invert_nb (2600)
invert_nb (2601)
invert_nb (2602)
invert_nb (2603)
invert_nb (2604)
invert_nb (2605)
invert_nb (2606)
invert_nb (2607)
invert_nb (2608)

(c) 점자 블록이 아닌 도로

<그림 5> 전 처리 시킨 이미지들 중 일부

2-3 학습 모델 구축

전 처리 되어 있는 데이터들을 불러오고 합성곱 신경망을 구성하여 이미지를 학습 시킬 것이다. 합성곱 신경망에는 합성곱 계층, 풀링 계층, Dense 계층 등 다양한 계층이 존재한다. 합성곱 계층은 합성곱 연산을 처리한다. 합성곱 연산은 이미지 데이터 처리에서 필터 연산에 속한다. 이 계층에서는 입력된 데이터들의 특징을 파악할 수 있도록 도와준다. 풀링 계층은 세로, 가로 방향의 공간을 줄여주는 연산이다. 풀링 계층은 이미지 데이터의 가로 및 세로 데이터의 공간을 줄이는 연산 방법이다. 풀링 계층에는 최대 풀링(Max Pooling), 평균 풀링(Average Pooling), 최소 풀링(Min Pooling)등 여러 가지 풀링 방법이 있다. Max Pooling은 영역에서 최댓값을 계산하는 방법이고 Average Pooling은 영역의 평균을 계산하는 방법이며 Min Pooling은 영역에서의 최솟값을 계산하는 방법이다. 풀링 계층은 입력에 따른 변화가 적다. flatten 계층에서는 2차원으로 추출되어 있는 데이터를 1차원 데이터로 바꾸는 역할을 합니다. Dense 계층은 fully connected 계층이라고도 한다. 이 계층에서는 추출된 특징을 입력 받아 이미지를 적당한 카테고리로 분류한다. dropout 계층은 정규화를 위한 과정이다. dropout 계층은 과적합(over fitting)을 막기 위한 방법 중 한가지로 Dense Layer와 소프트 맥스 함수 중간에 있다.

본 논문에서 사용하는 CNN은 3개의 합성곱 계층과 3개의 pooling 계층, 1개의 flatten 계층, 3개의 Dense 계층, 1개의 dropout 계층, 1개의 입력 계층, 1개의 출력 계층으로 총 13개의 계층으로 이루어져 있다. 여기서 3개의 pooling 계층은 Max pooling 기법을 사용할 것이다. 이루어진 점자 블록 형태를 구분하는 것이 목적이기 때문에 출력 계층은 소프트 맥스 함수를 사용한다.

이 연구에서 활용된 훈련 데이터 개수 점자 블록이 아닌 이미지(0) 2,878장과 점 형 점자 블록 이미지(1) 2,878장과 선형 점자 블록 이미지(2) 2,743장을 활용하였다. 테스트 데이터 개수는 점자 블록이 아닌 이미지(0) 320장과 점 형 점자 블록 이미지(1) 320장과 선형 점자 블록 이미지(2) 305장을 활용하였다.

```

Layer (type)
=====
Input_Layer (Conv2D)
max_pooling2d (MaxPooling2D)
conv2d (Conv2D)
max_pooling2d_1 (MaxPooling2D)
conv2d_1 (Conv2D)
conv2d_2 (Conv2D)
max_pooling2d_2 (MaxPooling2D)
flatten (Flatten)
dense (Dense)
dense_1 (Dense)
dense_2 (Dense)
dropout (Dropout)
Output_Layer (Dense)
=====

```

<그림 6> 모델 요약

```

Train Value Counts
1    2878
0    2878
2    2743
dtype: int64
~~~~~
Test Value Counts
1    320
0    320
2    305
dtype: int64
~~~~~
X Train Shape
(8499, 150, 150, 3)
~~~~~
X Test Shape
(945, 150, 150, 3)
~~~~~
Train on 8499 samples, validate on 945 samples

```

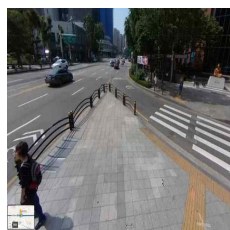
1:점 형 , 2 : 선형 , 0 : 점자 블록 x

<그림 7> 학습 모델 실행

1 epoch은 전체 데이터에 대해 한 번 학습을 완료한 상태이다. 학습 결과의 검증을 위해 검증 할 데이터를 생성한다. 검증 할 데이터와 학습 된 데이터를 비교하며 학습을 정해진 epoch 값까지 반복한다. epoch 값이 늘어날수록 학습 되는 데이터에 대한 오차가 줄어든다. 하지만, 배치 사이즈가 크면 처리해야할 양이 매우 방대하므로 학습 속도가 느려지며 시간이 오래 걸린다. 경우에 따라서 메모리가 부족할 수 있는 문제를 겪을 수도 있다. 또한, 너무 많은 epoch으로 예측이 심각하게 어긋나는 문제가 발생 할 수도 있는데 이러한 문제를 과적합(over fitting)이라고 한다. 반대로 너무 적은 epoch은 under fitting을 발생 할 수 있다. under fitting을 대비하여 epoch 값을 충분히 큰 값을 주면 된다. over fitting의 경우를 대비하여 epoch을 조기 종료 시킬 수 있는 알고리즘을 구현 해 주었다. 조기 종료 알고리즘은 생성된 검증 데이터가 매 epoch마다 검증 데이터에 대한 오류를 분석하여 모델의 훈련을 종료를 조정한다. 조기 종료가 안 된다고 epoch이 계속 진행이 된다면 훈련 손실과 검증 손실 둘 다 계속 줄어드는 게 아니고 훈련 손실은 꾸준히 감소하는 것을 보일 수 있지만, 검증 손실은 증가 할 가능성도 있다.

2-4 로드 뷰 이미 수집 및 데이터화

구글 API를 사용하여 GPX 파일을 읽음으로써 구글 지도에서 출발하는 지점과 도착하려고 하는 곳을 정해서 그 경로에 있는 위도랑 경도를 통해서 그 경로에 있는 이미지들을 자동으로 가지고 오는 시스템을 구현한다. 갖고 온 이미지를 데이터화한다.



<그림 8> 수집된 로드 뷰 이미지 중 일부

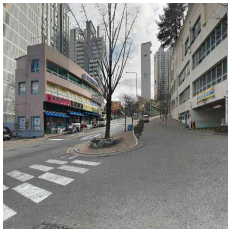
2-5 도로 분석

2-4에서 수집한 데이터를 가지고 기존에 학습 시킨 데이터를 기반으로 수집한 데이터를 입력을 하여 결과가 출력 되게 만든다. 결과는 출력 계층의 소프트 맥스(soft max)를 통해서 3가지 카테고리(선 형

점자블록, 점 형 점자 블록, 점자 블록이 없는 경우) 중에서 어느 카테고리에 속하는지 분류한다. 또한, 이때 분류 하였을 때, 몇 퍼센트 정도의 정확성을 가지는지를 같이 출력한다.

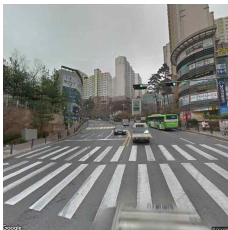
3. 결과

학습용 데이터의 생성을 위해 2가지 모양의 점자 블록과 점자 블록이 아닌 이미지를 분류하기 위해서 구현했다. 학습 결과의 검증을 위해 검증데이터를 생성한다. 학습 후에 분류 결과가 해당 카테고리에 속하는지에 대하여 확인하며 이와 같은 방법으로 학습을 반복한다. 모든 학습데이터에 대한 가중치 갱신 과정과 검증 데이터에 대한 검증과정을 반복하여 올바른 결과가 나오도록 학습과정을 반복하였다. 훈련 데이터와 테스트 데이터를 이용하여 분석한 결과 중 일부는 <그림 9>와 같다.



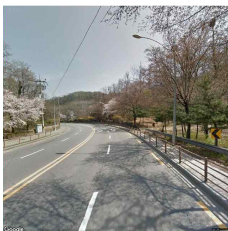
Prediction: 점자 블록이 없을 경우 99% Confident

(a) 점자 블록이 없는 도로 예측



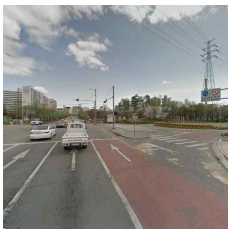
Prediction: 점자 블록 존재 87% Confident

(b) 점자 블록이 존재하는 도로 예측



Prediction: 점자 블록 없음 44% Confident

(c) 점자 블록이 없는 도로 예측



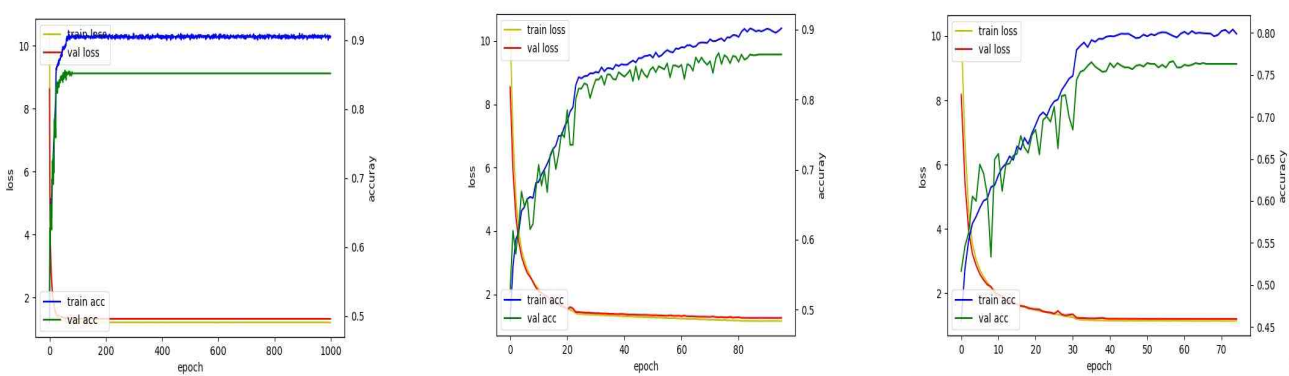
Prediction: 점자 블록 존재 69% Confident

(d) 점자 블록이 존재하는 도로 예측

<그림 9>

4. 분석

여러 번 실행을 했을 때 <그림 10>와 같은 정확도가 나오는 것을 알 수 있습니다. 하지만, 프로그램을 실행하였을 때, 대략, <그림 10>(a)와 같은 결과물을 얻기 위해서는 59시간이라는 시간이 걸렸습니다. 반면에 조기 종료 알고리즘으로 인해서 <그림 10>(b)의 결과물을 얻기 위해서 대략 5시간이 걸렸고, <그림 10>(c)의 결과물을 얻기 위해서는 4시간이 걸렸습니다. 매 번 실행 하였을 때 조기 종료 되는 값이 달라지며 정확도 또한 달라진다. <그림 10>(a)와 같이 일정 epoch이 지나면 훈련과 검증의 정확도가 하나의 값으로 수렴하게 되는 것을 알 수 있으며, 변화가 미미한 것을 알 수 있다. 그러므로 무조건 epoch 값이 큰 것이 좋지 않을 것을 알 수 있다. 하지만, 적당한 epoch 값이 언제인지 몰라서 조기 알고리즘이 필수 인 것을 알 수 있다.



(a) 조기 종료 없이 epoch 값이 1000인 경우 (b) 조기 종료로 epoch 값이 95일 때 (c) 조기 종료로 epoch 값이 70일 때

<그림 10> epoch 값에 따른 정확도



(a) 조기 종료 없이 epoch 값이 1000인 경우 (b) 조기 종료로 epoch 값이 95일 때

<그림 11> epoch 값에 따른 걸린 시간 (단위 : 초)

5. 결론 및 토론

이미지 처리 분류에는 다양한 기법이 있다. 그 중에서 딥 러닝을 이용한 합성곱 신경망(CNN)은 이미지 처리 분야에서 우수한 성능을 보인다. 구글 지도의 이미지를 입력으로 하여 도로에 점자 블록 유무를 분류할 수 있었다. 3개의 합성곱 계층과 3개의 pooling 계층, 1개의 flatten 계층, 3개의 Dense 계층, 1개의 dropout 계층, 1개의 입력 계층, 1개의 출력 계층을 구성하여 이미지의 특징을 추출하고 구글 지도의 이미지를 분류 할 수 있었다. 합성곱 계층과 풀링 계층을 여러 개 사용 하여 영상의 특징을 추출하고 flatten 계층을 사용하여 2차원으로 추출되어 있는 데이터를 1차원 데이터로 바꾸고 dense 계층을 사용하여 점자 블록이 있는지 없는지를 구분하였으며 dropout 계층을 이용하여 과 적합을 막았다. 또한, 조기 종료 알고리즘을 넣어서 추가적으로 over fitting 된 것을 막았다. 학습 데이터로는 다양한 점자블록 이미지랑 점자블록이 아닌 이미지를 사용함으로써 학습시켰다.

<그림 9>처럼 분류가 잘 될 때가 다수이긴 하지만, <그림 10>의 훈련 정확도를 보면 학습 정확도가 100%가 아니기 때문에 이미지 분류를 하였을 때, 완벽하게 분류가 안 될 수 있다. 또한, <그림 9>(c)와 같이 정확도가 50% 이하인 경우도 있다. 여러 원인 중 한 가지 이유로는 2-1에서 수집한 이미지가 충분하게 많지 않다고 판단이 되어 더욱 더 많은 이미지를 수집하여 학습시키면 분류 성능을 개선 할 수 있을 거라고 생각이 된다. 추후에 더 많은 이미지를 수집하여 정확도를 더욱 더 높인다면 점자 블록 유무에 대해서 자동 분류에 활용 할 수 있을 것이다.

6. 참고문헌 - References.

1. https://www.hani.co.kr/arti/economy/economy_general/981055.html [뉴스]
2. 부산광역시 빅 데이터 플랫폼
https://bigdata.busan.go.kr/visualztnBigDataAnals/bigDataAnalsListView.do?bigDataCl=big_data_cl_04&&menuCode=M000000000009 [인터넷 자료]
3. <https://blog.naver.com/theimc/222037066428> [인터넷 자료]
4. <https://www.bloomberg.com/news/articles/2017-05-11/new-york-city-s-vision-zero-traffic-exposure-model>
[인터넷 자료]
5. http://taas.koroad.or.kr/gis/mcm/mcl/initMap.do?menuId=GIS_GMP_ABS [인터넷 자료]
6. https://mk.kakaocdn.net/dn/if-kakao/conf2019/%EB%B0%9C%ED%91%9C%EC%9E%90%EB%A3%8C_2019/T03-S05.pdf [인터넷 자료]
7. <https://news.naver.com/main/read.naver?mode=LSD&mid=sec&sid1=001&oid=092&aid=0002146973>
[뉴스]
8. https://www.pharmabigdata.co.kr/PBD_brochure_200303.pdf [인터넷 자료]
9. <https://www.smartfarmkorea.net/file/download.do?fileId=864&type=BBS> [인터넷 자료]
10. 최기우 "핀테크와 빅데이터 기술에 대한 리뷰." 한국빅데이터학회 학회지 1.1 (2016): 77-84.[논문]
11. <https://www.tta.or.kr/data/androReport/ttaJnal/173-3-1.pdf> [인터넷 자료]
12. <https://www.hankookilbo.com/News/Read/201611050421476822> [인터넷 자료]