

Міністерство освіти та науки України  
Львівський національний університет імені Івана Франка

# Звіт

Про виконання лабораторної роботи №4  
Класифікація дорожніх знаків за допомогою TensorFlow/Keras/Deep  
learning

Виконав:  
студент групи Фес-31  
Мацевко Олександр  
Перевірив:  
Сінкевич О.О.

Львів 2019

**Мета :** “Встановити та запустити нейромережу з класифікації дорожніх знаків, за допомогою TensorFlow, Keras та Deep Learning”

## Теоретичні відомості :

### Класифікація дорожніх знаків з керасом та глибоким навчанням

Ми впровадимо `TrafficSignNet` - звивисту нейронну мережу, яку ми будемо тренувати за нашим набором даних.

Враховуючи нашу підготовлену модель, ми оцінимо її точність на тестових даних і навіть навчимося робити прогнози щодо нових вхідних даних.

Класифікація дорожніх знаків - це процес автоматичного розпізнавання дорожніх знаків на дорозі, включаючи знаки обмеження швидкості, знаки обмежень, знаки поворотів тощо.

У режимі реального часу розпізнавання дорожніх знаків є двоступеневим процесом:

1. **Локалізація:** визначте та локалізуйте, на вхідному зображенні / кадрі потрібно визначити де знаходиться дорожній знак.
2. **Розпізнавання:** візьміть локалізовану картинку та фактично *розпізнайте* та *класифікуйте* дорожній знак.

У наборі даних GTSRB є ряд проблем, **перше з яких - зображення низької роздільної здатності** , а ще гірше, **мають слабкий контраст** (як показано на **малюнку 2** вище). Ці зображення пікселізовані, і в деяких випадках надзвичайно складно, якщо не неможливо, людському оку та мозку розпізнати знак.

**Для успішного навчання точного класифікатора дорожніх знаків нам необхідно розробити експеримент, який може:**

- Попередньо обробити наші вхідні зображення, щоб поліпшити контраст.
- Акаунт для визначення рівня класу.

Я витягнув набір даних у свій каталог проектів, як ви можете бачити тут:

```
Traffic Sign Classification with Keras and Deep Lea <> Shell
1 $ tree --dirsfirst --filelimit 10
2 .
3 |
4 |--- examples [25 entries]
5 |--- gtsrb-german-traffic-sign
6 |   |--- Meta [43 entries]
7 |   |--- Test [12631 entries]
8 |   |--- Train [43 entries]
9 |   |--- meta-1 [43 entries]
10 |   |--- test-1 [12631 entries]
11 |   |--- train-1 [43 entries]
12 |   |--- Meta.csv
13 |   |--- Test.csv
14 |   |--- Train.csv
15 |--- output
16 |   |--- trafficsignnet.model
17 |   |   |--- assets
18 |   |   |--- variables
19 |   |   |   |--- variables.data-00000-of-00002
20 |   |   |   |--- variables.data-00001-of-00002
21 |   |   |   |--- variables.index
22 |   |   |--- saved_model.pb
23 |   |--- plot.png
24 |--- pyimagesearch
25 |   |--- __init__.py
26 |   |--- trafficsignnet.py
27 |--- train.py
28 |--- signnames.csv
29 |--- predict.py
30 13 directories, 13 files
```

Наш проект містить три основні каталоги та один модуль Python:

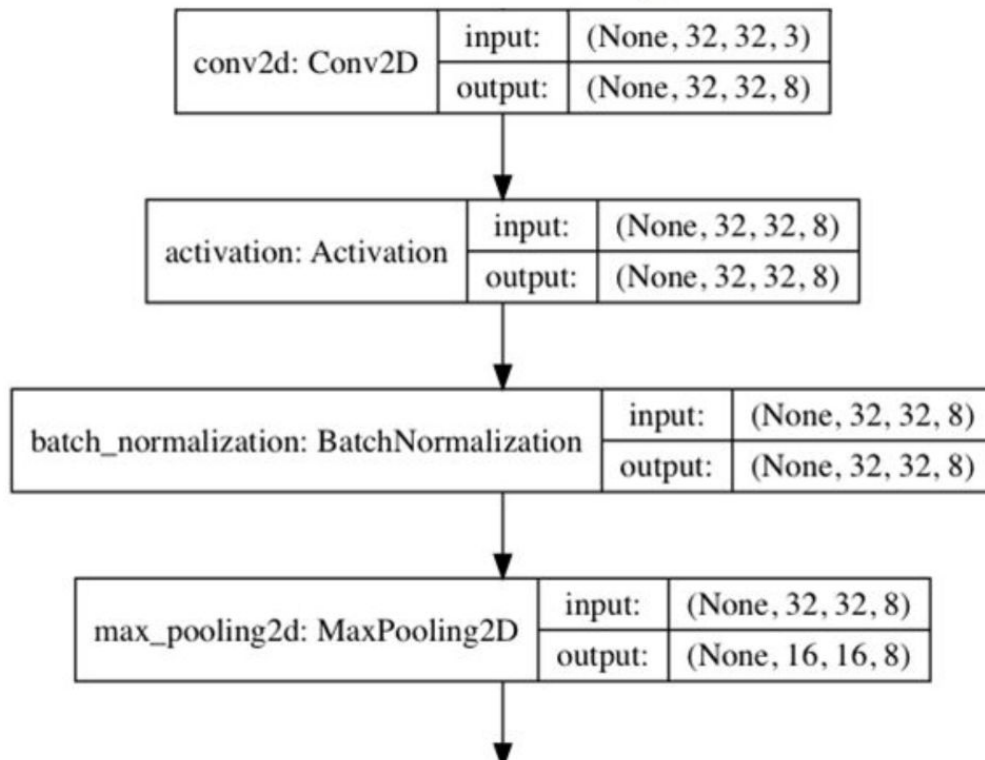
- gtsrb-german-traffic-sign/ : Наш GTSRB набір даних .
- output/ : Містить нашу вихідну модель та графік історії тренувань, створений train.py .
- examples/ : Містить випадкову вибірку з 25 анотованих зображень що генеруються predict.py .
- pyimagesearch : Модуль що включає наш TrafficSignNet CNN.

Ми також будемо проходити train.py та predict.py . Наш навчальний скрипт завантажує дані, збирає модель, тренує та виводить на диск серіалізовану модель та графічне зображення. Звідти наш сценарій прогнозування генерує помічені зображення для цілей візуальної перевірки.

Для цієї лабораторної роботи нам потрібно встановити такі пакети:

- OpenCV
- NumPy
- scikit-learn
- scikit-image
- imutils
- matplotlib
- TensorFlow 2.0 (CPU or GPU)

# TrafficSignNet



## Хід роботи:

1. Встановлюємо необхідні нам пакети:

```
Regression with Keras
$ workon traffic_signs
$ pip install opencv-contrib-python
$ pip install numpy
$ pip install scikit-learn
$ pip install scikit-image
$ pip install imutils
$ pip install matplotlib
$ pip install tensorflow==2.0.0 # or tensorflow-gpu
```

2. Після того як всі наші файли на місці та всі пакети встановлені розпочинаємо навчання за допомогою команди:

```
ura@ura-HP-Pavilion-15-Notebook-PC: ~/Downloads/traffic-sign-recognition
ura@ura-HP-Pavilion-15-Notebook-PC:~/Downloads/traffic-sign-recognition$ python3
train.py --dataset gtsrb-german-traffic-sign --model output/trafficsignnet.model
--plot output/plot.png
```

3. Очікуємо повідомлення про закінчення навчання :

```

611/612 [=====>.] - ETA: 0s - los612/612 [=====]
[INFO] evaluating network...
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/classification.py:1437: UndefinedMetric
'precision', 'predicted', average, warn_for)

```

	precision	recall	f1-score	support
Speed limit (20km/h)	0.00	0.00	0.00	60
Speed limit (30km/h)	0.00	0.00	0.00	720
Speed limit (50km/h)	0.00	0.00	0.00	750
Speed limit (60km/h)	0.00	0.00	0.00	450
Speed limit (70km/h)	0.00	0.00	0.00	660
Speed limit (80km/h)	0.00	0.00	0.00	630
End of speed limit (80km/h)	0.00	0.00	0.00	150
Speed limit (100km/h)	0.00	0.00	0.00	450
Speed limit (120km/h)	0.00	0.00	0.00	450
No passing	1.00	0.00	0.00	480
No passing for vehicles over 3.5 metric tons	0.00	0.00	0.00	660
Right-of-way at the next intersection	1.00	0.00	0.00	420
Priority road	0.00	0.00	0.00	690
Yield	1.00	0.00	0.00	720
Stop	0.22	0.04	0.07	270
No vehicles	0.00	0.00	0.00	210
Vehicles over 3.5 metric tons prohibited	0.00	0.00	0.00	150
No entry	1.00	0.38	0.55	360
General caution	0.00	0.00	0.00	390
Dangerous curve to the left	0.00	0.00	0.00	60
Dangerous curve to the right	0.01	0.28	0.01	90
Double curve	0.00	0.00	0.00	90
Bumpy road	0.00	0.00	0.00	120
Slippery road	0.00	0.00	0.00	150
Road narrows on the right	0.04	0.03	0.04	90
Road work	0.64	0.34	0.44	480
Traffic signals	0.00	0.00	0.00	180
Pedestrians	0.11	0.25	0.15	60
Children crossing	0.05	0.65	0.10	150
Bicycles crossing	0.00	0.00	0.00	90
Beware of ice/snow	0.00	0.00	0.00	150
Wild animals crossing	0.00	0.00	0.00	270
End of all speed and passing limits	0.00	0.00	0.00	60
Turn right ahead	0.02	0.32	0.03	210
Turn left ahead	0.00	0.00	0.00	120
Ahead only	0.00	0.00	0.00	390
Go straight or right	0.00	0.00	0.00	120
Go straight or left	0.00	0.00	0.00	60
Keep right	0.00	0.00	0.00	690
Keep left	0.03	1.00	0.06	90
Roundabout mandatory	0.00	0.00	0.00	90
End of no passing	0.00	0.00	0.00	60
End of no passing by vehicles over 3.5 metric tons	0.00	0.00	0.00	90
accuracy			0.05	12630
macro avg	0.12	0.08	0.03	12630
weighted avg	0.19	0.05	0.04	12630

```

[INFO] serializing network to 'output/trafficsignnet.model'...
Traceback (most recent call last):
  File "train.py", line 144, in <module>
    model.save(args["model"])

```

## Висновок :

На цій лабораторній роботі я зрозумів принцип роботи згорткової нейромережі та власноруч запустив нейромережу з класифікації дорожніх знаків.