



Deep Learning

Traffic Sign Classification

U Kang
Seoul National University



In This Lecture

- German Traffic Sign Benchmarks
- Multi-class classification from traffic sign images
- Implement a deep learning model using Tensorflow2 (Keras)



Outline

- ➡ ☐ Introduction
- ☐ Dataset
- ☐ Preprocessing Codes



About the Task

- The German Traffic Sign Benchmark is a multi-class classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011
 - ❑ Link: <https://benchmark.ini.rub.de/index.html>
 - ❑ Number of classes: 43
 - ❑ Imbalanced classification



Problem Definition

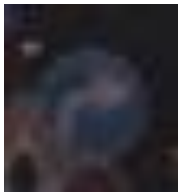
- **Given:** German Traffic Sign data
- **Classify:** test data into the correct categories
 - E.g.



class 0
"Speed limit (20km/h)"



class 4
"Speed limit (70km/h)"



class 33
"Turn right ahead"



class 37
"Go straight or left"



Outline

☒ Introduction

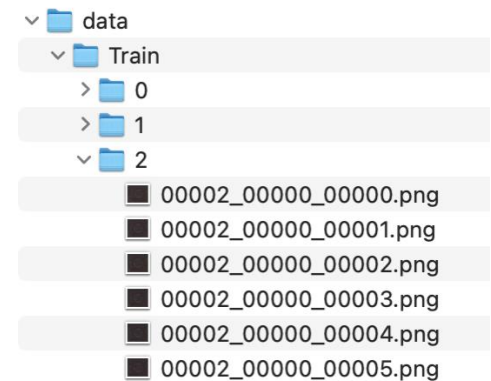
 ☐ **Dataset**

☐ Preprocessing Codes



Training Dataset

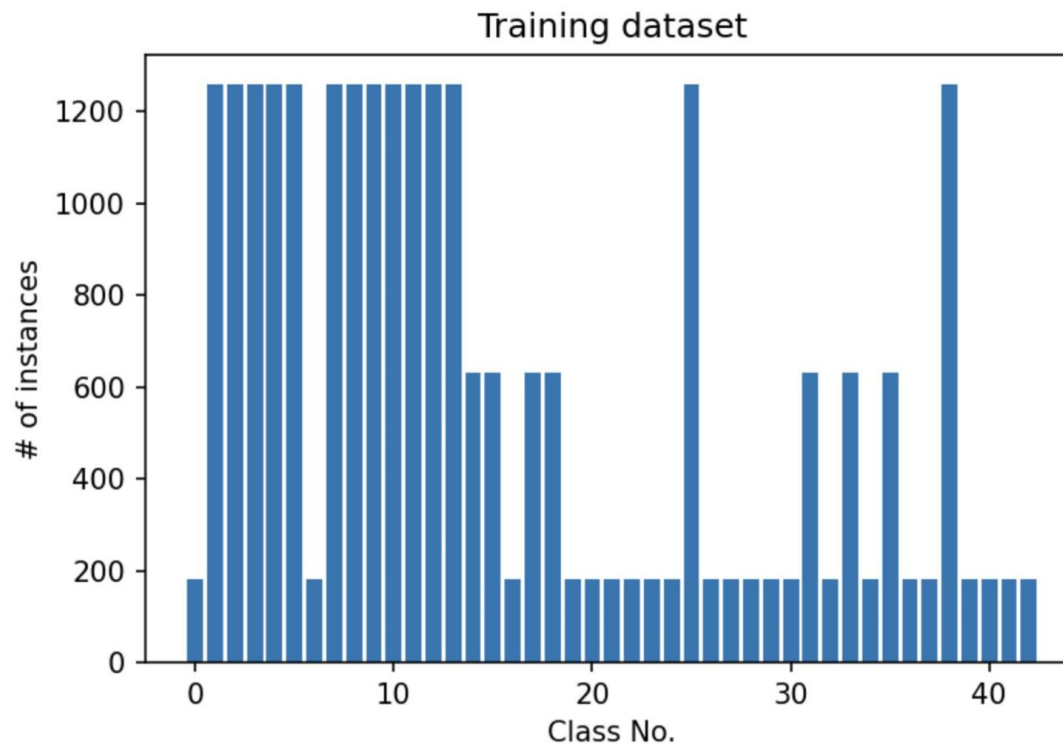
- Approximately 30,000 images
- Image sizes vary between 15x15 to 250x250 pixels
 - Images are not necessarily squared
- The training set archive is structures as follows:
 - One directory per class
 - Training images are grouped by tracks
 - Each track contains 30 images of one single physical traffic sign
 - Filename format
 - {class id}_{track id}_{i-th image for the track}.png





Training Dataset

- Label distribution of training data





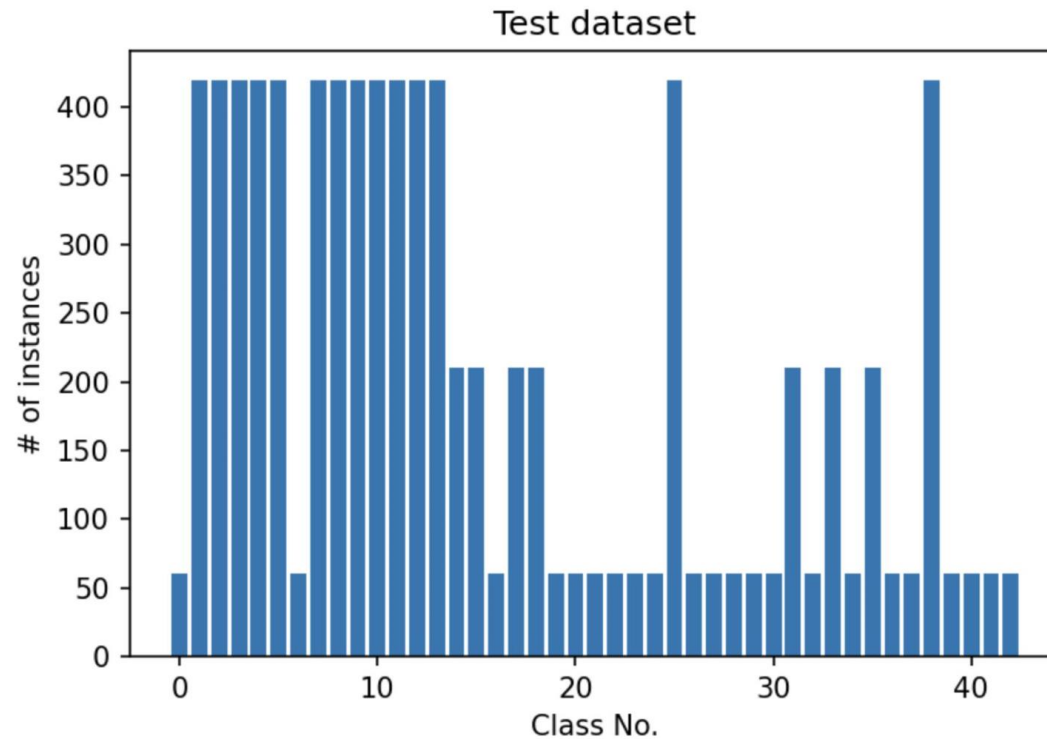
Test Dataset

- Approximately 10,000 images
- Image sizes vary between 15x15 to 250x250 pixels
 - Images are not necessarily squared



Test Dataset

- Label distribution of test data





Outline

☒ Introduction

☒ Data

 ☐ **Preprocessing Codes**



Import libraries

- Import the libraries such as numpy, cv2, os, pillow, tensorflow, and keras

```
In [1]: import os
import numpy as np
import cv2
from PIL import Image

import tensorflow as tf
from tensorflow.keras.utils import to_categorical

tf.__version__
```



Loading the Dataset

- Read input training data and put images and labels into numpy arrays

```
In [2]: data = []
labels = []

height = 30
width = 30
channels = 3
num_classes = 43

for i in range(num_classes) :
    path = f'data/Train/{i}/'
    Class = os.listdir(path)
    for a in Class:
        image = cv2.imread(path + a)
        image_from_array = Image.fromarray(image, 'RGB')
        size_image = image_from_array.resize((height, width))
        data.append(np.array(size_image))
        labels.append(i)

data = np.array(data)
labels = np.array(labels)
```



Test

- Test measure

- $F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$

- Save results to csv file

- Save path: './result.csv'
 - 1st column: test file name
 - 2nd column: predicted classe
 - Separator: comma

jupyter result.csv

	File	Edit	View	Language
1	00000.png,12			
2	00001.png,11			
3	00002.png,33			
4	00003.png,33			
5	00004.png,11			
6	00005.png,3			
7	00006.png,8			
8	00007.png,42			
9	00008.png,25			
10	00009.png,35			



Submission Guide

- Submit the “team#.zip” that contains followings:
 - “team#.csv”: result including prediction labels
 - Refer to the page 14
 - “team#.ppt”: ppt summarizes your work
 - 8 min for presentation 2 min for Q&A
 - Preprocessing, model architecture, training technique etc.
 - “team#.ipynb”: your code
- E-mail address: bandalg1997@gmail.com
 - Mail title: DS2_proj_{team#}_{submit#}
- Leaderboard [[link](#)]
- Submission Due: 2022/ 06/ 07 00:00



Questions?