# Projekt „Rozpoznawanie znaków drogowych”

W oparciu o data set zawierający 43 klasy ze znakami drogowymi, uczymy model CNN a następnie wynik uczenia zapisany zostaje w pliku \*.hdf5. Jest to niezbędne by można było uczyć nasz model np. w Google Colab w środowisku GPU, następnie zapisany model wczytujemy w dowolnym środowisku lokalnym do sprawdzenia jak rozpoznaje pokazywane znaki.

Procedura ta przyspiesza proces uczenia z kilku godzin w środowisku lokalnym na CPU do ok. 25 min. W Google Colab na GPU

#Projekt składa się z:

1. Katalog myData -> zawiera nasz zbiór danych podzielony na 43 klasy w wersji startowej mamy plik myData.zip który należy rozpakować w katalogu z projektem (to niestety zabiera sporo czasu zwłaszcza jeśli wgrywamy to na gogle drive.

2. labels.csv -> plik zawiera opisy klas.

3. my\_model\_30\_epok\_nzb\_25\_06\_22.hdf5 -> zapis modelu w wersji z danymi nie zbalansowanymi

4. my\_model\_30\_epok\_zb\_25\_06\_22.hdf5 -> zapis modelu w wersji z danymi po przeprowadzeniu operacji balansowania danych wejściowych

5. oraz opisane poniżej pliki.

# Projekt podzielony został na 3 pliki utworzone w Python

1. Znaki\_drogowe\_kod\_trenujacy\_model.py -> w tym pliku zaszyte są funkcje, które mają za zadanie wczytać nasz zbiór danych i na ich podstawie wytrenować model.

2. Rozpozanwanie\_znakow.py -> w tym pliku wczytywany jest nasz wyuczony model oraz odpalana jest kamerka w oddzielnym okienku. Gdy pokażemy znak program rozpozna do jakiej klasy należy doda opis oraz poda prawdopodobieństwo z jakim rozpoznał znak.

# W projekcie jest też plik znaki\_plik\_uczacy\_26\_06\_22.ipynb plik ten jest przygotowany w taki sposób by można było go uruchamiać w Gogle Colab lub w jupyterlab poprzez wybór odpowiednich komórek.

# Projekt wykonany w oparciu o:

1. Python version: 3.8.8

2. Numpy version: 1.22.4

3. Patplotlib version: 3.5.2

4. Opencv version: 4.6.0

5. Pandas version: 1.4.3

6. Tensorflow version: 2.9.0-dev20220316

# Instalacje niezbędne do działania:

1. Python:

https://www.python.org/

PyCharm:

https://www.jetbrains.com/pycharm/

2. Wgranie najnowszej wersji pip

• pip install --upgrade pip

3. Zestaw niezbędnych bibliotek

pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose scikit-learn opencv-python tensorflow scikit-learn opencv-python tensorflow

# Po dokonaniu instalacji należy pamiętać by zmienić ścieżki dostępowe do:

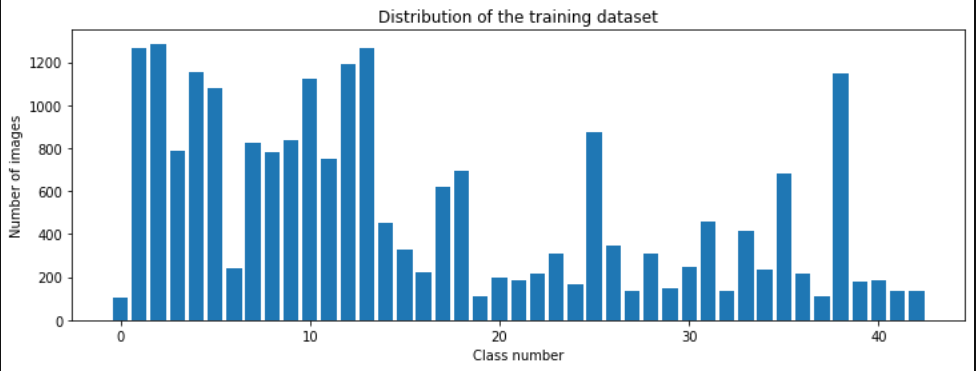
1. Katalogu myData

2. Do pliku labels.csv

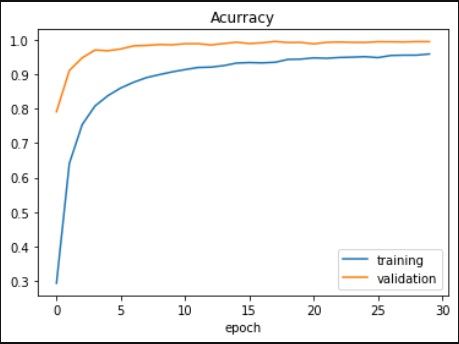
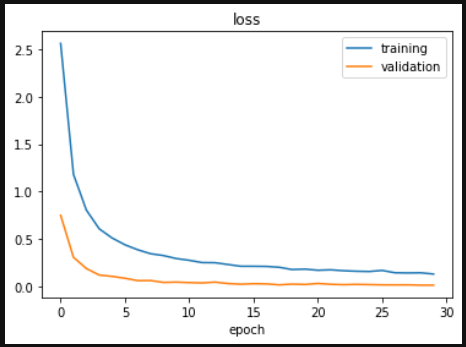
3. Do pliku, w którym zapiszemy nasz model

4. Do pliku z zapisanym modelem

Zbiór niezbalansowany:



GRAFICZNA REPREZENTACJA WYNIKÓW NAUKI NASZEGO MODELU



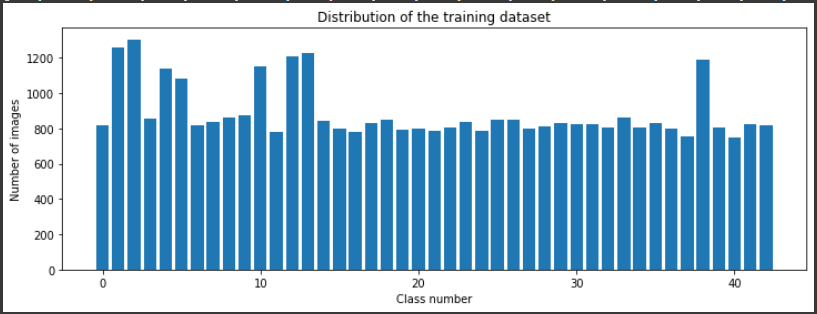
Test Score: 0.02027769200503826

Test Accuracy: 0.9946839213371277

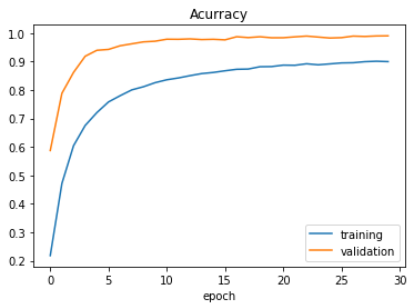
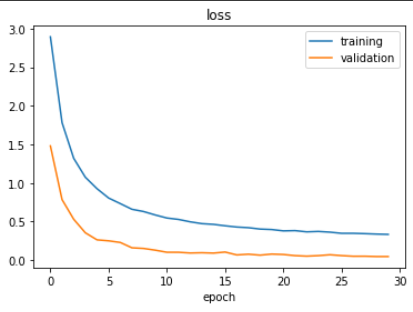
A tak wygląda test rozpoznawania znaków.



Zbiór po zbalansowaniu:



GRAFICZNA REPREZENTACJA WYNIKÓW NAUKI NASZEGO MODELU



Test Score: 0.044132836163043976

Test Accuracy: 0.9909342527389526

# Wnioski

Mimo zbalansowania danych wejściowych model przestał sobie radzić z rozpoznawaniem znaków. Nie rozpoznał żadnego ze znaków.