

---

# Detekcja znaków drogowych CNN

Autor: Kamil Barszczak

---

## Podejście 'sliding window'

Schemat działania

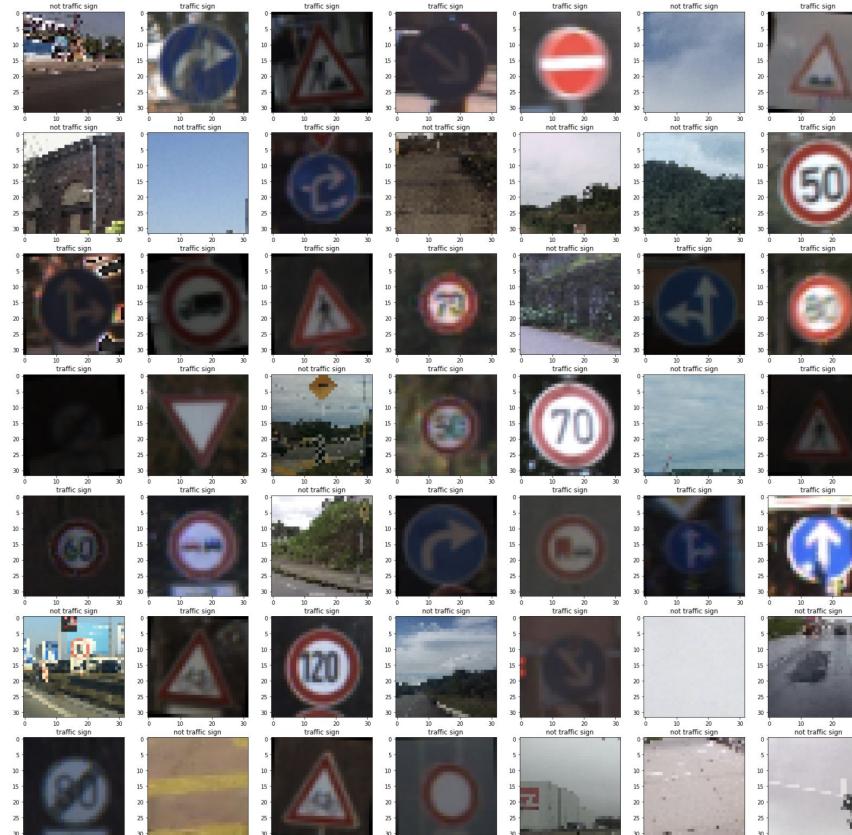
1. Zainicjalizowanie okna, które przesuwa się po całym obrazie wejściowym
2. Klasyfikowanie binarne wyciętego okna
3. Jeśli klasyfikator binarny wykrył znak drogowy to sklasyfikuj ten znak i zapamiętaj okno, w którym znak został wykryty



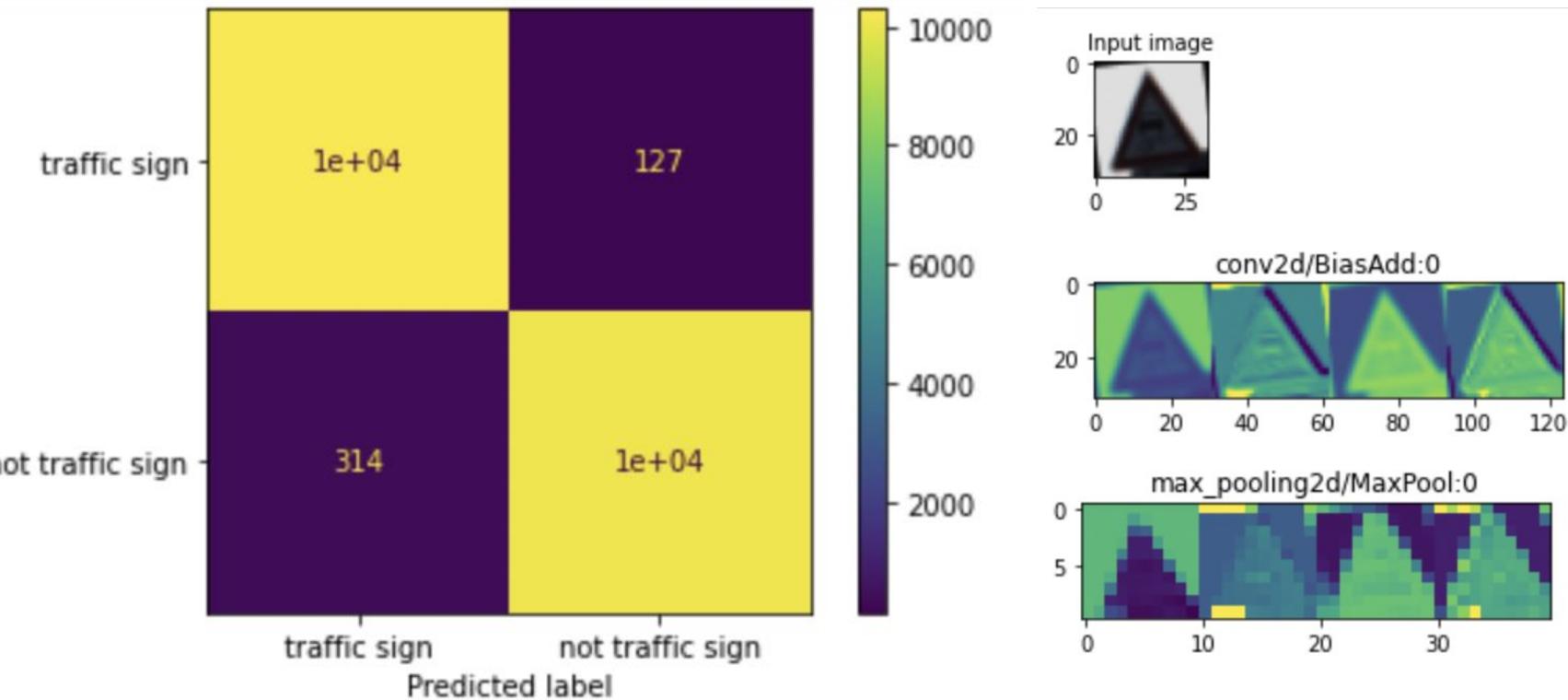
# Klasyfikacja binarna

| Layer (type)                  | Output Shape      | Param # |
|-------------------------------|-------------------|---------|
| <hr/>                         |                   |         |
| conv2d (Conv2D)               | (None, 31, 31, 4) | 52      |
| max_pooling2d (MaxPooling2D ) | (None, 10, 10, 4) | 0       |
| flatten (Flatten)             | (None, 400)       | 0       |
| dense (Dense)                 | (None, 1)         | 401     |
| <hr/>                         |                   |         |
| Total params: 453             |                   |         |
| Trainable params: 453         |                   |         |
| Non-trainable params: 0       |                   |         |

# Zbiór danych do treningu klasyfikatora binarnego

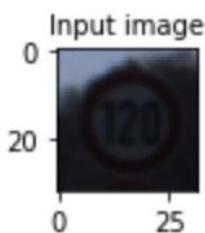


# Wyniki klasyfikacji binarnej

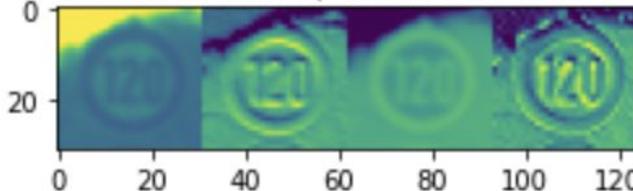


# Wyniki klasyfikacji binarnej

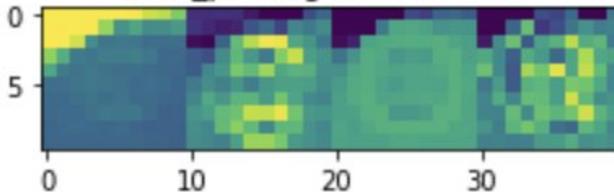
Input image



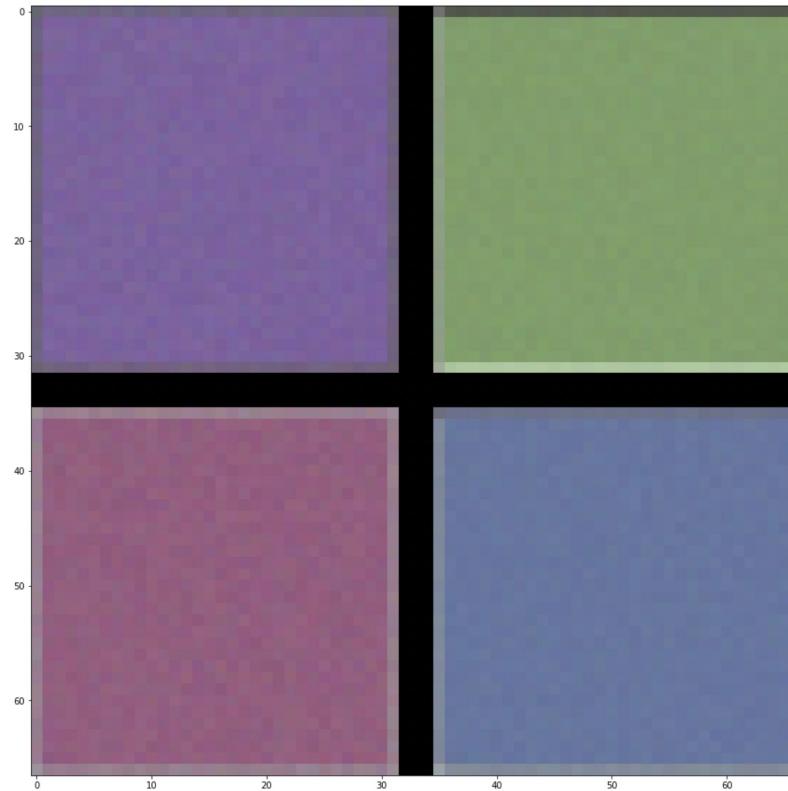
conv2d/BiasAdd:0



max\_pooling2d/MaxPool:0



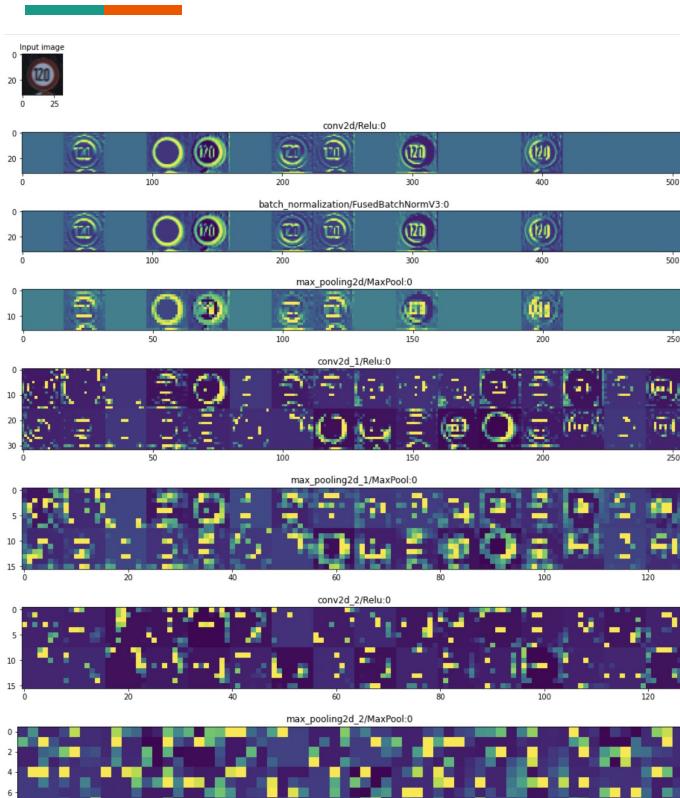
Layer name: conv2d



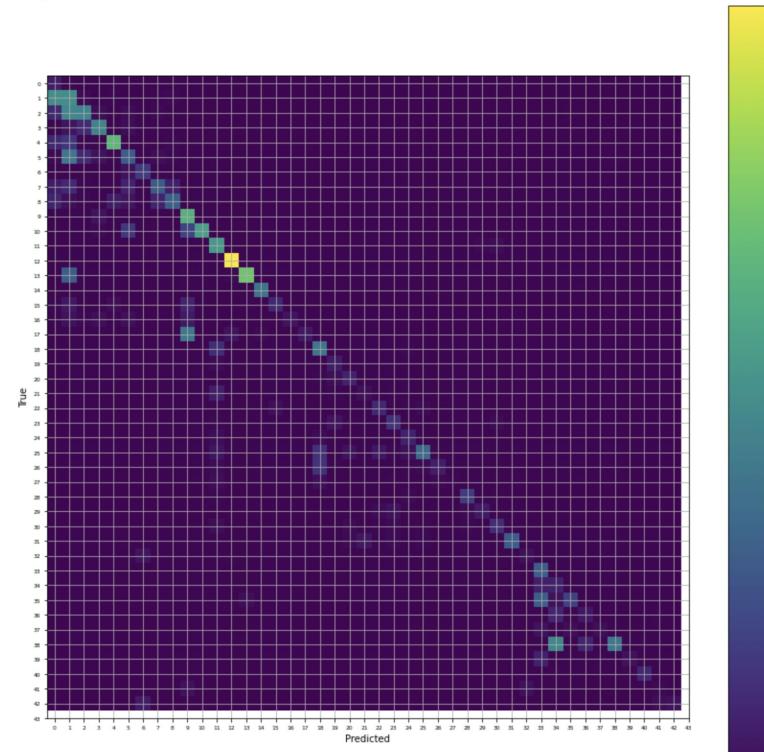
# Zbiór danych do treningu klasyfikatora wieloklasowego



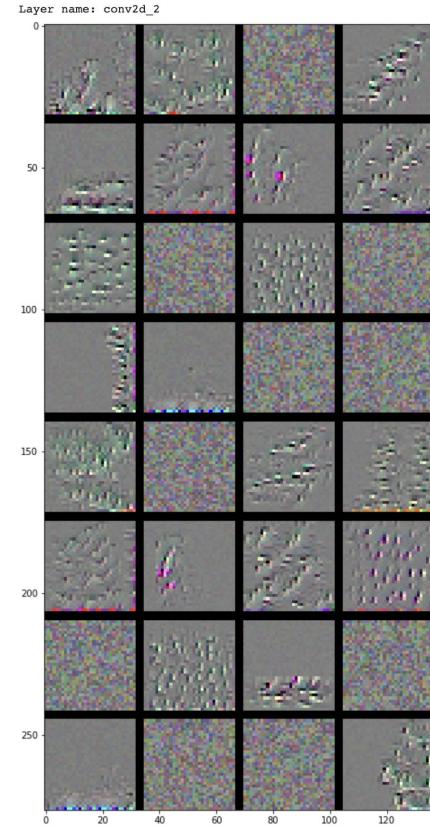
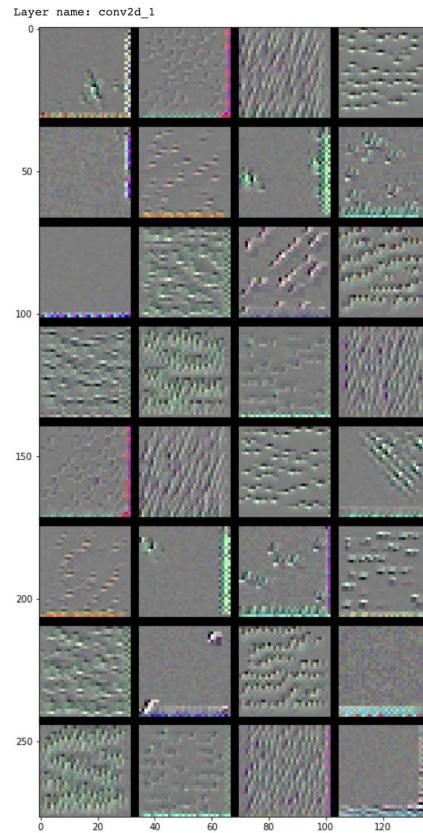
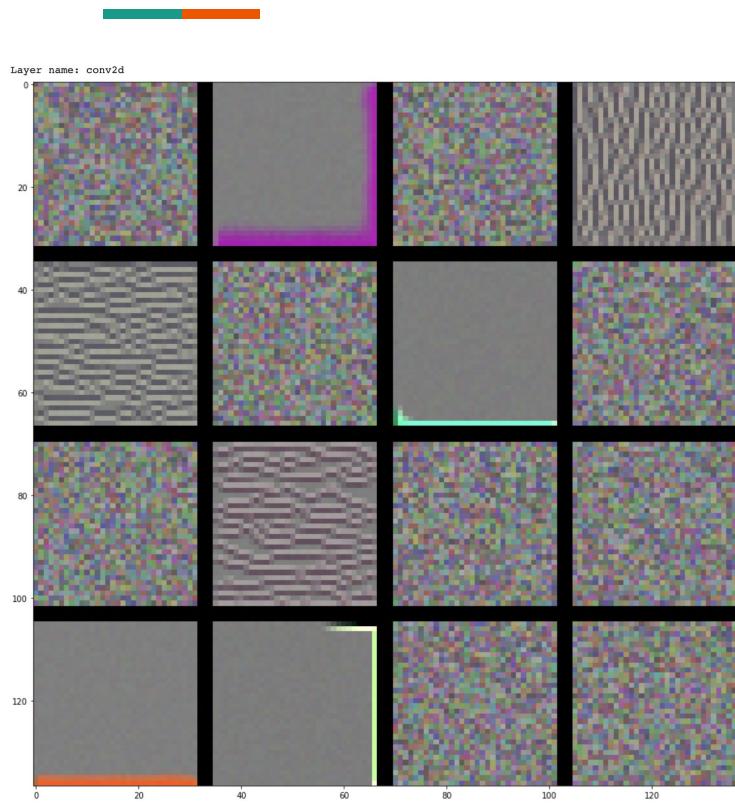
# Klasyfikacja wieloklasowa - model v1



<Figure size 432x288 with 0 Axes>



# Klasyfikacja wieloklasowa - model v1



---

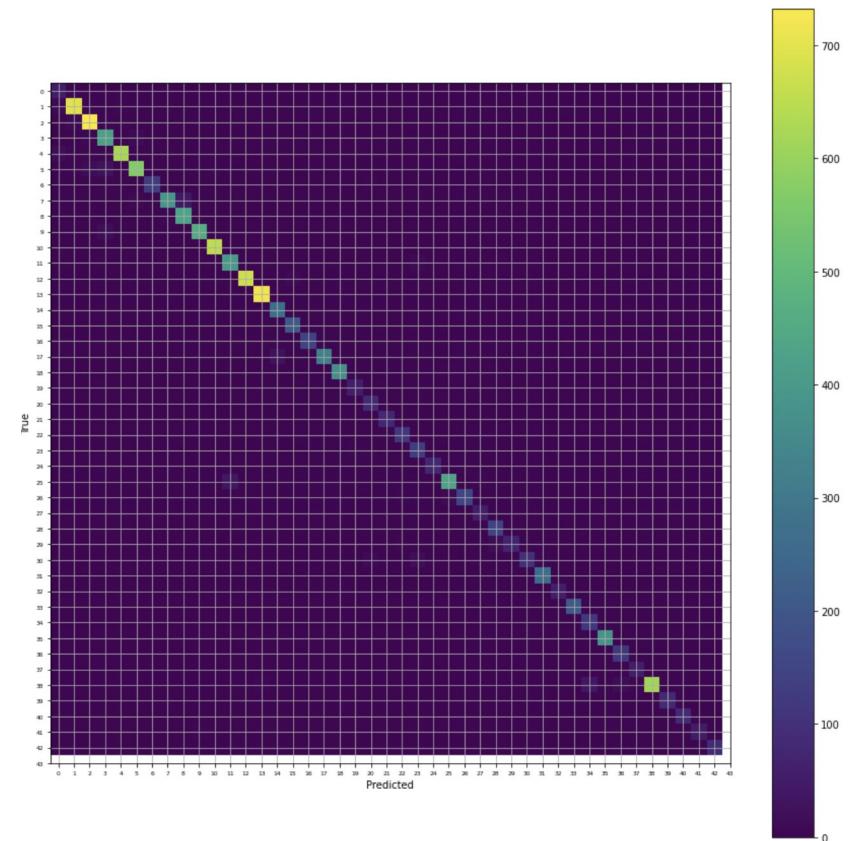
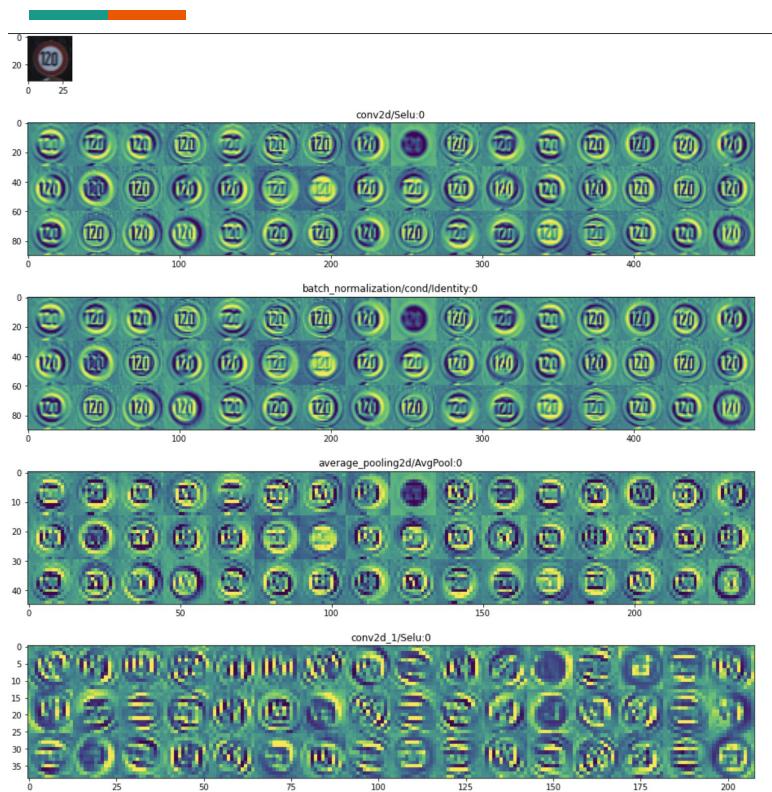
## Klasyfikacja wieloklasowa - model v3

Zmiany względem modelu v1

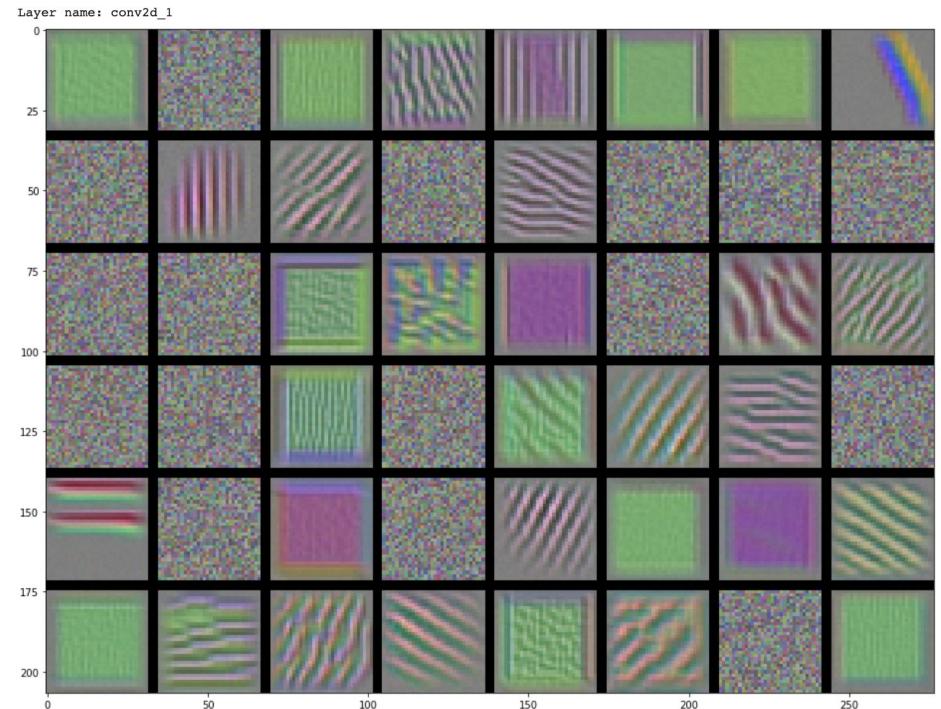
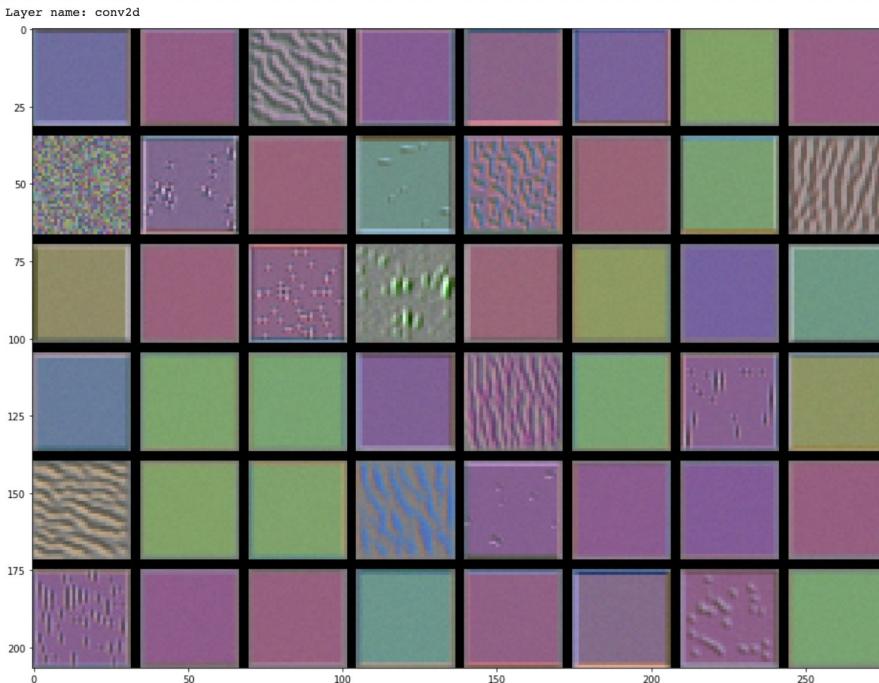
1. Dodana normalizacja wsadowa
2. Zwiększone wartości współczynników regularyzacji (0.003)
3. Zwiększyony dropout (0.18)
4. Zastosowanie selu jako funkcji aktywacji
5. Zredukowanie liczby warstw konwolucyjnych z 3 do 2
6. Zmniejszenie liczby filtrów
7. Zastosowanie AveragePooling zamiast MaxPooling

Efekty?

# Klasyfikacja wieloklasowa - model v3

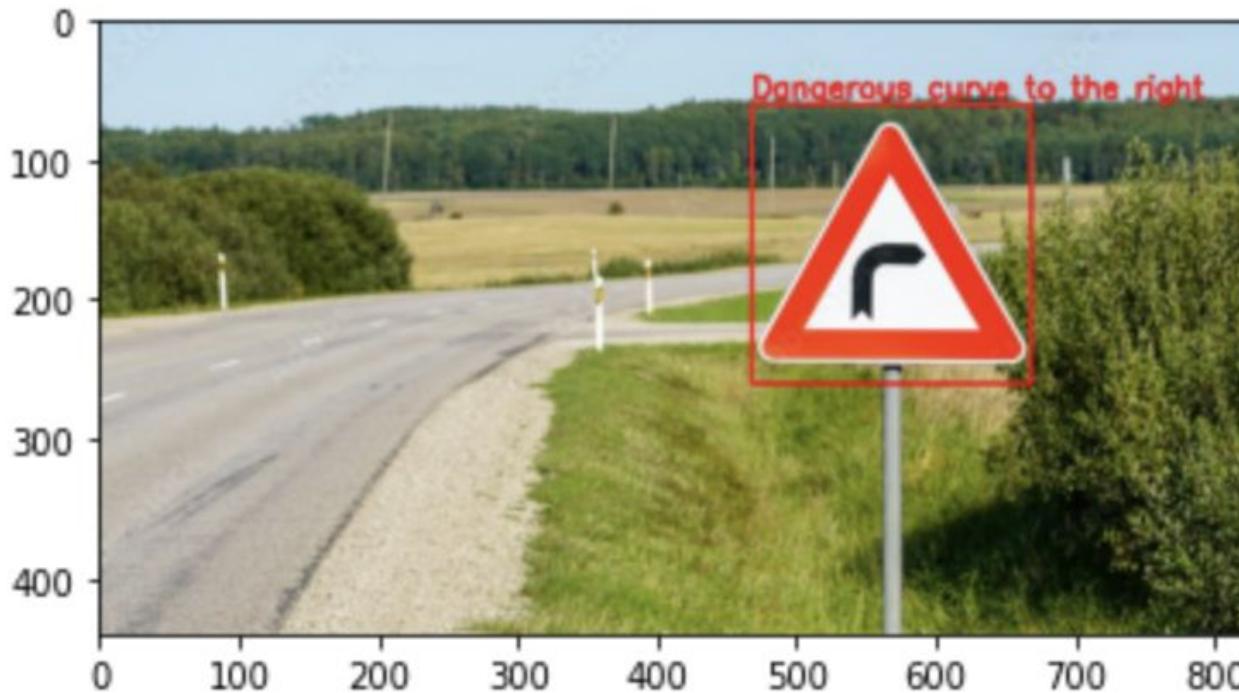


# Klasyfikacja wieloklasowa - model v3



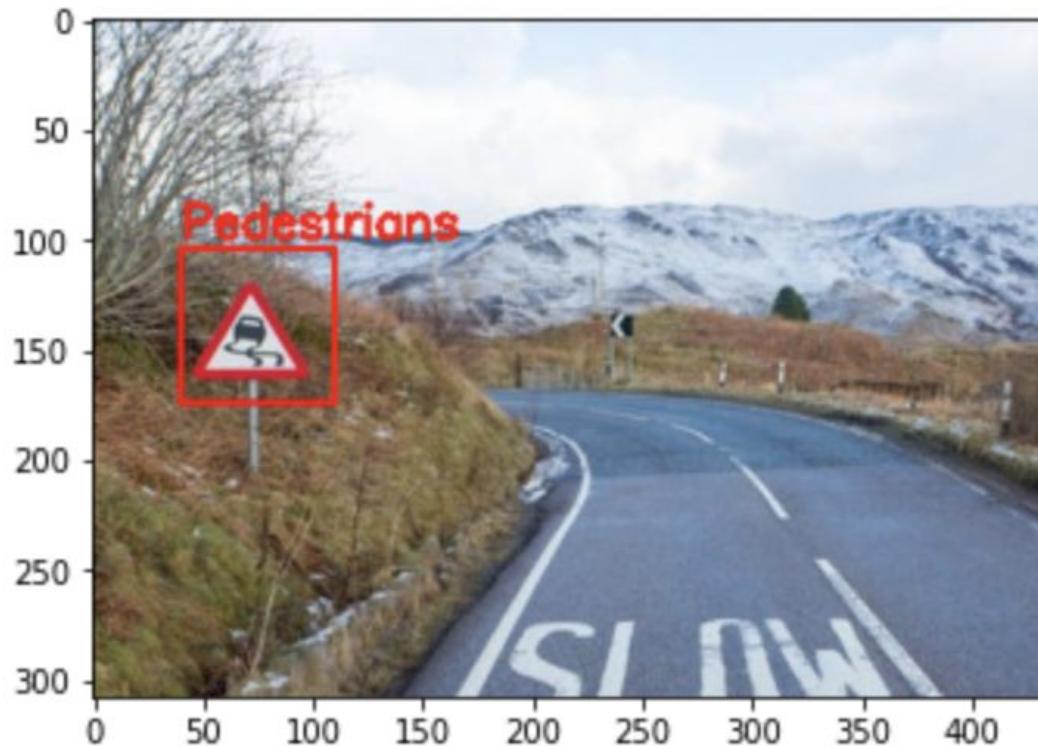
# Detekcja

---

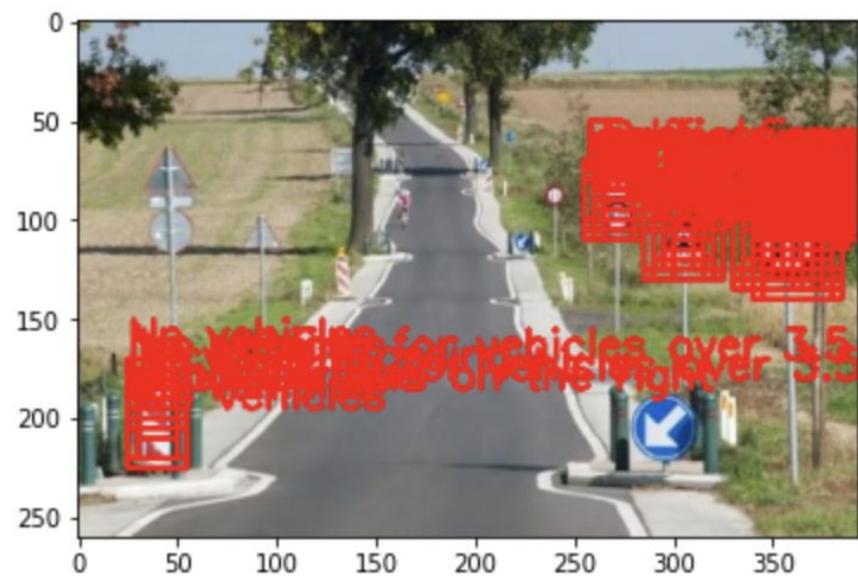


# Detekcja

---



# Detekcja



---

## Problemy i kolejne kroki

- 1. Długi czas oczekiwania na wynik
- 2. Brak mechanizmu likwidacji kolizji
- 3. Braki w rozpoznawanych znakach
- 1. Zastosowanie mechanizmu likwidującego kolizje
- 2. Wykorzystanie sieci YOLO

# Klasyfikacja wieloklasowa - model v9

---

Teraz:

- Test accuracy: 97.69%
- Test loss: 0.1

Wcześniej:

- Test accuracy: 96.41%
- Test loss: 0.18

```
model = models.Sequential()

model.add(layers.Conv2D(64, (4, 4), activation=tf.keras.layers.PReLU(), kernel_regularizer=regularizers.L1L2(),
                      input_shape=(img_rows, img_cols, 3), padding="valid"))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(.2))

model.add(layers.Conv2D(64, (4, 4), activation=tf.keras.layers.PReLU(), kernel_regularizer=regularizers.L1L2(),
                      padding="valid"))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(.2))
model.add(layers.Flatten())

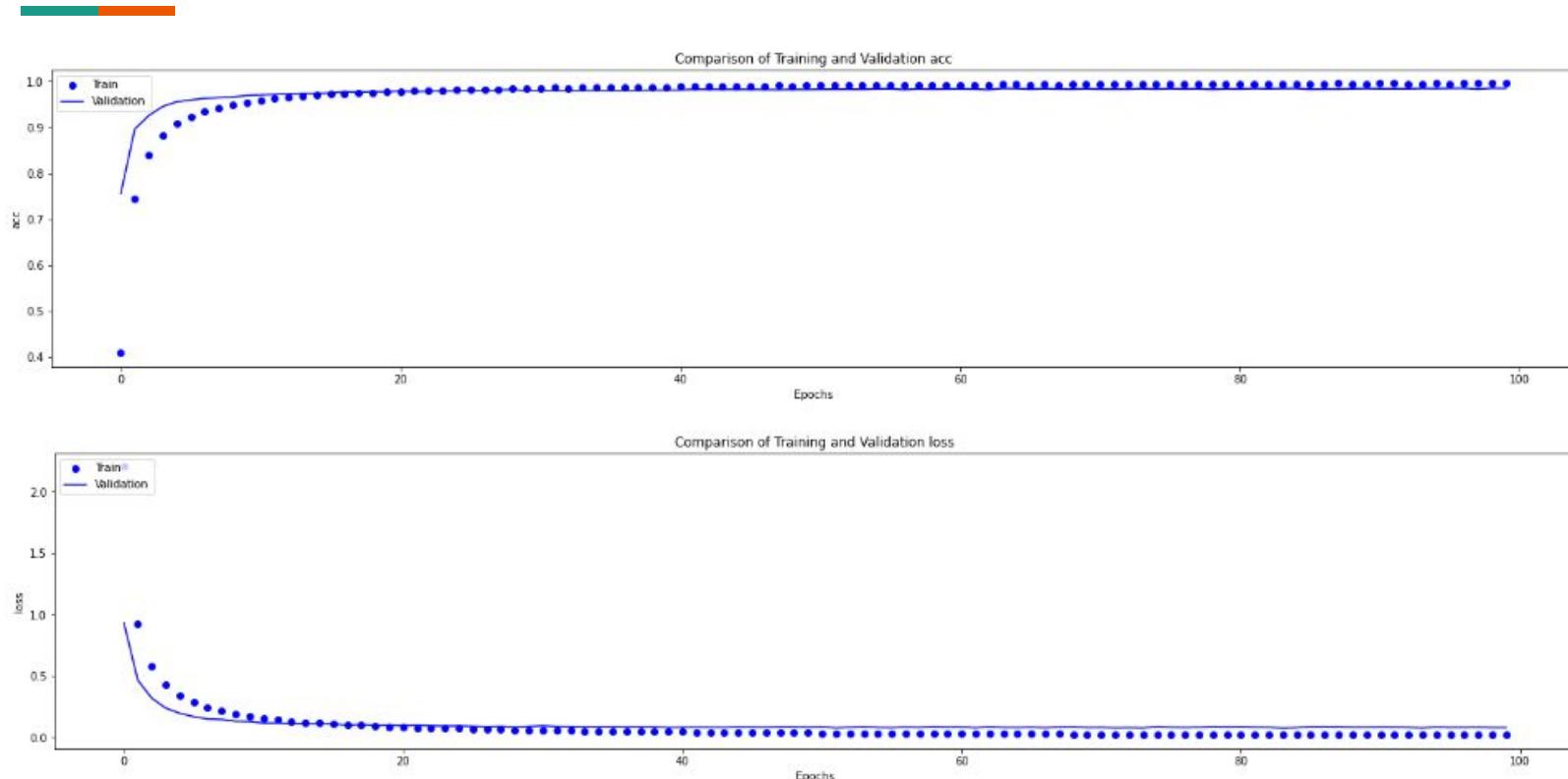
model.add(layers.Dense(output_size*2, kernel_regularizer=regularizers.L1L2(), activation='selu'))
model.add(layers.Dropout(.2))

model.add(layers.Dense(output_size*2, kernel_regularizer=regularizers.L1L2(), activation='selu'))
model.add(layers.Dropout(.2))

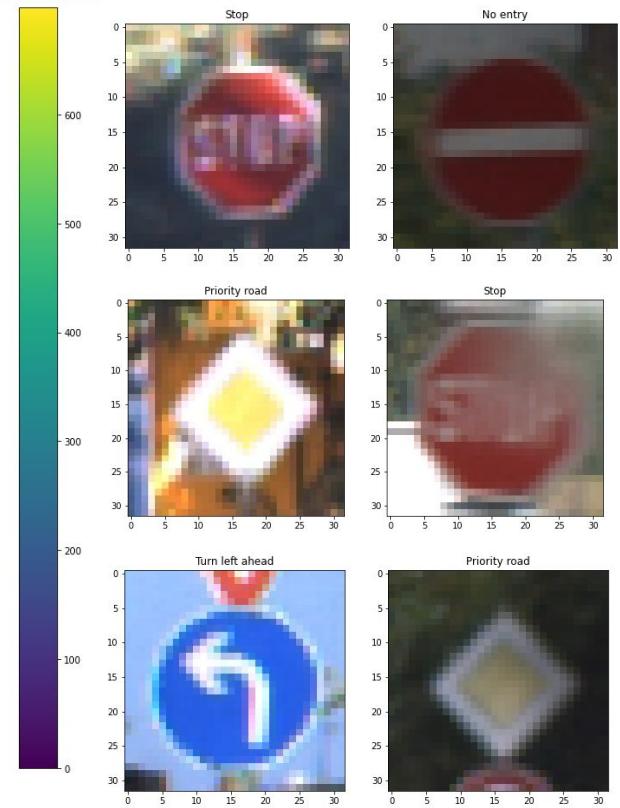
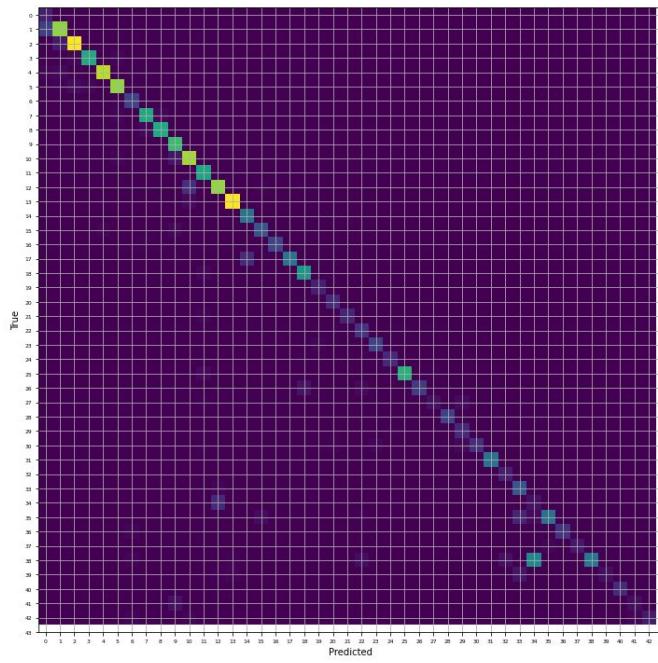
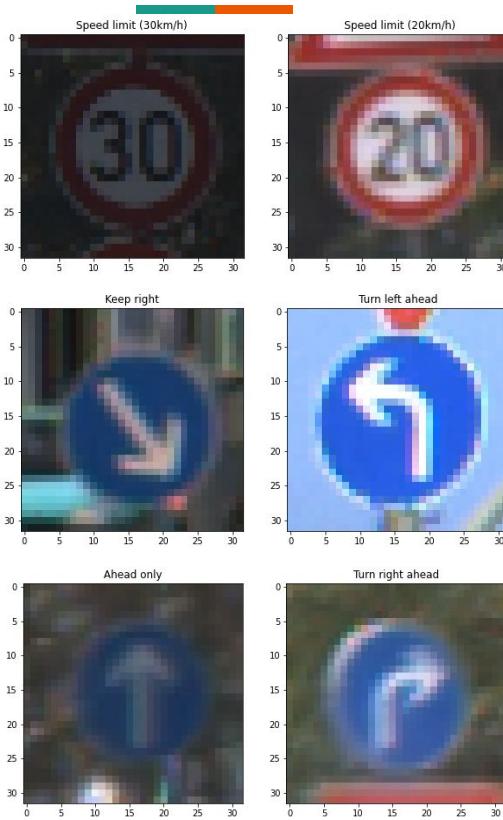
model.add(layers.Dense(output_size, activation='sigmoid'))

model.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
              optimizer=tf.keras.optimizers.SGD(learning_rate=0.0001, momentum=0.9),
              metrics=['acc'])
```

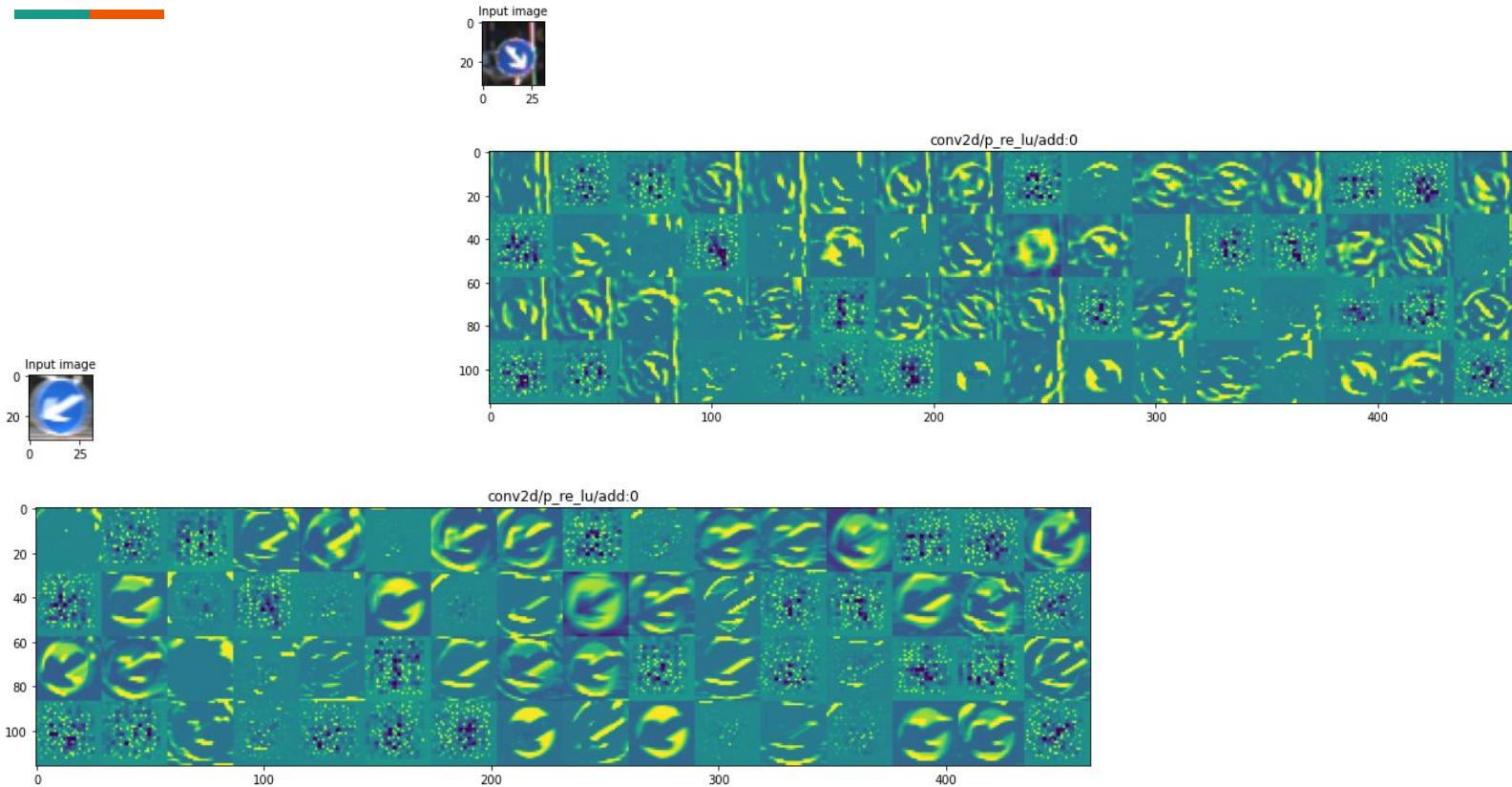
# Klasyfikacja wieloklasowa - model v9



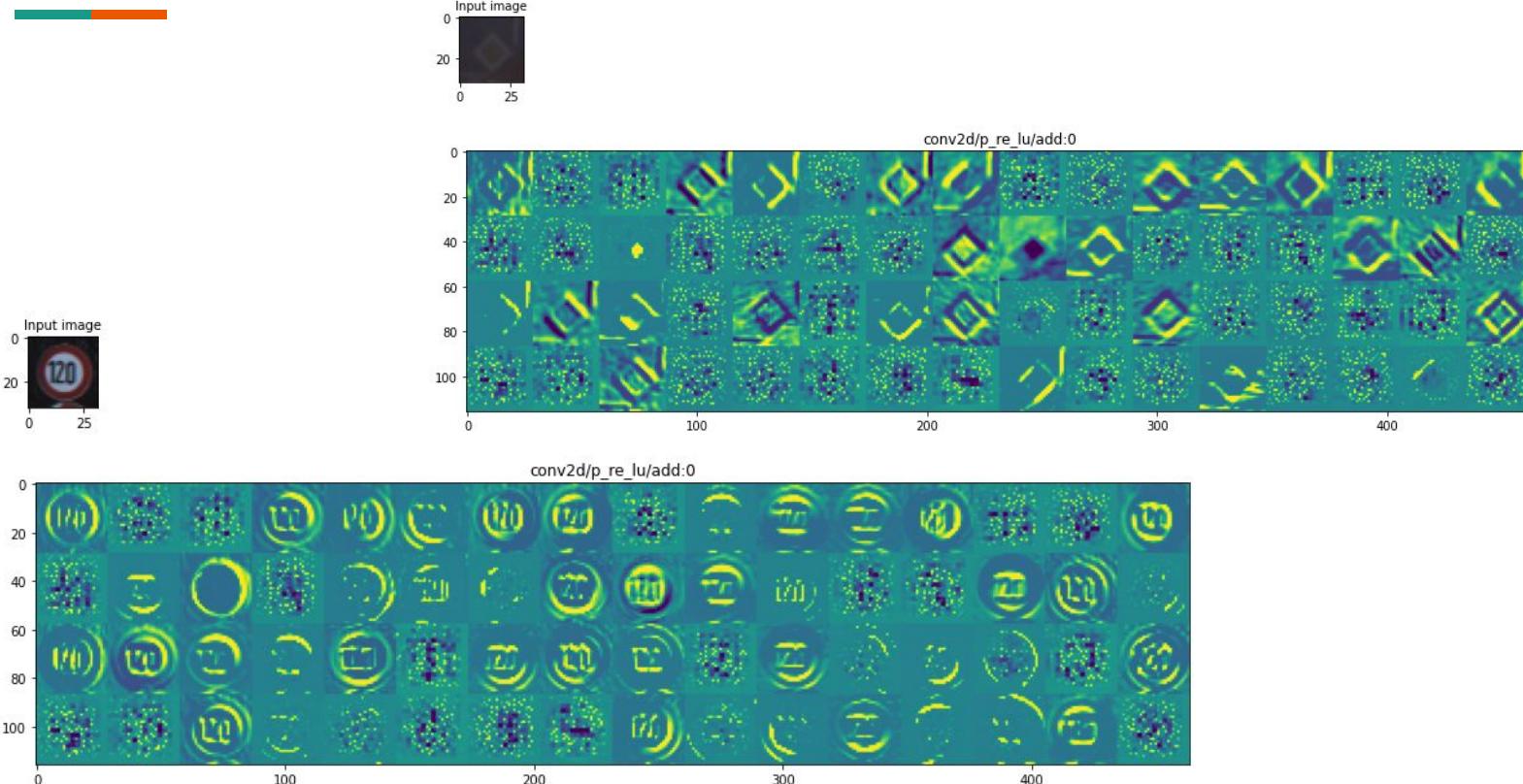
# Klasyfikacja wieloklasowa - model v9



# Klasyfikacja wieloklasowa - model v9



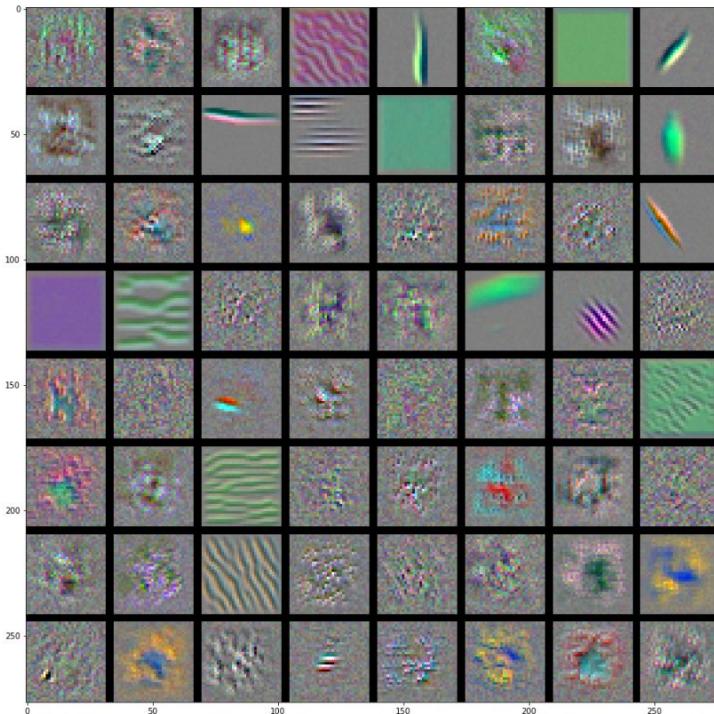
# Klasyfikacja wieloklasowa - model v9



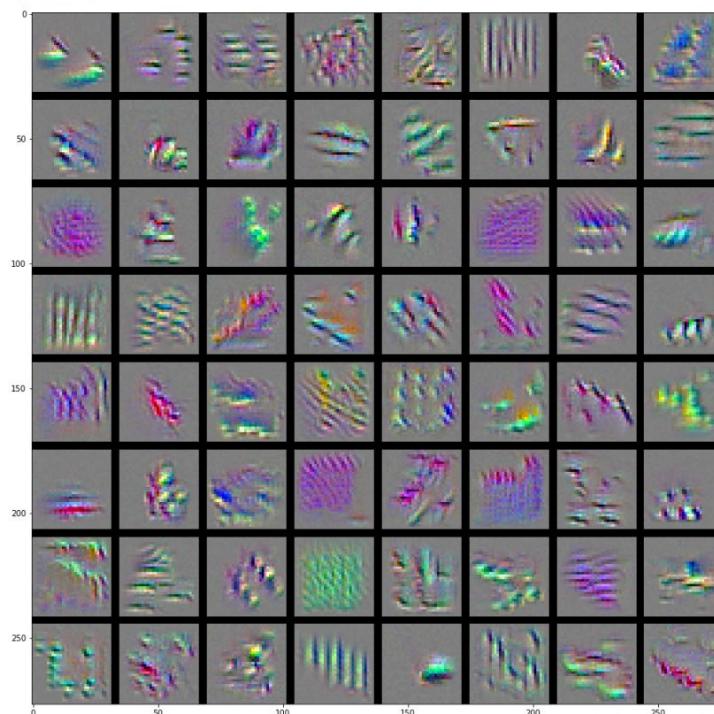
# Klasyfikacja wieloklasowa - model v9



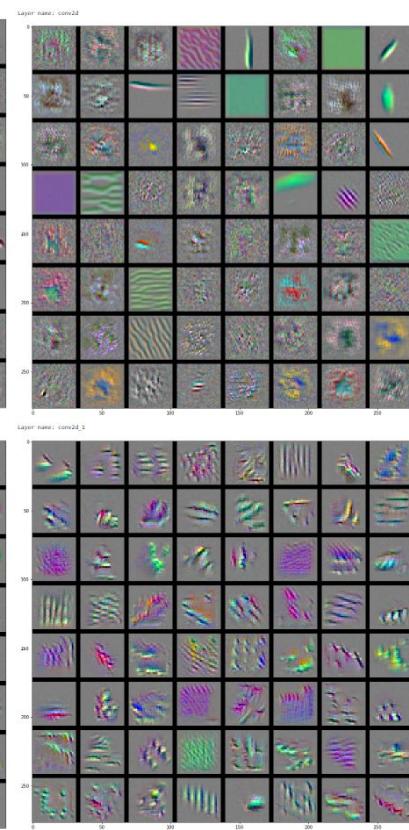
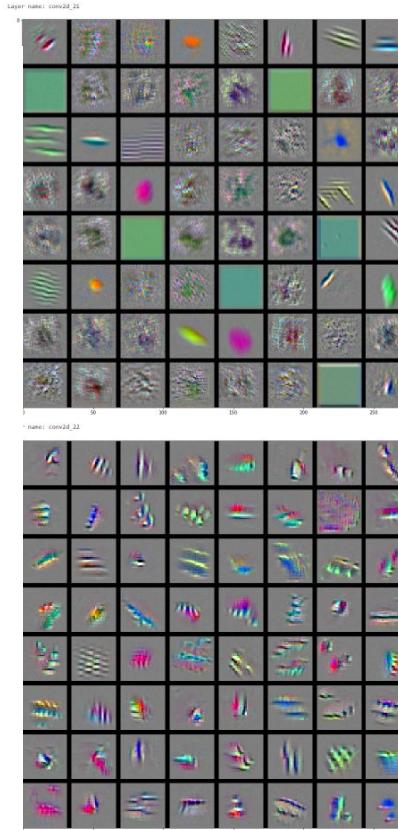
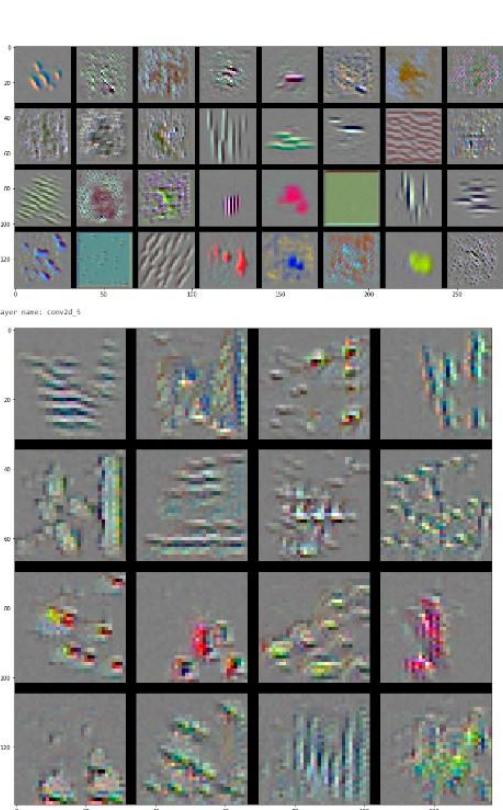
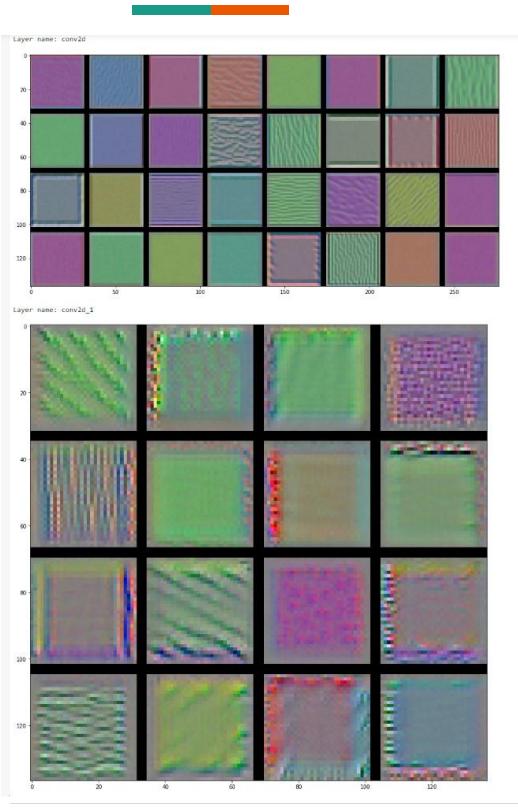
Layer name: conv2d



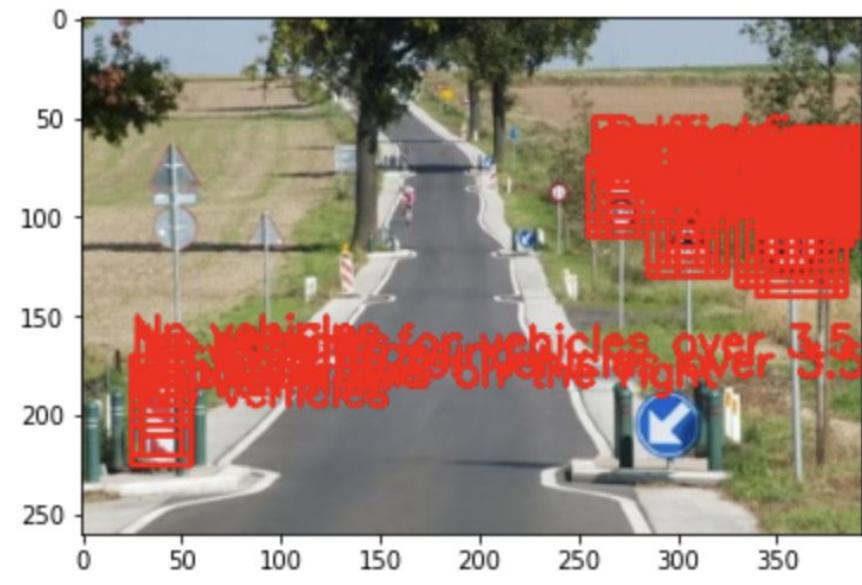
Layer name: conv2d\_1



# Problemy z uczeniem



# Detekcja z algorytmem likwidacji kolizji



- Łączny czas predykcji to 362s
  - Czas przetwarzania: 0.68s