

# **Klasyfikacja obrazów z użyciem biblioteki Keras**

**Implementacja prostej konwolucyjnej sieci neuronowej**

**Jakub Kowalewski, czerwiec 2024 r.**

# Zbiór danych

- Znaki drogowe
- 43 klasy
- Rozdzielczość 28 x 28

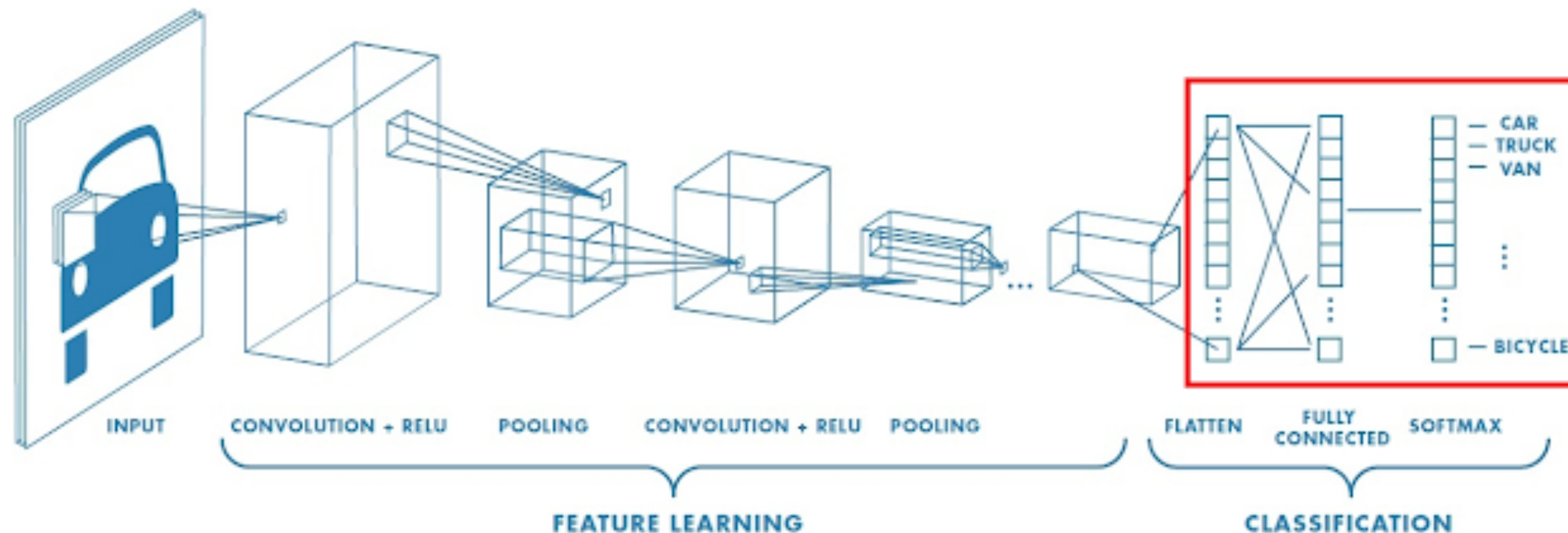


# Wykorzystane biblioteki

- **tensorflow** - przygotowanie modelu, przeprowadzenie predykcji
- **sklearn** - podzielenie zbioru na podzbiór treningowy (train set) i walidacyjny (validation set)
- **pandas** - praca ze zbiorem danych - wczytanie zdjęć, dataframe'y
- **numpy**

# Model sieci neuronowej

1. Dwie dwuwymiarowe warstwy konwolucyjne (Conv2D) - 32 filtry i 64 filtry
2. Funkcja aktywacji - ReLu
3. Próbkowanie w dół - MaxPooling2D
4. Przekształcenie w wektor - Flatten
5. Klasyfikacja - Dense (FC layer, warstwa *gęsta*)



# Model sieci neuronowej

```
[19] 1 model = Sequential(  
2     layers =  
3     [layers.Conv2D(filters=32, kernel_size=(3, 3), activation='relu'),  
4       layers.Conv2D(64, kernel_size=(3, 3), activation='relu'),  
5       layers.MaxPooling2D(pool_size=(2, 2)),  
6       layers.Flatten(),  
7       layers.Dense(128, activation='relu'),  
8       layers.Dense(43, activation='softmax')]  
9 )
```

Executed at 2024.06.06 22:12:44 in 13ms

```
[29] 1 model.compile(  
2     loss = losses.CategoricalCrossentropy(), # Funkcja straty  
3     optimizer = 'Adam',  
4     metrics= ['accuracy'] # Accuracy = Number of Correct Predictions / Total Predictions Made  
5 )
```

Executed at 2024.06.06 22:18:45 in 103ms




# Wyniki


```
model.fit(  
    X_train,  
    y_train,  
    epochs= 3,  
    validation_data = (X_valid, y_valid)  
)
```

Executed at 2024.06.06 22:13:51 in 1m 7s 559ms


Epoch 1/3

**919/919**  **21s** 22ms/step - accuracy: 0.6021 - loss: 1.5619 - val\_accuracy: 0.9424 - val\_loss: 0.2181

Epoch 2/3

**919/919**  **24s** 26ms/step - accuracy: 0.9672 - loss: 0.1246 - val\_accuracy: 0.9600 - val\_loss: 0.1430

Epoch 3/3

**919/919**  **23s** 24ms/step - accuracy: 0.9856 - loss: 0.0556 - val\_accuracy: 0.9800 - val\_loss: 0.0782