지능형 시스템

(교통 표지판 분류)

17681005 김규영
17681026 장정우

nt

목차 a table of contents

Part 1,
프로젝트 개요

Part 1

프로젝트 개요(주제소개)

'교통 표지판 분류 및 인식'

딥러닝 교과목에서 진행한 최종 프로젝트에서 진행한 '자동차 번호판' 인식 프로그램과 연관 시켜 '교통 표지판 분류 및 인식'을 하면 자율주행 자동차들이 교통 법규를 지키며 안전한 운행을 할 것이라는 기대를 가지고 주제를 선정 하였습니다.

Part 1

프로젝트 개요(팀원 역할)

김규영

데이터 전 처리

정확도 분석
시각화

PPT작성

공통

주제 선정

아이디어 공유

데이터 선택

장정우

데이터 학습 코드
작성

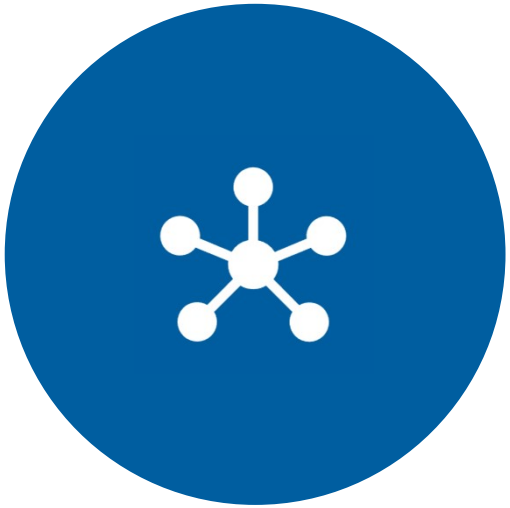
정확도 분석

과제 결과 보고서

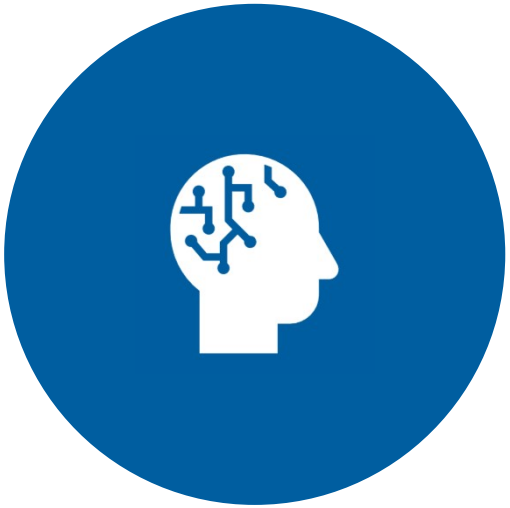
Part 1

제목을 입력하세요

Kaggle – GTSRB 데이터



CNN 모델

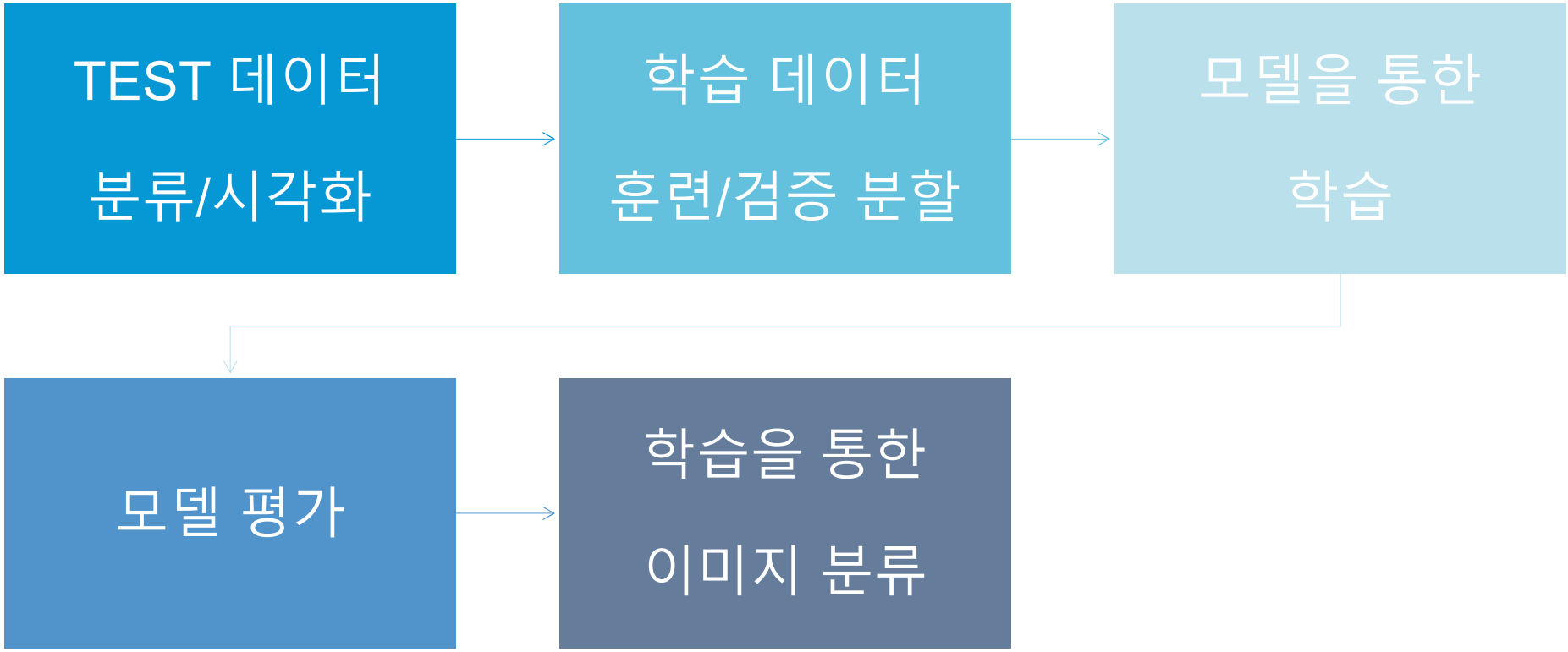


머신 러닝

Part 2,
프로젝트 설명

Part 2

프로젝트 설명(순서도)



Part 2

프로젝트 설명

In [1]:

```
import numpy as np
import pandas as pd
import os
import cv2
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
from PIL import Image
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import accuracy_score
np.random.seed(42)

from matplotlib import style
style.use('fivethirtyeight')
```

프로젝트 진행 시 필요한 라이브러리 import code

Part 2

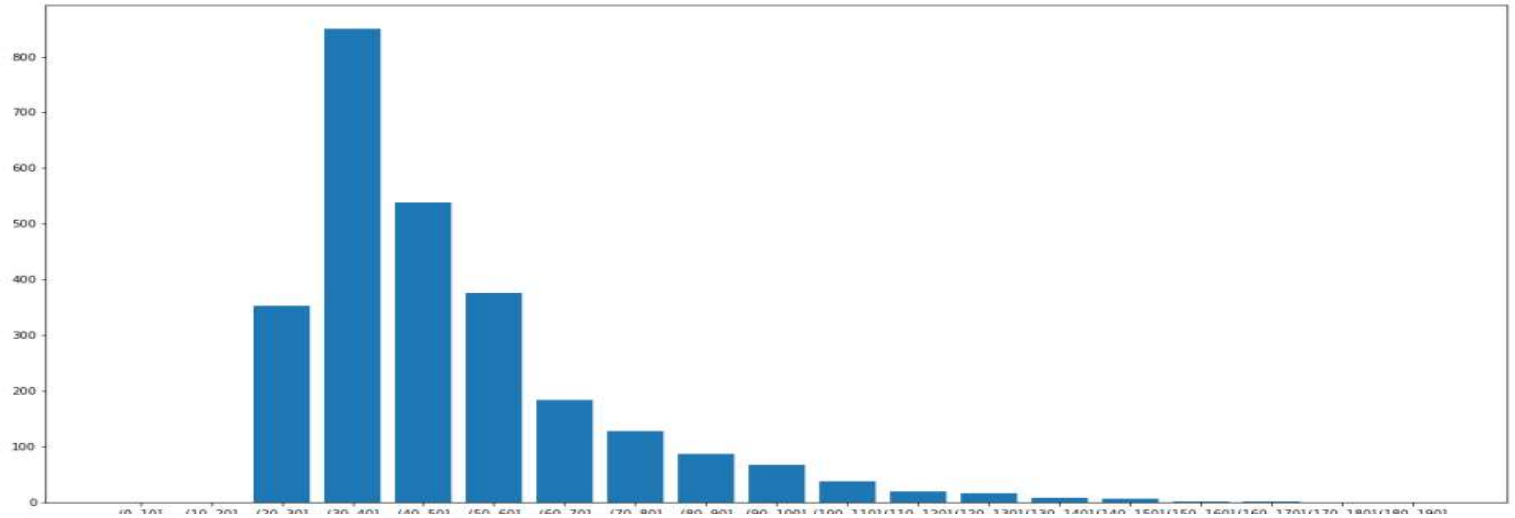
프로젝트 설명

이미지 크기 분포 확인

In [69]:

```
df_cutWidth = pd.cut(df_Train['Width'], np.arange(0,200,10)).value_counts(sort=False

fig, ax = plt.subplots(figsize=(20,10))
ax.bar(range(len(df_cutWidth)),df_cutWidth.values)
ax.set_xticks(range(len(df_cutWidth)))
ax.set_xticklabels(df_cutWidth.index)
fig.show()
```



Part 2

프로젝트 설명

In [2]:

```
data_dir = '../input/gtsrb-german-traffic-sign'
train_path = '../input/gtsrb-german-traffic-sign/Train'
test_path = '../input/gtsrb-german-traffic-sign/'

# Resizing the images to 30x30x3
IMG_HEIGHT = 30
IMG_WIDTH = 30
channels = 3
```

```
classes = { 0:'Speed limit (20km/h)',
            1:'Speed limit (30km/h)',
            2:'Speed limit (50km/h)',
            3:'Speed limit (60km/h)',
            4:'Speed limit (70km/h)',
            5:'Speed limit (80km/h)',
            6:'End of speed limit (80km/h)',
            7:'Speed limit (100km/h)',
            8:'Speed limit (120km/h)',
            9:'No passing',
            10:'No passing veh over 3.5 tons',
            11:'Right-of-way at intersection',
            12:'Priority road',
            13:'Yield',
            14:'Stop',
            15:'No vehicles',
            16:'Veh > 3.5 tons prohibited',
            17:'No entry',
            18:'General caution',
            19:'Dangerous curve left',
            20:'Dangerous curve right',
            21:'Double curve',
            22:'Bumpy road',
            23:'Slippery road',
            24:'Road narrows on the right',
            25:'Road work',
            26:'Traffic signals',
            27:'Pedestrians',
            28:'Children crossing',
            29:'Bicycles crossing',
            30:'Beware of ice/snow',
            31:'Wild animals crossing',
            32:'End speed + passing limits',
            33:'Turn right ahead',
            34:'Turn left ahead',
            35:'Ahead only',
            36:'Go straight or right',
            37:'Go straight or left',
            38:'Keep right',
            39:'Keep left',
            40:'Roundabout mandatory',
            41:'End of no passing',
            42:'End no passing veh > 3.5 tons' }
```

Kaggle에서 받은 GTSRB 데이터 읽기 및 이미지 파일 크기 통일화

클래스 번호를 사전의 교통 표지판 이름에 매핑하는 과정.

Part 2

프로젝트 설명

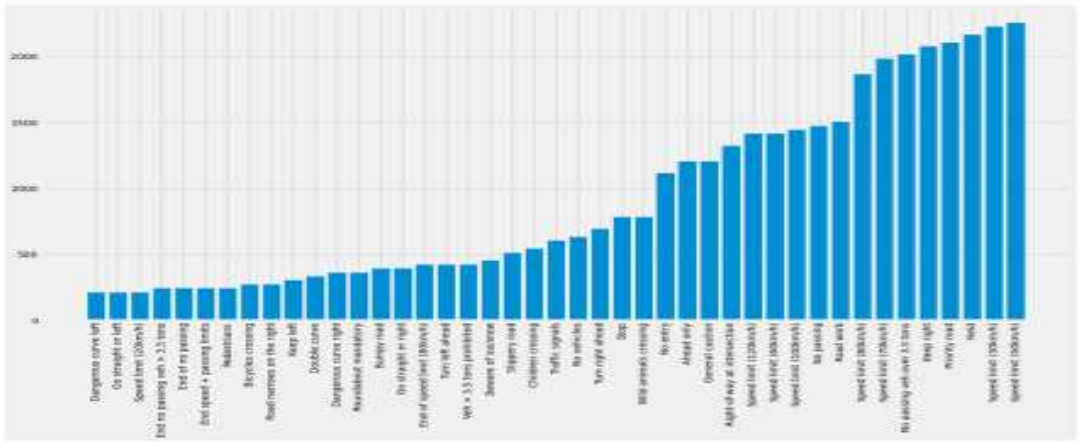
```
folders = os.listdir(train_path)

train_number = []
class_num = []

for folder in folders:
    train_files = os.listdir(train_path + '/' + folder)
    train_number.append(len(train_files))
    class_num.append(classes[int(folder)])

# Sorting the dataset on the basis of number of images in each class
zipped_lists = zip(train_number, class_num)
sorted_pairs = sorted(zipped_lists)
tuples = zip(*sorted_pairs)
train_number, class_num = [ list(tuple) for tuple in  tuples]

# Plotting the number of images in each class
plt.figure(figsize=(21,10))
plt.bar(class_num, train_number)
plt.xticks(class_num, rotation='vertical')
plt.show()
```



각 클래스 별 이미지 수를 기준으로 데이터 세트 정렬 및 각클래스 이미지 수 시각화

Part 2

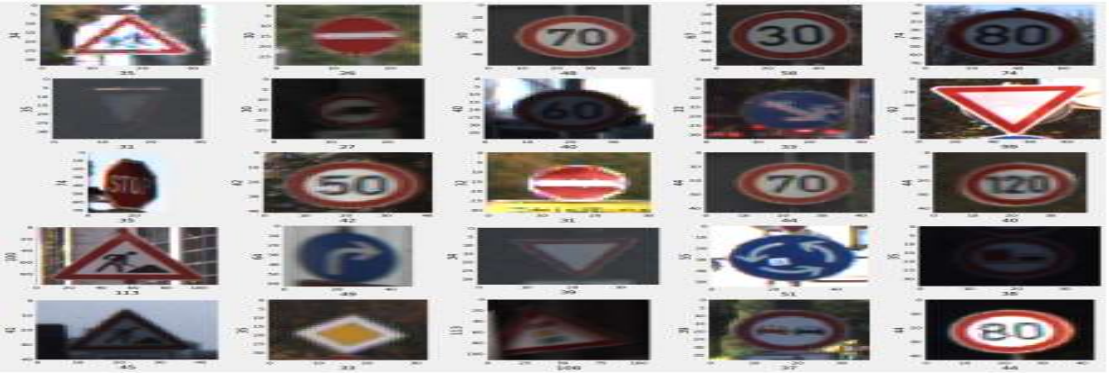
프로젝트 설명

```
# Visualizing 25 random images from test data
import random
from matplotlib.image import imread

test = pd.read_csv(data_dir + '/Test.csv')
imgs = test["Path"].values

plt.figure(figsize=(25,25))

for i in range(1,26):
    plt.subplot(5,5,i)
    random_img_path = data_dir + '/' + random.choice(imgs)
    rand_img = imread(random_img_path)
    plt.imshow(rand_img)
    plt.grid(b=None)
    plt.xlabel(rand_img.shape[1], fontsize = 20)#width of image
    plt.ylabel(rand_img.shape[0], fontsize = 20)#height of image
```



무작위 이미지
데이터 시각화

Part 2

프로젝트 설명

```
image_data = []
image_labels = []

for i in range(NUM_CATEGORIES):
    path = data_dir + '/Train/' + str(i)
    images = os.listdir(path)

    for img in images:
        try:
            image = cv2.imread(path + '/' + img)
            image_fromarray = Image.fromarray(image, 'RGB')
            resize_image = image_fromarray.resize((IMG_HEIGHT, IMG_WIDTH))
            image_data.append(np.array(resize_image))
            image_labels.append(i)
        except:
            print("Error in " + img)

# Changing the list to numpy array
image_data = np.array(image_data)
image_labels = np.array(image_labels)

print(image_data.shape, image_labels.shape)
```

```
shuffle_indexes = np.arange(image_data.shape[0])
np.random.shuffle(shuffle_indexes)
image_data = image_data[shuffle_indexes]
image_labels = image_labels[shuffle_indexes]
```

학습용 데이터 수집
훈련용 데이터 섞기

Part 2

프로젝트 설명

```
model = keras.models.Sequential([
    keras.layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', input_shap
    keras.layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu'),
    keras.layers.MaxPool2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(axis=-1),

    keras.layers.Conv2D(filters=64, kernel_size=(3,3), activation='relu'),
    keras.layers.Conv2D(filters=128, kernel_size=(3,3), activation='relu'),
    keras.layers.MaxPool2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(axis=-1),

    keras.layers.Flatten(),
    keras.layers.Dense(512, activation='relu'),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(rate=0.5),

    keras.layers.Dense(43, activation='softmax')
])
```

```
lr = 0.001
epochs = 30

opt = Adam(lr=lr, decay=lr / (epochs * 0.5))
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
```

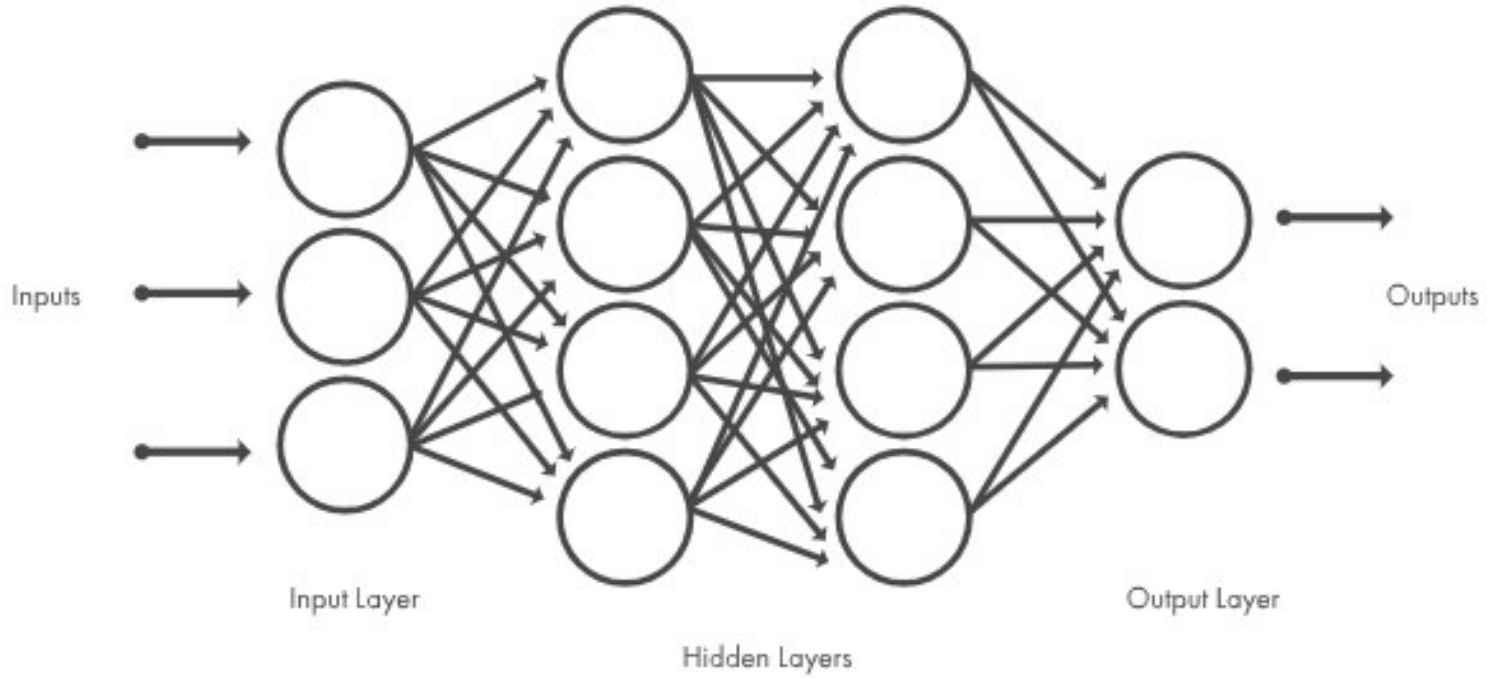
학습 모델 구성

CNN 모델

CNN 모델?

Part 2

프로젝트 설명



학습 모델을 통한 학습 중

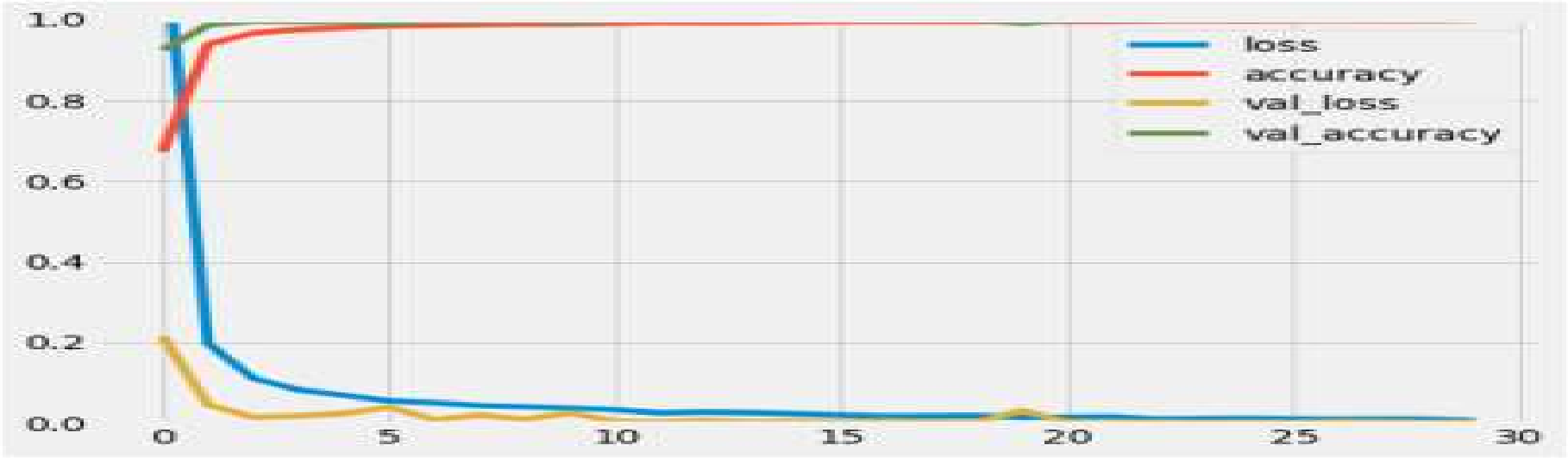
```
aug = ImageDataGenerator(
    rotation_range=10,
    zoom_range=0.15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.15,
    horizontal_flip=False,
    vertical_flip=False,
    fill_mode="nearest")

history = model.fit(aug.flow(X_train, y_train, batch_size=32), epochs=epochs, valida
```

```
Epoch 1/30
858/858 [==============================] - 19s 23ms/step - loss: 1.218
0 - accuracy: 0.6729 - val_loss: 0.2146 - val_accuracy: 0.9243
Epoch 2/30
858/858 [==============================] - 18s 21ms/step - loss: 0.197
4 - accuracy: 0.9384 - val_loss: 0.0455 - val_accuracy: 0.9856
Epoch 3/30
858/858 [==============================] - 21s 24ms/step - loss: 0.111
3 - accuracy: 0.9662 - val_loss: 0.0144 - val_accuracy: 0.9953
Epoch 4/30
858/858 [==============================] - 20s 23ms/step - loss: 0.082
3 - accuracy: 0.9746 - val_loss: 0.0179 - val_accuracy: 0.9943
Epoch 5/30
858/858 [==============================] - 19s 22ms/step - loss: 0.068
9 - accuracy: 0.9795 - val_loss: 0.0235 - val_accuracy: 0.9934
```

학습 모델 평가

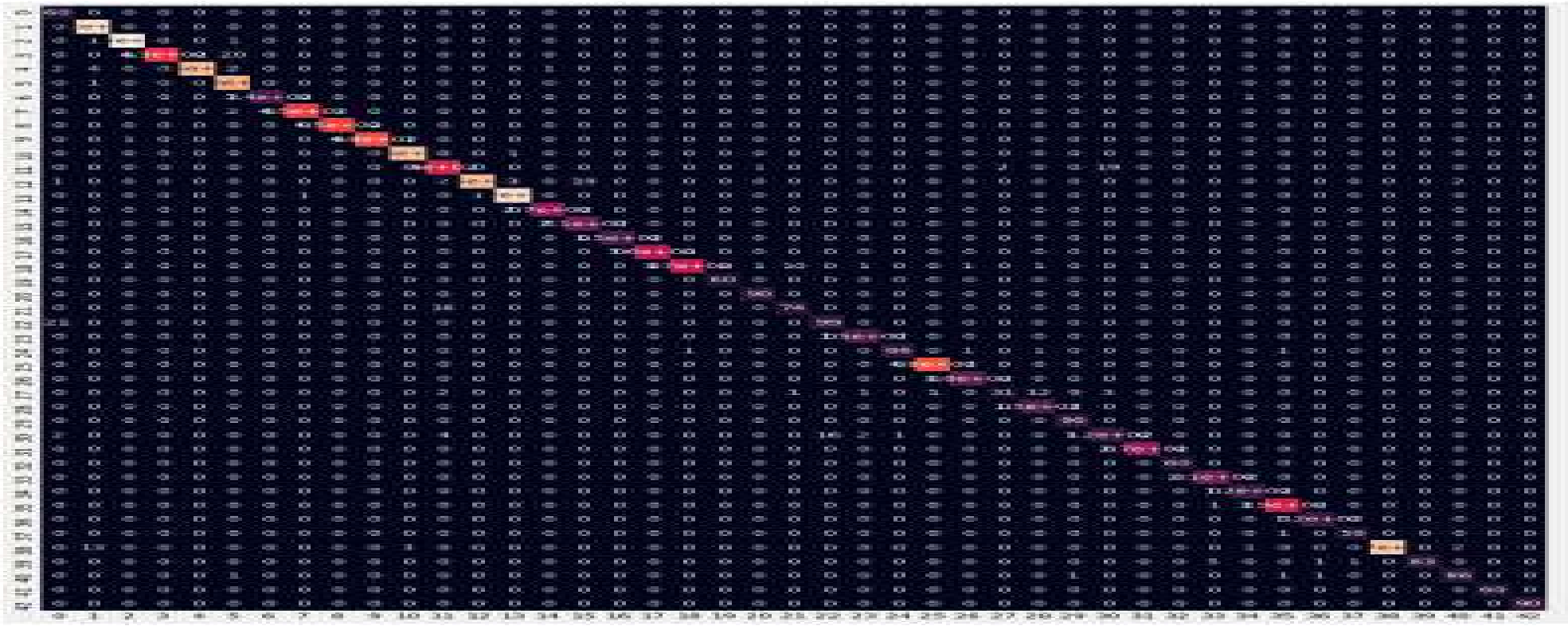
```
pd.DataFrame(history.history).plot(figsize=(8, 5))
plt.grid(True)
plt.gca().set_ylim(0, 1)
plt.show()
```



Part 2

프로젝트 설명

```
import seaborn as sns
df_cm = pd.DataFrame(cf, index = classes,  columns = classes)
plt.figure(figsize = (20,20))
sns.heatmap(df_cm, annot=True)
```



Part 2

프로젝트 설명

```
from sklearn.metrics import classification_report

print(classification_report(labels, pred))
```

	precision	recall	f1-score	support
0	0.70	1.00	0.82	60
1	0.97	1.00	0.99	720
2	1.00	1.00	1.00	750
3	1.00	0.96	0.98	450
4	1.00	0.99	0.99	660
5	0.96	1.00	0.98	630
6	1.00	0.96	0.98	150
7	1.00	0.99	1.00	450
8	1.00	1.00	1.00	450
9	1.00	1.00	1.00	480
10	0.99	1.00	1.00	660
11	0.94	0.95	0.95	420
12	1.00	0.96	0.98	690
13	0.99	1.00	1.00	720
14	1.00	1.00	1.00	270
15	0.92	1.00	0.96	210
16	1.00	1.00	1.00	150
17	1.00	1.00	1.00	360
18	1.00	0.96	0.98	390
19	1.00	1.00	1.00	60
20	0.98	1.00	0.99	90
21	0.87	0.82	0.85	90
22	0.86	0.82	0.84	120
23	0.97	1.00	0.99	150
24	0.99	0.96	0.97	90
25	1.00	1.00	1.00	480
26	0.99	1.00	0.99	180
27	0.94	0.52	0.67	60
28	0.91	0.98	0.95	150
29	0.87	1.00	0.93	90
30	0.85	0.83	0.84	150
31	1.00	1.00	1.00	270
32	1.00	1.00	1.00	60
33	0.97	1.00	0.99	210
34	0.97	1.00	0.98	120
35	0.99	0.99	0.99	390
36	0.98	1.00	0.99	120
37	0.98	0.98	0.98	60
38	1.00	0.97	0.99	690
39	1.00	0.92	0.96	90
40	0.96	0.96	0.96	90
41	1.00	1.00	1.00	60
42	0.97	1.00	0.98	90
accuracy			0.98	12630
macro avg	0.97	0.97	0.96	12630
weighted avg	0.98	0.98	0.98	12630

Part 2 프로젝트 설명

예측 및 정답 비교

```
plt.figure(figsize = (25, 25))

start_index = 0
for i in range(25):
    plt.subplot(5, 5, i + 1)
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])
    prediction = pred[start_index + i]
    actual = labels[start_index + i]
    col = 'g'
    if prediction != actual:
        col = 'r'
    plt.xlabel('Actual={} || Pred={}'.format(actual, prediction), color = col)
    plt.imshow(X_test[start_index + i])
plt.show()
```

Part 3,

시연 영상

Part 3

시연영상

https://www.youtube.com/watch?v=bCBDZMAjqwA

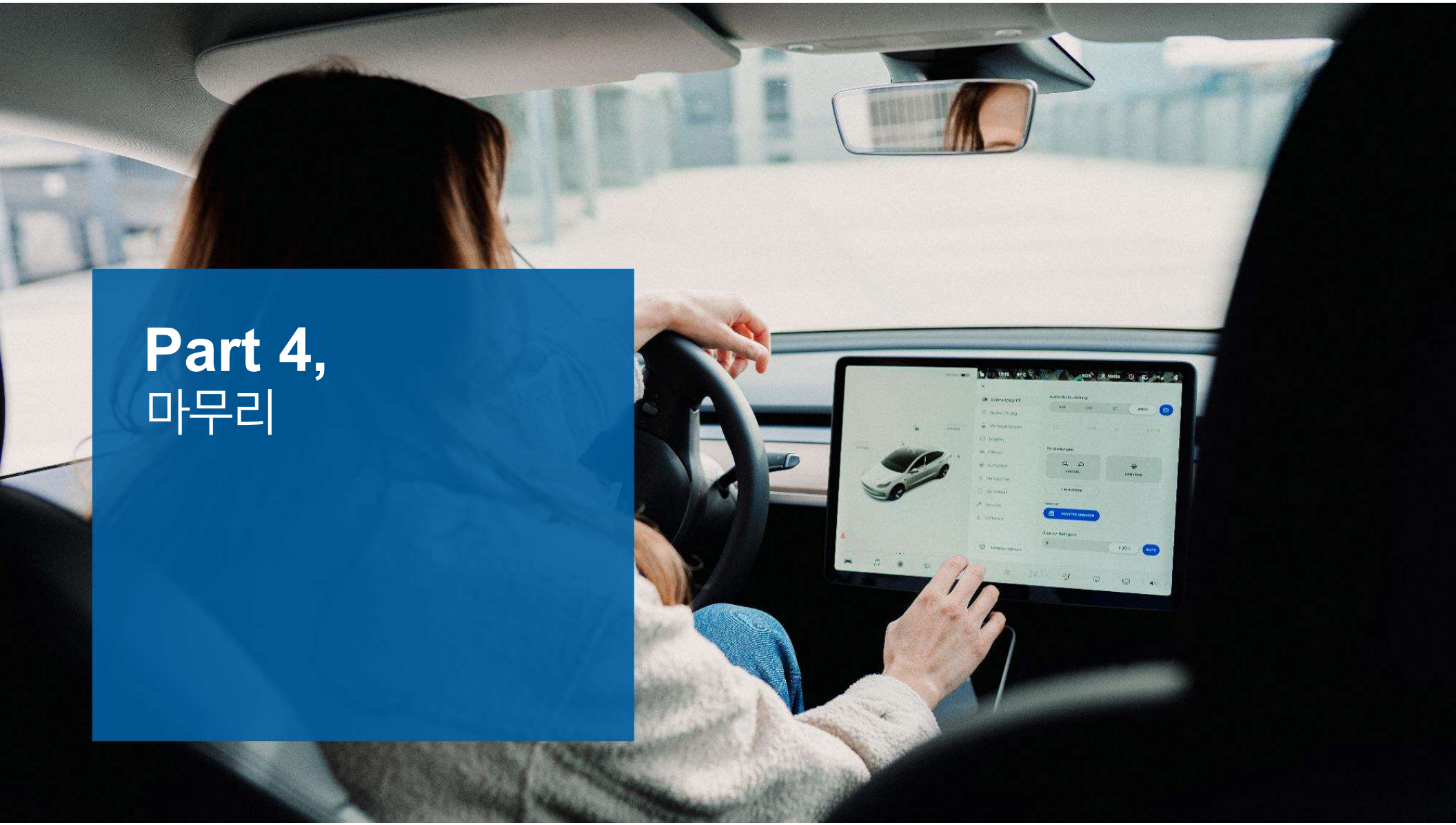
[Project] 교통 표지판 이미지 분류 ¶

프로젝트 목표

- 교통 표지판 이미지 데이터를 분석하고 딥러딩 모델을 통하여 표지판 종류를 예측하는 분류 모델 수행
- 대량의 이미지 데이터를 전 처리하는 과정과 이에 따른 CNN 모델의 성능 변화를 학습

프로젝트 목차

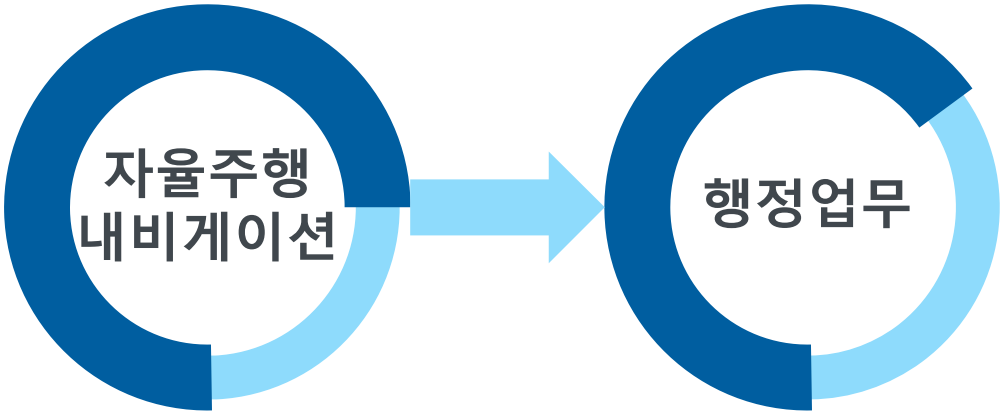
1. **데이터 분석:** 이미지 데이터를 이루고 있는 요소에 대해서 Dataframe를 사용하여 분석 및 확인
   1-1. 이미지 데이터 정보 파악하기 - Meta
   1-2. 이미지 데이터 정보 파악하기 - Train
   1-3. 이미지 데이터 정보 파악하기 - Test
2. **데이터 전 처리:** 이미지 데이터를 읽어오고 딥러닝 모델의 입력으로 전 처리



Part 4,
마무리

Part 4

마무리(프로젝트 발전 방향)



자율주행 자동차에 표지판 인식을 통해 속도 조절 및 위험/주의 사항 인지 및 대처 준비 등 자율주행에 필요한 요소 및 내비게이션 업데이트에 활용

자율 주행 자동차에 부착된 카메라를 통해 손상된 표지판을 인식 판단하여 해당 업무를 하는 기관에 즉각 조치 할 수 있도록 연결

각 차량이 과속 단속 카메라가 되어 주행하고 있는 도로의 제한속도 보다 높은 차량을 감지하고 번호판 인식 프로그램(딥러닝)과 병합하여 해당 차량번호를 해당 업무 기관으로 전송

Part 4

마무리(소감)

김규영

이번 프로젝트와 딥러닝을 연계하여 더 진행 한다면 사회에 기여가능한 프로젝트가 될 수 도 있겠다고 생각하여 딥러닝과 지능형프로젝트를 수업시간 이외에도 더욱 발전 시켜야겠다고 생각했습니다. 이런 프로젝트를 진행하기까지 한 학기동안 기초부터 다양한 활용까지 수업을 통해 가르쳐 주신 교수님께 너무 감사드리고 앞으로 남은 학기동안 다시 교수님께 배울 수 있는 과목들을 기대하며 지능형 시스템 과목을 끝으로 한학기 잘 마무리 하겠습니다. 감사합니다.

장정우

딥러닝에 이어 지능형시스템까지 교통 관련 프로젝트를 진행 해 보았습니다. 두 프로젝트 전부 이미지 분류 성향이 강해서 cnn모델을 사용하다보니 더욱 이해도가 깊어졌고, 데이터를 수집하는 과정에서도 많이 막히곤 했습니다. 또한 정확도를 위해 팀원과 함께 머리를 싸매고 진행하면서도 부족함이 많았습니다. 최선을 다했고, 99퍼정도되는 정확도를 얻을 수 있었습니다. 앞으로도 최선을 다 하여 이 프로젝트를 토대로 공모전까지 진행해 보고싶습니다. 오늘도 배울 수 있는 기회를 주셔서 감사합니다. 앞으로도 많이 배우겠습니다. 고생 많으셨습니다 교수님.

“대한민국 모든 도로가 안전하길 바랍니다