|  |  |
| --- | --- |
| Projekt – sieci neuronowe | Data złożenia projektu: 13.05.2022 |
| Numer grupy projektowej: CWP 4 | Imię i nazwisko I: Szymon Musiał  Imię i nazwisko II: Piotr Wilkosz |

TSR – System rozpoznawania znaków (polskich)

# Opis problemu i danych

Ogólny opis dziedziny, jakiej dotyczą dane, ilość rekordów, nazwy i ilość zmiennych wejściowych z podziałem na jakościowe i ilościowe. Podstawowe statystyki (min, max, średnia, odchylenie standardowe) i histogram dla zmiennej wyjściowej. Jasne określenie, jaki problem jest rozwiązywany (regresja, klasyfikacja, segmentacja).

W ramach projektu poruszana jest dziedzina klasyfikacji obrazów. Klasyfikowanymi obrazami są polskie znaki drogowe. Celem jest nauczenie sieci poprawnej klasyfikacji zdjęcia znaku do symbolu znaku.

Zdjęcia z kamery przedniej samochodu, skalsyfikowane przez model mogą być pomocą dla kierowcy.

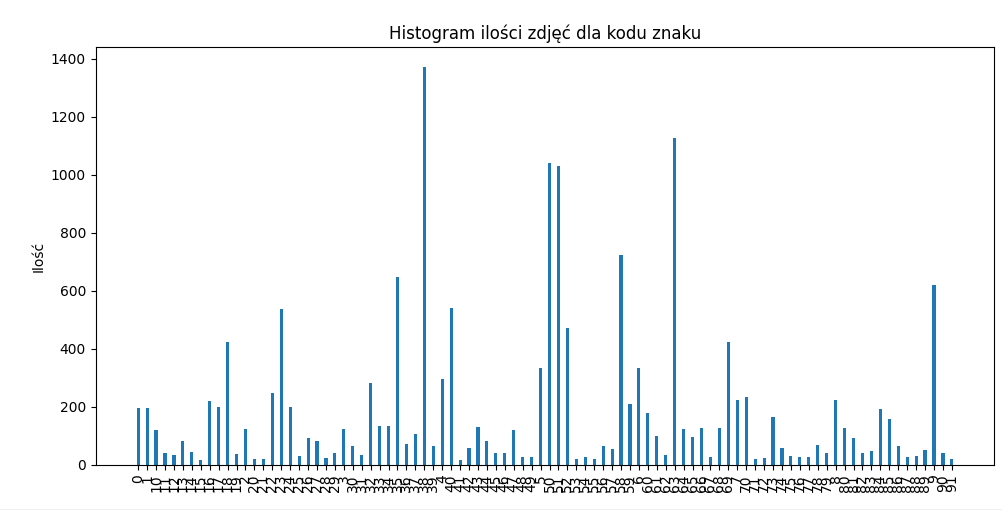
Analizowany zbiór danych składa się z 21 tysięcy rekordów. Znajdziemy w nim 92 zmienne wejściowe. Liczność wybranych zmiennych zaprezentowana jest na poniższym histogramie. Problematyczne staje się przedstawienie podstawowych statystyk ze względu, iż rozpatrywany jest problem klasyfikacji obrazów.

Pod względem liczności rekordów w klasie możemy zauważyć, że maksymalna ilość obrazów w klasie wynosi 1370 dla klasy 38 o symbolu B-33 – Ograniczenie prędkości.

Przechodząc do analizy klasy o minimalnej ilości rekordów możemy wyróżnić klasę 15 o symbolu A15 – Śliska jezdnia o liczności równej 17.

Średnia ilość rekordów wynosi 182 rekordy.

Odchylenie standardowe wynosi 255.



Wykres 1. Ilość zdjęć dla danego kodu znaku

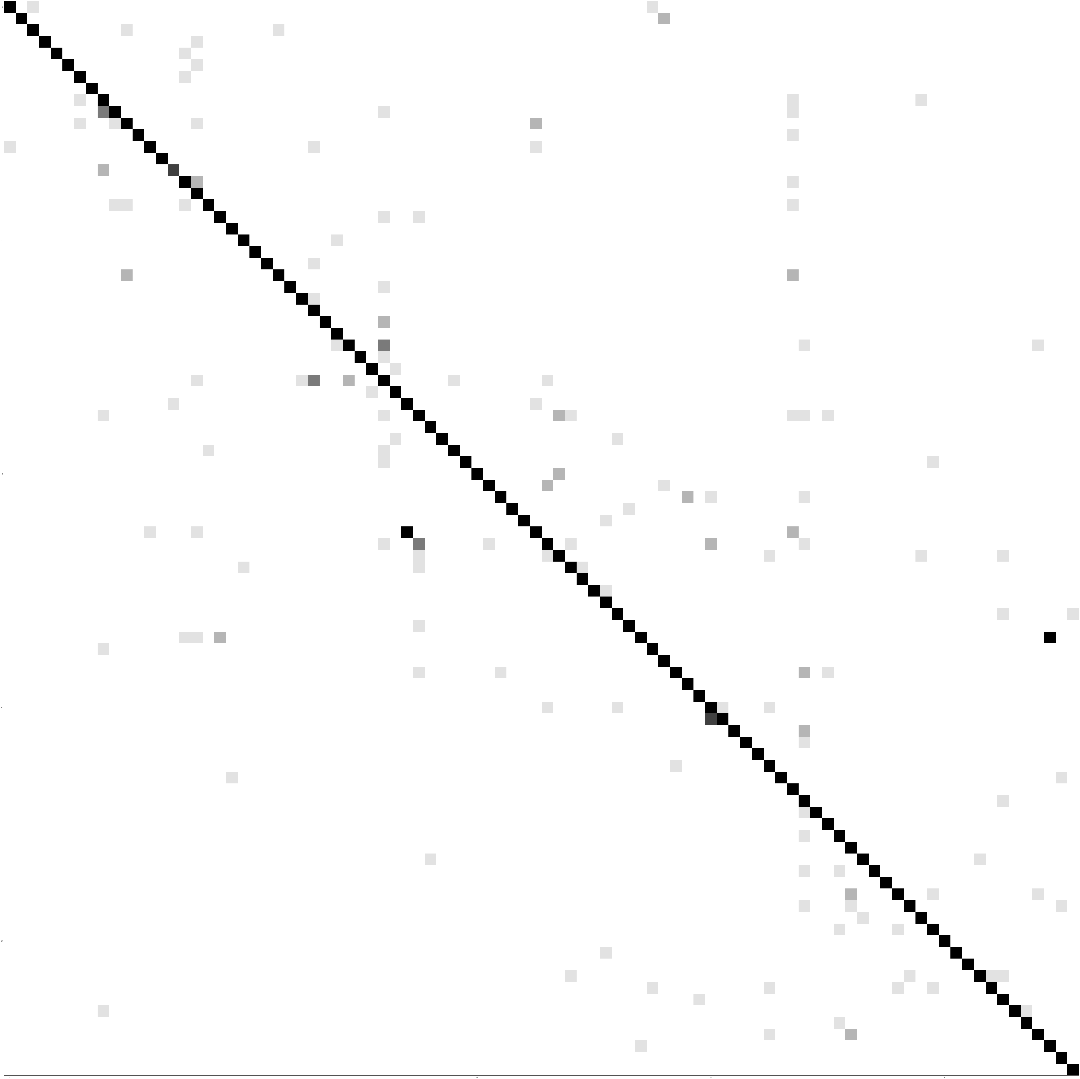
# Obróbka danych

Ilość brakujących rekordów i sposób rozwiązania tego problemu, normalizacja/standaryzacja danych wejściowych, transformaty wykonane na danych (zależnie od problemu, np. dla szeregów czasowych średnia z ostatnich 10 rekordów, dla rozpoznawania obrazów przetworzenie przez filtry graficzne). Sposób podziału na zbiór uczący oraz zbiór testujący.

Otrzymany zbiór posiadał już podział na testujący oraz treningowy. Ze zbioru treningowego zostało przeniesione 10% zdjęć do zbioru walidującego używanego przy trenowaniu.

Ponadto zbiór testujący po wytrenowaniu sieci został użyty w celi walidacji dokładności

Poniżej zaprezentowano macierz po odfiltrowaniu zdjęć złej jakości, na której to osi X znajduje się klasa przypisana do zdjęcia testującego a do osi Y klasyfikacja modelu. Odchyłki od głównej przekątnej stanowią błędy dopasowania. Kolor elementów macierzy jest szary dla 5 elementów



Rys 1. Graficzna macierz dopasowania modelu

Macierz ta była decyzyjną w procesie filtrowania, a zaprezentowana powyżej jest już wynikową.

Ponadto zastosowany ImageDataGenerator od razu dokonuje powiększenia zbioru stosując np. przesunięcie, skalowanie, rotacje i tym podobne przekształcenia.

# Opis zastosowanych sieci neuronowych

Typ i architektura sieci, ilość epok uczenia, sposób uczenia sieci. Informacja o własnej implementacji/ użytym frameworku. Co najmniej 2-3 różne architektury sieci.

W projekcie zastosowano konwolucyjną sieć neuronową

Ilość epok uczenia została empirycznie dobrana na 25, mniejsza liczba powodowała niską dokładność a większa powodowała przetrenowanie co skutkowało rozbieżnością na poziomie kilkudziesięciu procent w testach dokładności na danych testujących i walidujących podczas uczenia modelu

Do uczenia użyto frameworku keras który jako backend używa tensorflow 2, który natomiast wykorzystuje kartę graficzną w celu przyspieszenia uczenia.

# Dyskusja wyników oraz wnioski

Zebrane w tabeli wyniki sieci z podziałem na zbiór uczący i testujący. Co najmniej 2 miary, wraz z komentarzem, dlaczego zostały wybrane takie, a nie inne. Określenie (subiektywne), czy wyniki są satysfakcjonujące wraz z uzasadnieniem. Wnioski, dalsze propozycje rozwoju projektu.