

Sztuczna Inteligencja

Sieć neuronowa klasyfikująca pojazdy

Projekt

Maciej STANUCH
171995

Mikołaj TRYLEWICZ
171582

Przemysław RESZKA
172038

14 maja 2019

1 Cel projektu

Celem projektu było zaprojektowanie i zbudowanie sieci neuronowej potrafiącej zaklasyfikować pojazdy znajdujące się na zdjęciach do jednej z czterech klas pojazdów lub rozpoznać pieszego.

2 Przygotowanie Danych

Pierwszym krokiem jest przygotowanie bazy zdjęć, pieszych oraz pojazdów różnych typów, które zostaną użyte do nauki oraz późniejszego testowania.

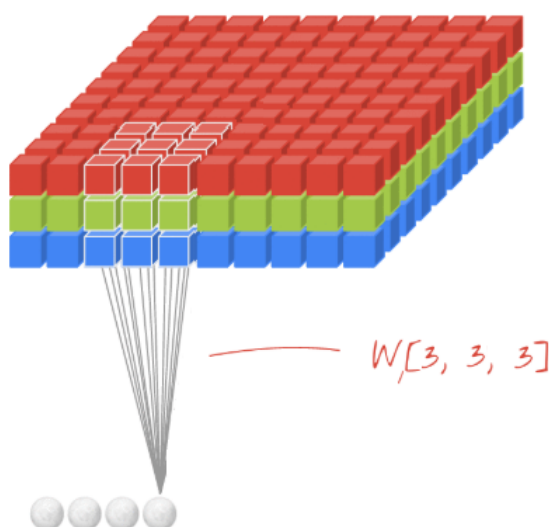
Klasa	Zestaw treningowy	Zestaw walidujący
Autobusy	5000	2000
Samochody osobowe	5000	2000
Jednoślady	3000	1000
Ciężarówki	5000	2000
Piesi	4000	1000
Razem	22000	8000

Użyty zbiór zawiera zdjęcia zrobione z różnych kątów oraz w różnej jakości. Dodatkowo mają one różne rozmiary. Konieczne zatem jest przeskalowanie ich do wymaganej wielkości tj. 96x96 pikseli. Będą one także rozbijane na trzy kanały (RGB) wewnątrz sieci.

3 Projekt Sieci

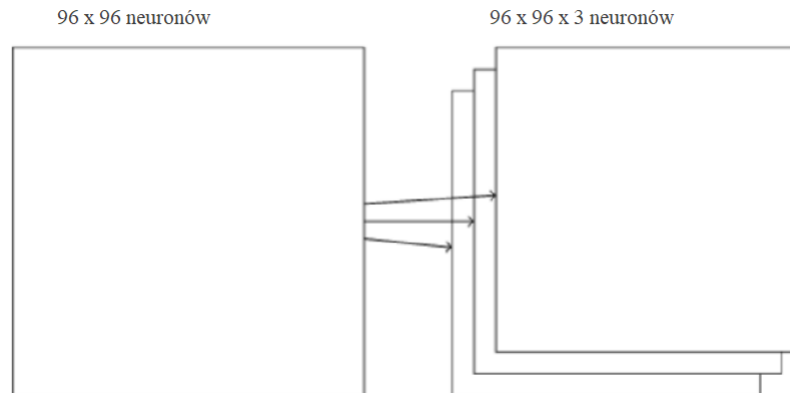
3.1 Wstępne założenia

Idealnym rozwiązaniem wydaje się zastosowanie sieci konwolucyjnej. Jej idea polega na przefiltrowaniu sygnału poruszając się po nim w ramach mniejszego okna, tak aby uchwycić cechy z sąsiedztwa. W tym przypadku będzie to 3x3 piksele, zatem dla obrazu o wymiarach 96x96, sieć konwolucyjna będzie zawierać 96x96x3 jednostki w warstwie konwolucyjnej.



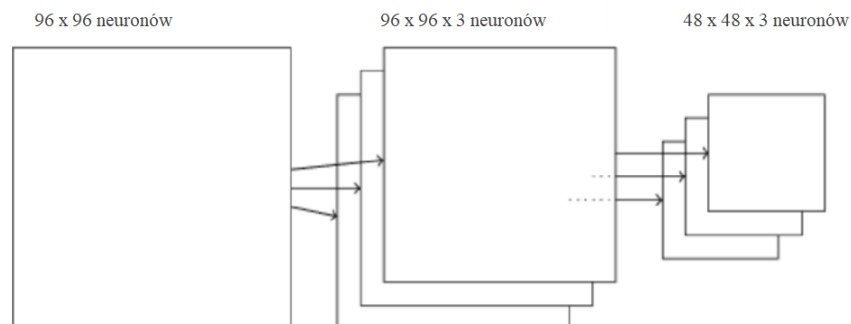
Rysunek 1: Warstwa konwolucyjna

Jedna warstwa zwykle odpowiada jednej cesze, dlatego w sieci występuje wiele warstw. W ten sposób można wydobywać kolejne cechy obrazu. Dla większej liczby cech potrzebna jest większa liczba warstw konwolucyjnych.



Rysunek 2: Warstwa konwolucyjna

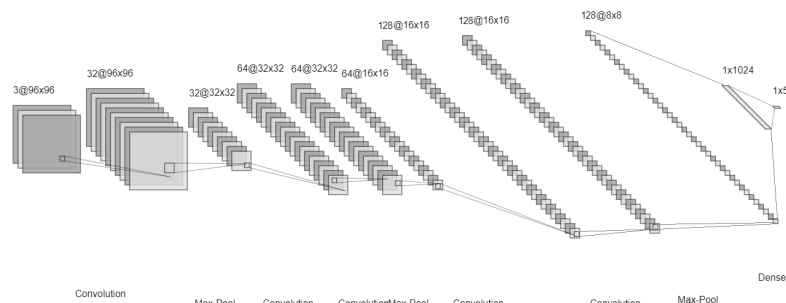
Pociąga to za sobą jednak pewne skutki. Sąsiednie jednostki częściowo pokrywają te same obszary danych wejściowych, pojawia się redundancja. W celu poradzenia sobie z tym zjawiskiem wprowadzamy kolejną warstwę tzw. pooling layer. Ma ona za zadanie uogólnić informację wydobytą przez jednostkę. W naszym przypadku poprzez wybranie wartości maksymalnej na zadanym obszarze np. 3×3 . Dodatkowo warstwa pooling zmniejsza wymiarowość problemu.



Rysunek 3: Zastosowanie pooling layer

3.2 Ostateczny model i Implementacja

Kolejnym krokiem po ustaleniu ogólnej budowy sieci było dobranie wielkości warstw konwolucyjnych oraz warstw pooling metodą prób i błędów.

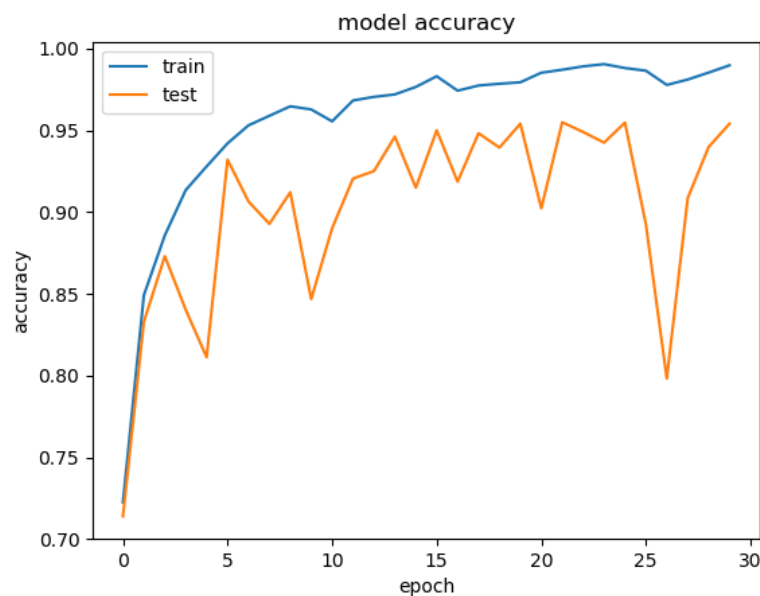


Rysunek 4: Dokładny model sieci

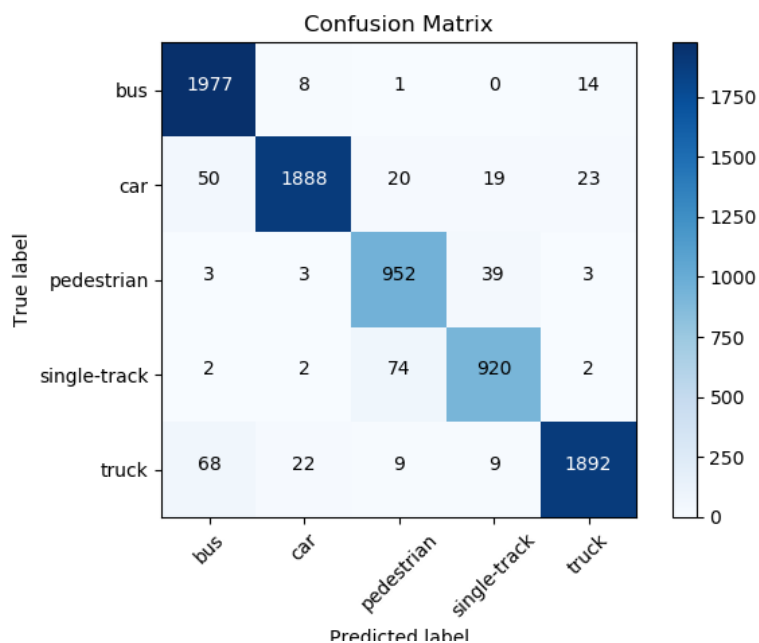
Implementacja wykonana została w języku python z użyciem biblioteki Keras (z backendem Tensorflow). Sieć uczona była optymalizatorem Adam, z funkcją straty MSE.

Kolejne paczki testowe były tworzone poprzez losowanie 128 obrazów z puli. W celu uniknięcia przeuczenia i przyspieszenia obliczeń zastosowany został Early Stopping.

4 Otrzymane Wyniki



Rysunek 5: Skuteczność modelu w kolejnych epokach



Rysunek 6: Wyniki klasyfikacji

Zauważalna jest niższa skuteczność w rozpoznawaniu jednoślądów oraz pieszych. Spowodowana jest tym że sieć rozpoznaje człowieka kierującego jednoślądem i dopasowuje go do obu kategorii. Zsumowanie tych klas dałoby skuteczność rzędu 99%.

Klasa	Skuteczność
Autobusy	98.9%
Samochody osobowe	94.4%
Jednoślądy	92.0%
Ciężarówki	94.6%
Piesi	95.2%
Łącznie	95.4%

Łączna skuteczność sieci w rozpoznawaniu pojazdów i pieszych wyniosła 95.4%. Biorąc pod uwagę wcześniej wspomniany błąd wynik ten jest bardzo wysoki.