



WYKRYWANIE POLSKICH ZNAKÓW DROGOWYCH

Karol Weyna, Przemysław Sulecki - UTP Bydgoszcz 2018/2019



Opis projektu	2
Założenia:.....	2
Wymagania:.....	2
Instalacja.....	2
Instalacja CUDA ToolKit:.....	2
Ustawienie zmiennych środowiskowych.....	2
Instalacja cuDNN 7.6.....	3
Aktualizacja sterowników karty graficznej (krok dodatkowy).....	3
Instalacja Anacondy (Python 3.6).....	3
Instalacja TensorFlow-GPU.....	3
Sprawdzenie naszej konfiguracji	4
Tworzenie struktury katalogów:	4
Importowanie modelu:.....	4
Pobranie niezbędnych plików.....	4
Instalacja niezbędnych modułów	4
Konfiguracja PYTHONPATH	4
Kompilacja plików proto	4
Tworzenie zestawu danych.....	5
Skrypty automatyzujące	5
xml_to_csv.py.....	5
generate_tfrecord.py.....	5
pathChecker.py.....	6
checkClasses.py	6
Konfiguracja.....	6
Trening modelu	6
Eksportowanie wykresu wnioskowania.....	7
Uwzględnione znaki drogowe	7
Znaki ostrzegawcze	7
Znaki zakazu	9
Znaki nakazu.....	10
Znaki informacyjne.....	10
Przykład działania	12
Na bazie zdjęć.....	12
Na bazie filmów	13

OPIS PROJEKTU

Projekt zakłada stworzenie sztucznej sieci neuronowej w celu detekcji polskich znaków drogowych, wykorzystując jako dane wejściowe plik wideo, zdjęcie lub obraz w czasie rzeczywistym przy wykorzystaniu framework'a TensorFlow.

ZAŁOŻENIA:

- obliczenia są przeprowadzane na GPU (rdzenie CUDA, konieczna karta graficzna NVIDIA),
- algorytm powinien wykrywać 84 polskie znaki drogowe,
- próg wykrycia obiektu jest ustawiony na minimum 80%,
- wykryty obiekt jest umieszczany w prostokącie wraz z procentowym przewidywaniem wykrycia,
- na jednej klatce może być wykryte wiele obiektów,
- każdy obiekt jest opisany jako kod znaku obowiązujący w Polsce (np. D1 - droga z pierwszeństwem).

WYMAGANIA:

- GTX 650 lub nowszy,
- Anaconda z python 3.6,
- CUDA Tool kit v 10.0,
- cuDNN 7.6.

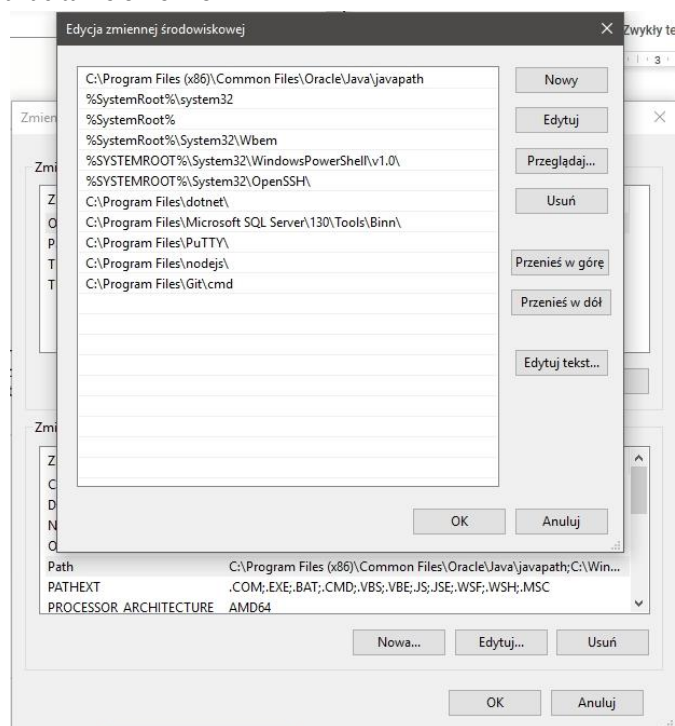
INSTALACJA

INSTALACJA CUDA TOOLKIT:

- Przechodzimy na stronę <https://developer.nvidia.com/cuda-10.0-download-archive>.
- Wybieramy nasz system operacyjny, w naszym przypadku jest to Windows.
- Po wybraniu systemu operacyjnego należy wybrać architekturę, w naszym przypadku jest to x86_64.
- Po wybraniu architektury systemu należy wybrać wersję systemu, w naszym przypadku jest to Windows 10.
- Następnie wybieramy typ instalatora.
- Po wybraniu Typu instalatora klikamy zielony przycisk **Download**.
- Po pobraniu pliku, klikamy na niego 2 razy i dokonujemy jego instalacji w naszym systemie.

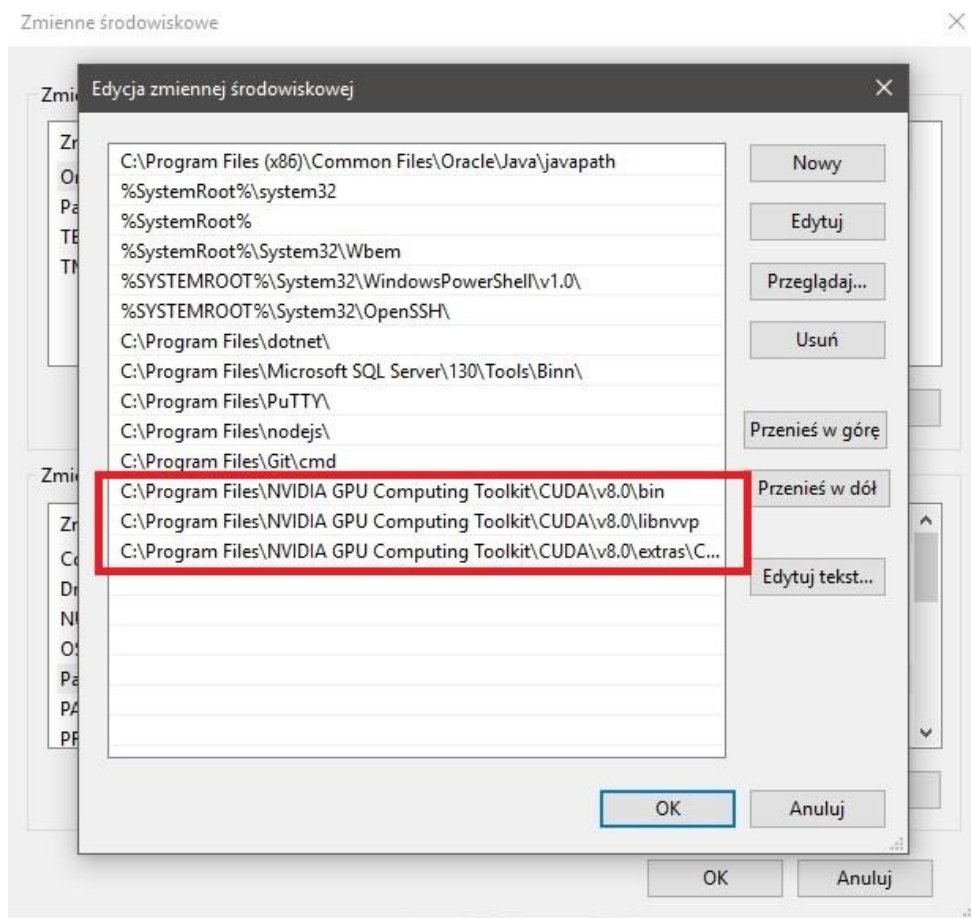
USTAWIENIE ZMIENNYCH ŚRODOWISKOWYCH

- Klikamy klawisz Windows na klawiaturze i w pole szukaj wpisujemy "Edytuj zmienne środowiskowe systemu".
- Następnie klikamy "Zmienne środowiskowe...", wybieramy z zmiennych systemowych zmienną Path i klikamy edytuj.
- Naszemu oczom powinno się ukazać takie okienko.



- Następnie klikamy "Nowy" i dodajemy po kolei trzy ścieżki:

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\bin
 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\libnvvp
 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\extras\CUPTI\libx64
 -Ostatecznie nasza zmienna systemowa Path powinna mieć takie wpisy:



INSTALACJA CUDNN 7.6

- Przechodzimy pod adres <https://developer.nvidia.com/rdp/cudnn-download>, (konieczne jest posiadanie konta NVIDIA).
- Klikamy "I Agree..." i wybieramy "Download cuDNN v7.6.0 (May 20, 2019), for CUDA 10.0".
- Następnie wybieramy nasz system operacyjny, w naszym wypadku był to Windows 10.
- Po pobraniu pliku rozpakowujemy go, a folder "cuda" przenosimy bezpośrednio na dysk C:
- Dodajemy do zmiennej systemowej Path kolejną ścieżkę:

C:\cuda\bin

```
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\bin
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\libnvvp
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\extras\C...
C:\cuda\bin
```

AKTUALIZACJA STEROWNIKÓW KARTY GRAFICZNEJ (KROK DODATKOWY)

- Przechodzimy pod adres <https://www.nvidia.pl/Download/index.aspx?lang=pl>
- Po przejściu pod wskazany adres wybieramy naszą kartę graficzną oraz system operacyjny, w naszym przypadku był to Geforce 840M z 364 rdzeniami CUDA oraz system Windows 10, następnie klikamy "szukaj", po przeładowaniu strony klikamy "pobierz".
- Po pobraniu sterowników dokonujemy ich instalacji, następnie wykonujemy restart komputera.

INSTALACJA ANACONDY (PYTHON 3.6)

- Przechodzimy pod adres <https://www.anaconda.com/distribution/#download-section>
- Wybieramy opcję z Python 3.6.
- Instalujemy pobrany plik.

INSTALACJA TENSORFLOW-GPU

- Uruchamiamy wiersz poleceń - cmd.exe jako administrator.

-Wpisujemy **"conda create -n tensorflow pip python=3.6"**.

-Następnie **"activate tensorflow"**.

-Jeżeli obok ścieżki w wierszu poleceń w nawiasie jest wyświetlone (tensorflow) to oznacza, że wszystko do tego punktu wykonaliśmy poprawnie.

-Wpisujemy w wierszu poleceń: **"pip install --ignore-installed --upgrade tensorflow-gpu"**.

SPRAWDZENIE NASZEJ KONFIGURACJI

-Tworzymy plik test.py z zawartością:

```
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

-Następnie go uruchamiamy wpisując w konsoli **"python test.py"**.

-W konsoli powinien się wyświetlić napis **"Hello, TensorFlow!"** (może to chwilę potrwać).

TWORZENIE STRUKTURY KATALOGÓW:

-Pobieramy pliki z repozytorium TensorFlow na githubie <https://github.com/tensorflow/models>

-Tworzymy na dysku C: katalog tensorflow i do niego rozpakowujemy pobrany plik.

-Po wejściu do katalogu tensorflow na dysku C: możemy zmienić nazwę z models-master na po prostu **"models"**.

TensorFlow zapewnia kilka modeli wykrywania obiektów (wstępnie wyszkolonych klasyfikatorów o specyficznych architekturach sieci neuronowych). Niektóre modele (takie jak model SSD-MobileNet) mają architekturę, która pozwala na szybsze wykrywanie, ale z mniejszą dokładnością, podczas gdy niektóre modele (takie jak model Faster-RCNN) dają wolniejsze wykrywanie, ale z większą dokładnością. W naszym przypadku wybraliśmy model Faster-RCNN-Inception-V2-COCO

IMPORTOWANIE MODELU:

-Przechodzimy pod adres

http://download.tensorflow.org/models/object_detection/faster_rcnn_inception_v2_coco_2018_01_28.tar.gz i pobieramy model

-Następnie rozpakowujemy model do katalogu C:\tensorflow\models\research\object_detection

POBRANIE NIEZBĘDNYCH PLIKÓW

Pobranie niezbędnych plików do min. generowania TFRecords, plików CSV z XML, a także zliczania liczby klas w wszystkich plikach XML

-Przechodzimy pod adres https://github.com/przsul/Traffic-Sign-Recognition/tree/prepared_files i pobieramy pliki z repozytorium

-Następnie wypakowujemy archiwum na C:\tensorflow\models\research\object_detection

INSTALACJA NIEZBĘDNYCH MODUŁÓW

-Ponownie otwieramy cmd.exe jako administrator i wpisujemy **"active tensorflow"**

-Przechodzimy w konsoli do katalogu C:\tensorflow\models\research\object_detection

-Wpisujemy w konsoli **"conda install -c anaconda protobuf"**

-Następnie wpisujemy w konsoli takie polecenia:

```
"pip install pillow",
"pip install lxml",
"pip install Cython",
"pip install jupyter",
"pip install matplotlib",
"pip install pandas",
"pip install opencv-python"
```

KONFIGURACJA PYTHONPATH

-Otwieramy konsolę cmd.exe jako administrator

-Wpisujemy set

PYTHONPATH=C:\tensorflow\models;C:\tensorflow\models\research;C:\tensorflow\models\research\slim

KOMPILACJA PLIKÓW PROTO

-Otwieramy konsolę cmd.exe jako administrator

-Przechodzimy do folderu C:\models\research

-Wpisujemy **for /f %f in ('dir /b object_detection\protos*.proto') do protoc object_detection\protos\%f --python_out=.**

-Następnie wpisujemy **python setup.py build** oraz **python setup.py install**

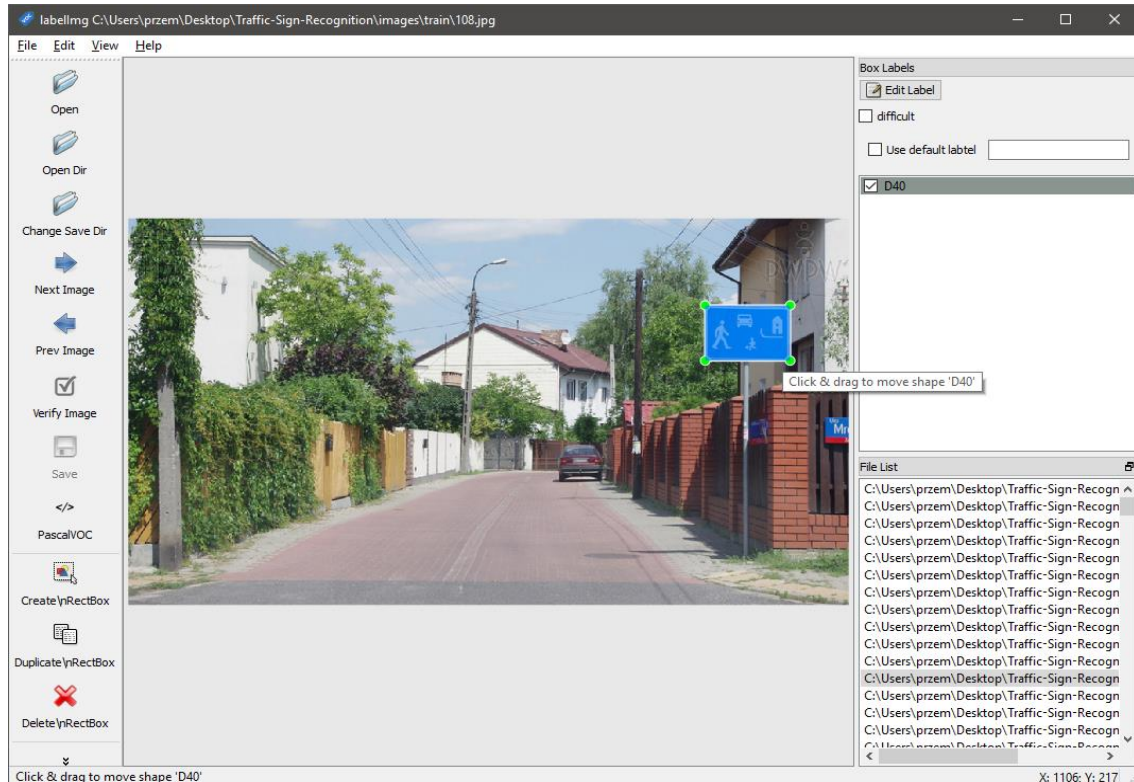
TWORZENIE ZESTAWU DANYCH

Zestaw danych zawiera 80% danych treningowych (*/images/train/*) oraz 20% danych testowych (*/images/test/*). W celu stworzenia zestawu danych, zostało wykorzystanych łącznie 967 zdjęć przedstawiających znaki drogowe. Głównym zadaniem było wygenerowanie plików XML zawierających dane m. in:

- współrzędnych obszaru przedstawiających znak do wykrycia,
- etykietę obszaru,
- rozdzielczość zdjęcia wejściowego,
- ścieżkę do zdjęcia wejściowego.

Źródłem danych były materiały prywatne oraz ogólnodostępna baza pytań egzaminacyjnych na prawo jazdy.

Narzędzie wykorzystane do generowania plików XML to *labellmg*:



SKRYPTY AUTOMATYZUJĄCE

Wykorzystane skrypty w celu utworzenia zestawu danych potrzebnych do nauczania modelu.

XML_TO_CSV.PY

Plik ten służy do wygenerowania dwóch plików CSV (*test_labels.csv* oraz *train_labels.csv*) na podstawie plików .XML znajdujących się w folderach */images/train* i */images/test*.

Uruchomienie skryptu:

-Uruchomienie konsoli cmd.exe jako administrator i wpisanie "active tensorflow"

-Przejsięcie do ścieżki *C:/tensorflow/models/research/object_detection*

-Wpisanie **python xml_to_csv.py**

-Po wykonaniu skryptu w folderze images powinny utworzyć się dwa pliki z rozszerzeniem .csv

GENERATE_TFRECORD.PY

Plik ten służy do wygenerowania pliku z rozszerzeniem .tfrecord na podstawie wcześniej utworzonego pliku z rozszerzeniem .csv

Uruchomienie skryptu:

-Uruchomienie konsoli cmd.exe jako administrator i wpisanie "active tensorflow"

-Przejsięcie do ścieżki *C:/tensorflow/models/research/object_detection*

-Wpisanie **python generate_tfrecord.py --csv_input=images\train_labels.csv --image_dir=images\train --output_path=train.record** w konsoli

-Wpisanie **python generate_tfrecord.py --csv_input=images\test_labels.csv --image_dir=images\test --output_path=test.record** konsoli

-Po wykonanie z każdej tych komend powinien się utworzyć plik z rozszerzeniem .tfrecord w katalogu object_detection(Generowanie pliku .tfrecord może zająć chwile czasu)

PATHCHECKER.PY

Plik ten znajduje się w folderze **train** oraz **test** służy on do naprawienia ścieżek w plikach .XML w przypadku przenoszenia ich między różnymi komputerami

Uruchomienie skryptu:

-Uruchomienie konsoli cmd.exe jako administrator i wpisanie "active tensorflow"

-Przejdźcie do ścieżki C:/tensorflow/models/research/object_detection/images/train lub test

-Wpisanie **python pathChecker.py** w konsoli,

-Po wykonaniu skryptu w plikach .XML będą widnieć prawidłowe ścieżki

CHECKCLASSES.PY

Plik ten znajduje się w folderze train, a służy do wypisania i policzenia wszystkich klas znajdujących się w plikach .XML

Uruchomienie skryptu:

-Uruchomienie konsoli cmd.exe jako administrator i wpisanie "active tensorflow"

-Przejdźcie do ścieżki C:/tensorflow/models/research/object_detection/images/train

-Wpisanie **python checkClasses.py** w konsoli,

-Po wykonaniu skryptu w konsoli otrzymamy informacje o ilości klas, a także ich nazwach

KONFIGURACJA

-Na początku w katalogu C:/tensorflow/models/research/object_detection/training należy utworzyć plik labelmap.pbtxt, który będzie zawierał nazwy klas oraz ich id

Plik dostępny pod adresem: https://github.com/przsu1/Traffic-Sign-Recognition/blob/prepare_files/training/labelmap.pbtxt

-Z ścieżki C:/tensorflow/models/research/object_detection/samples/configs kopiujemy plik

faster_rcnn_inception_v2_pets.config do folderu C:/tensorflow/models/research/object_detection/training oraz go otwieramy

Plik dostępny pod adresem:

https://github.com/przsu1/Traffic-Sign-Recognition/blob/prepare_files/training/faster_rcnn_inception_v2_pets.config

-Następnie zmieniamy parametr **num_classes**, w naszym przypadku ustawiamy 84

-Należy także skonfigurować ścieżki do modeli, w naszym przypadku ścieżka do modelu wygląda w następujący sposób:

C:/tensorflow/models/research/object_detection/faster_rcnn_inception_v2_coco_2018_01_28/model.ckpt

-Kolejną rzeczą jaką należy wykonać jest ustawienie ścieżek do plików .tfrecord, w naszym przypadku wyglądają one w następujący sposób:

C:/tensorflow/models/research/object_detection/train.record

C:/tensorflow/models/research/object_detection/test.record

-Ostatnią rzeczą jest ustawienie ścieżki do pliku .pbtxt zawierającego wszystkie nasze klasy wraz z id , w naszym przypadku wygląda ona w następujący sposób:

C:/tensorflow/models/research/object_detection/training/labelmap.pbtxt

TRENING MODELU

Po skonfigurowanie wszystkich wymaganych ścieżek w pliku .config należy uruchomić trenowanie modelu.

-Uruchomić konsolę cmd.exe jako administrator, musimy znajdować się w katalogu

C:/tensorflow/models/research/object_detection

-Wpisać "active tensorflow"

-Uruchomić skrypt poprzez wpisanie **python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.config**

tor: Wiersz polecenia - python train.py --logtostderr --train_dir=training/ --pipeline_config_path=

```
low:global step 120559: loss = 0.0039 (1.300 sec/step)
low:global step 120560: loss = 0.0152 (1.306 sec/step)
low:global step 120560: loss = 0.0152 (1.306 sec/step)
low:global step 120561: loss = 0.0120 (1.304 sec/step)
low:global step 120561: loss = 0.0120 (1.304 sec/step)
low:global step 120562: loss = 0.0260 (1.313 sec/step)
low:global step 120562: loss = 0.0260 (1.313 sec/step)
low:global step 120563: loss = 0.0241 (1.312 sec/step)
low:global step 120563: loss = 0.0241 (1.312 sec/step)
```

Szkolenie algorytmu w naszym przykładzie zajęło 50 godzin. Czas tego szkolenia jest zależny od szybkości naszego GPU, wielkości VRAM, ilości rdzeni CUDA, wielkości dataset'u, a także użytego modelu.

EKSPORTOWANIE WYKRESU WNIOSKOWANIA

-Uruchomić konsolę cmd.exe jako administrator

-Wpisać "active tensorflow"

-Wpisać polecenie **python export_inference_graph.py --input_type image_tensor --pipeline_config_path training/faster_rcnn_inception_v2_pets.config --trained_checkpoint_prefix training/model.ckpt-XXXX --output_directory inference_graph**












W miejsce XXXX - należy podać największą wartość (ostatni step)

















Wyeksportowany wykres wnioskowania powinien znajdować się w folderze


C:/tensorflow/models/research/object_detection/inference_graph pod nazwą **frozen_inference_graph.pb**

UWZGLĘDNIONE ZNAKI DROGOWE

ZNAKI OSTRZEGAWCZE



Etykieta/kod znaku	Nazwa	Zdjęcie
A-1	Niebezpieczny zakręt w prawo	
A-2	Niebezpieczny zakręt w lewo	
A-3	Niebezpieczne zakręty, pierwszy w prawo	
A-4	Niebezpieczne zakręty, pierwszy w lewo	
A-5	Skrzyżowanie dróg	
A-6a	Skrzyżowanie z drogą podporządkowaną występującą po obu stronach	
A-6b	Skrzyżowanie z drogą podporządkowaną występującą po prawej stronie	
A-6c	Skrzyżowanie z drogą podporządkowaną występującą po lewej stronie	
A-6d	Wlot drogi jednokierunkowej z prawej strony	
A-6e	Wlot drogi jednokierunkowej z lewej strony	
A-7	Ustąp pierwszeństwa	

A-9	Przejazd kolejowy z zaporami	
A-11	Nierówna droga	
A-11a	Próg zwalniający	
A-12a	Zwężenie jezdni obustronne	
A-12b	Zwężenie jezdni prawostronne	
A-14	Roboty drogowe	
A-15	Śliska jezdnia	
A-16	Przejście dla pieszych	
A-17	Dzieci	
A-18b	Zwierzęta dzikie	
A-20	Odcinek jezdni o ruchu dwukierunkowym	
A-21	Tramwaj	
A-23	Stromy podjazd	
A-28	Sypki żwir	
A-29	Sygnały świetlne	
A-30	Inne niebezpieczeństwo	










A-32	Oszronienie jezdni	
------	--------------------	---

ZNAKI ZAKAZU




B-1	Zakaz ruchu w obu kierunkach	
B-2	Zakaz wjazdu	
B-5	Zakaz wjazdu samochodów ciężarowych	
B-16	Zakaz wjazdu pojazdów o wysokości ponad ... m	
B-20	STOP	
B-21	Zakaz skrętu w lewo	
B-22	Zakaz skrętu w prawo	
B-23	Zakaz zawracania	
B-25	Zakaz wyprzedzania	
B-31	Pierwszeństwo dla nadjeżdżających z przeciwka	
B-33	Ograniczenie prędkości	
B-35	Zakaz postoju	
B-36	Zakaz zatrzymywania się	
B-41	Zakaz ruchu pieszych	















B-42	Koniec zakazów	
B-43	Strefa ograniczonej prędkości	

ZNAKI NAKAZU

C-2	Nakaz jazdy w prawo za znakiem	
C-4	Nakaz jazdy w lewo za znakiem	
C-5	Nakaz jazdy prosto	
C-6	Nakaz jazdy prosto lub w prawo	
C-7	Nakaz jazdy prosto lub w lewo	
C-8	Nakaz jazdy w prawo lub w lewo	
C-9	Nakaz jazdy z prawej strony znaku	
C-10	Nakaz jazdy z lewej strony znaku	
C-12	Ruch okrężny	

ZNAKI INFORMACYJNE

D-1	Droga z pierwszeństwem	
D-2	Koniec drogi z pierwszeństwem	
D-3	Droga jednokierunkowa	

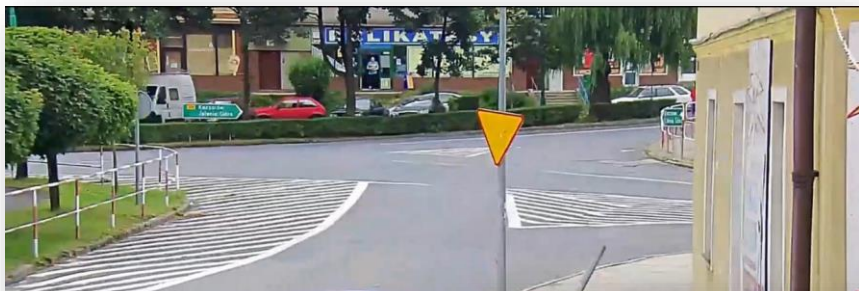
D-4a	Droga bez przejazdu	
D-5	Pierwszeństwo na zwężonym odcinku jezdni	
D-6	Przejście dla pieszych	
D-6b	Przejście dla pieszych i przejazd dla rowerzystów	
D-7	Droga ekspresowa	
D-8	Koniec drogi ekspresowej	
D-9	Autostrada	
D-10	Koniec autostrady	
D-18	Parking	
D-40	Strefa zamieszkania	
D-41	Koniec strefy zamieszkania	
D-42	Obszar zabudowany	
D-43	Koniec obszaru zabudowanego	
D-51	Automatyczna kontrola prędkości	

D-52	Strefa ruchu	 Strefa ruchu
------	--------------	---

PRZYKŁAD DZIAŁANIA

NA BAZIE ZDJĘĆ

Przed:



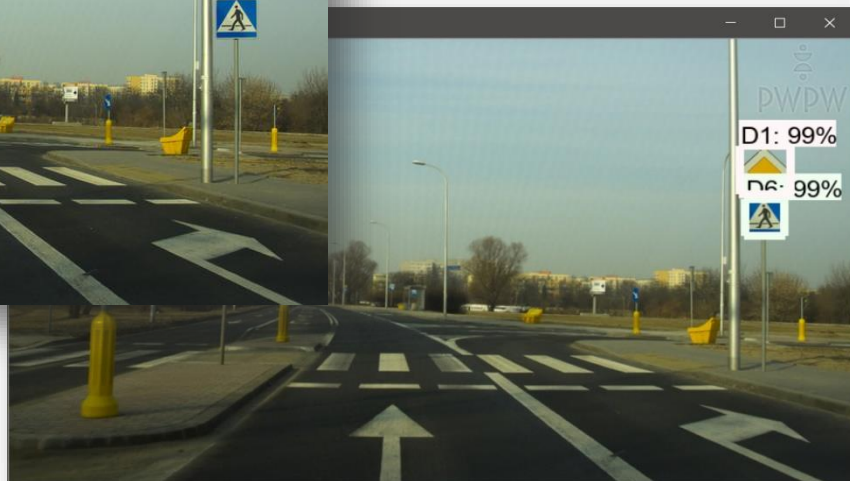
Po:



Przed:



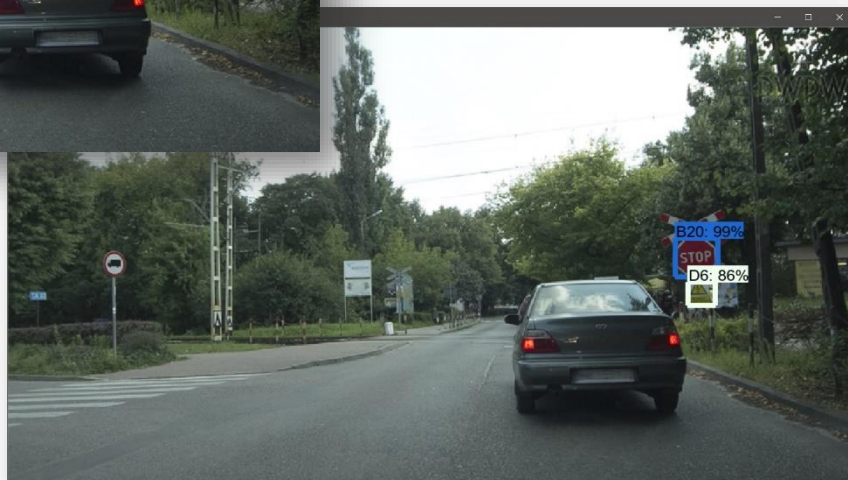
Po:



Przed:



Po:



NA BAZIE FILMÓW

Film w serwisie YouTube prezentujący działanie w ciągu dnia:

https://youtu.be/Ly_NLAvLdu0

Film w serwisie YouTube prezentujący działanie w różnych warunkach oświetleniowych:

<https://youtu.be/A-vrJHdpYkk>