GTSRB - German Traffic Sign Recognition Benchmark

교통 표지판 이미지 분류 프로젝트

김혜수

개발 환경

OS : Window 10 Pro

개발 언어: Python 3.10

Library & open source: Tensorflow 2.0, Keras, Pandas, Numpy, OpenCV, Matplotlib

프로젝트 개요

프로젝트 주제 및 선정 배경
- 교통 표지판은 도로에서 가치있는 정보를 제공
- 도로 교통 표지판은 카메라에 정확하게 인식할 필요가 있음

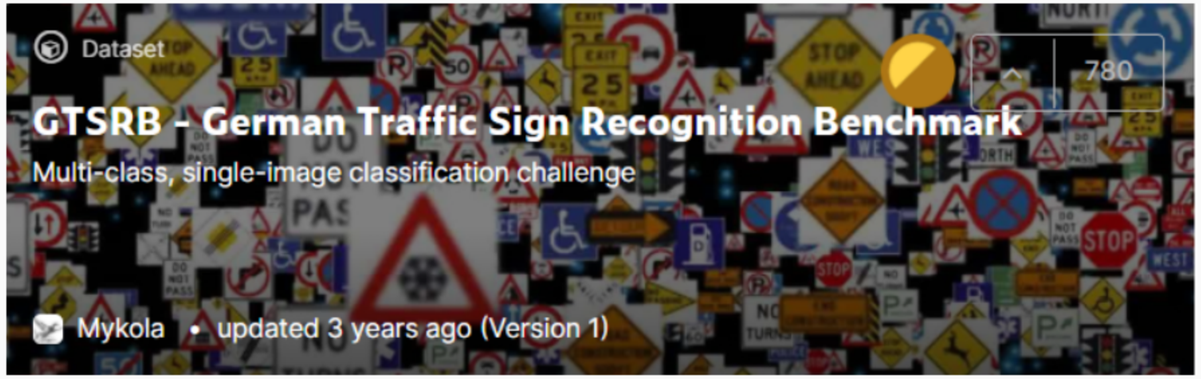
프로젝트 개요
- 교통 표지판 이미지 데이터를 분석하고 딥러닝 모델을 통하여 표지판 종류를 예측하는 분류 모델 수행

기대 효과
- 모델이 교통 표지판을 정확하게 인식하고 분류하여 자율 주행 시 교통 표지판 인식에 도움을 준다

프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획	> 4/21 ~4/22	-프로젝트 기획 및 주제 선정 -기획 안 작성	아이디어 선정
데이터 수집	> 4/22	-필요 데이터 서치(케글)	
데이터 전처리	> 4/25 ~4/26	-데이터 셋 확인 및 정규화 -이미지 증강	
모델 구축	> 4/27 ~4/28	-CNN 모델 구축 -평가지표 수립	최적화
테스트 및 시연	> 4/28 ~ 4/29	-저장된 모델 카메라로 연결후 시연	오류 수정

1. 학습 데이터 소개



GTSRB - German Traffic Sign Recognition Benchmark

- Multi-class, single-image classification problem
- 43 classes in total
- More than 50,000 images in total
- Large, lifelike database

2. 데이터 전처리

1. grayscale

2. normalization

3. data generator

```
width_shift_range= 0.1,
height_shift_range= 0.1,
zoom_range= 0.2,
shear_range=0.1,
rotation_range=10
```

3. 학습 모델 파라미터 및 모델 요약

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 60)        1560

 conv2d_1 (Conv2D)           (None, 24, 24, 60)        90060

 max_pooling2d (MaxPooling2D  (None, 12, 12, 60)       0
 )

 conv2d_2 (Conv2D)           (None, 10, 10, 30)        16230

 conv2d_3 (Conv2D)           (None, 8, 8, 30)          8130

 max_pooling2d_1 (MaxPooling  (None, 4, 4, 30)         0
 2D)

 dropout (Dropout)           (None, 4, 4, 30)          0

 flatten (Flatten)           (None, 480)               0

 dense (Dense)               (None, 500)               240500

 dropout_1 (Dropout)         (None, 500)               0

 dense_1 (Dense)             (None, 43)                21543

=================================================================
Total params: 378,023
Trainable params: 378,023
Non-trainable params: 0
```

파라미터

batch_size_val = 50

steps_per_epoch_val = 100

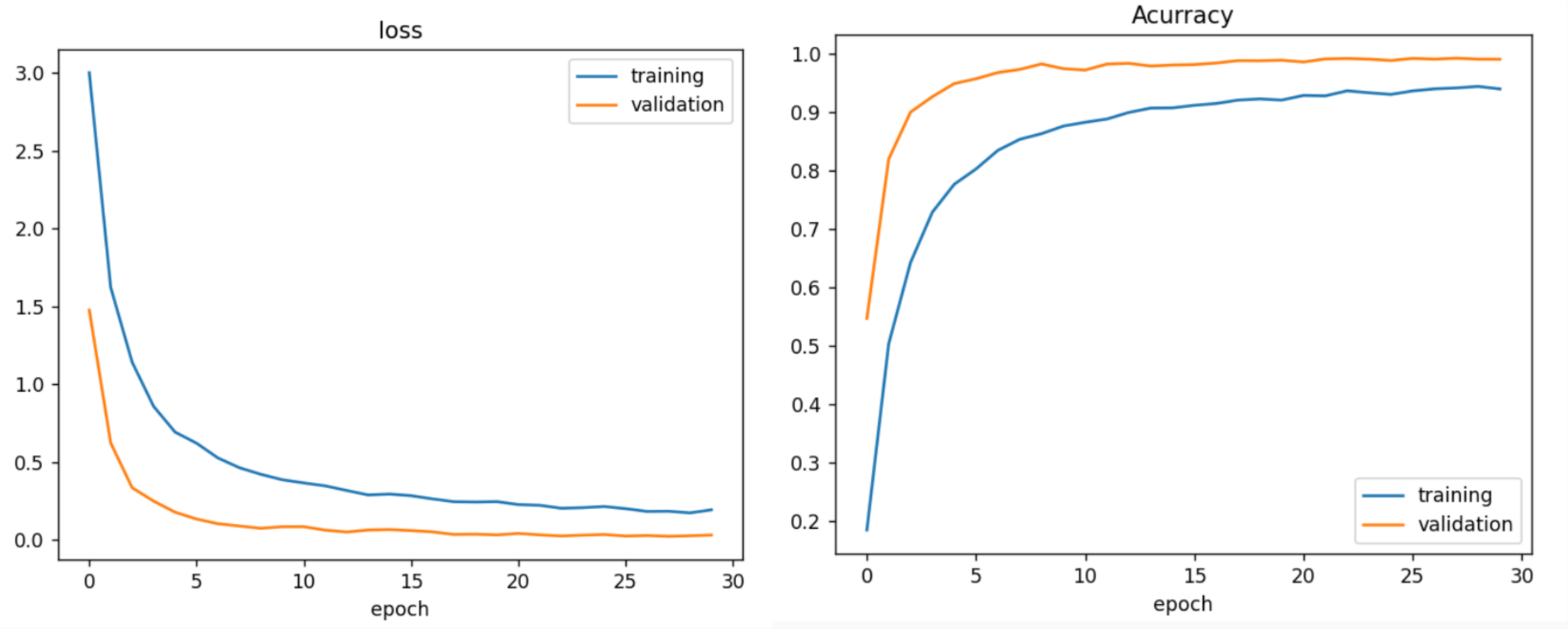
epochs_val =10

imageDimesions = (32, 32, 3)

testRatio = 0.2

validationRatio = 0.2

4. CNN 모델 정확도




```
Test Score: 0.024224549531936646
Test Accuracy: 0.9920976758003235
```

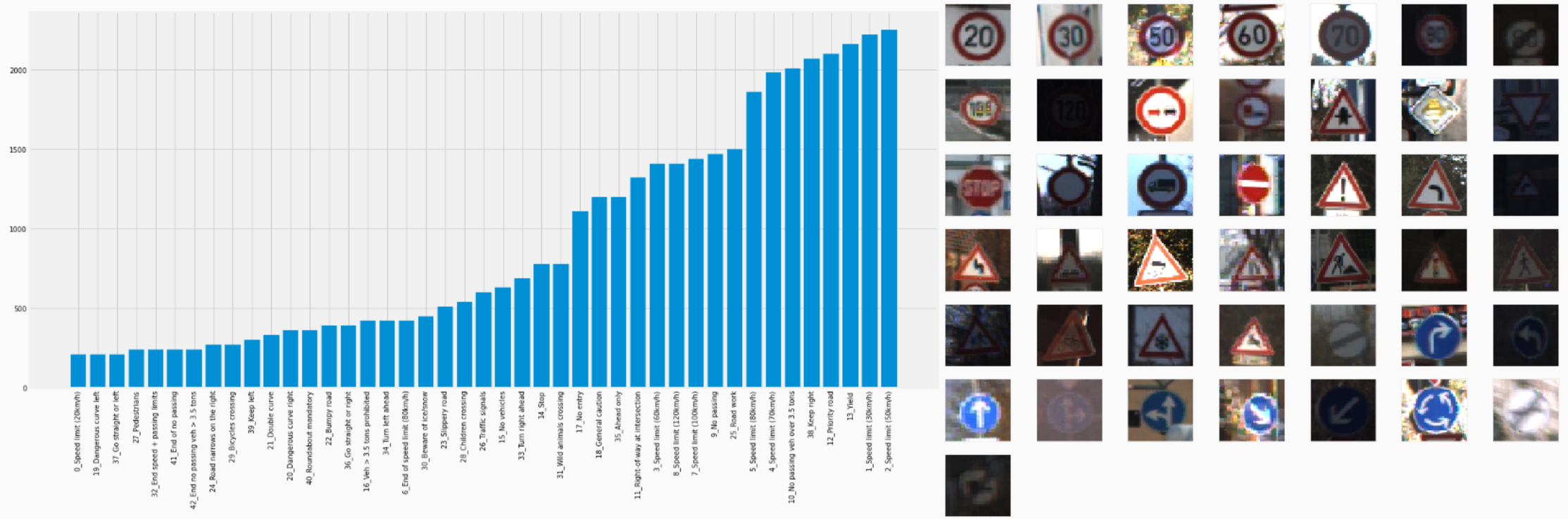
모델의 크기가 조금만 커져도 돌아가지 않고 다운되었기 때문에 모델을 작게 만듦
> 테스트 정확도도 99%로 높은 정확도를 보였음

4. 프로젝트 수행 결과

모델 시연 영상

유튜브 링크
https://youtu.be/GVTeDUWCv9k

5. 문제점



문제. 실제 시연했을시 실제 이미지(=새로운 이미지)에 대해 낮은 정확성을 보임

원인 1. 이미지 증강을 했음에도 데이터가 굉장히 불균형한 상태로 학습
원인 2. 더 다양한 환경을 가진 이미지로 학습이 필요(낮, 밤, 눈, 비, 흐린날 등등)

추후 발전 방향

1. 주피터에서 상대적으로 무거운 모델을 수행 후 model.save한 후 그대로 로컬로 가져와 시연 파일에서 실행
(코렙은 opencv를 사용할 수 없습니다.)

2. 일부 클래스의 데이터 수를 늘리고 다양한 모델 생성 후 비교하여 정확도가 높은 모델로 Test진행

THANK YOU!