
Klasyfikacja sportów poprzez zdjęcia

Tymon Jasiński 2P2

Cel

Znalezienie najlepszego klasyfikatora i jego konfiguracji do predykcji danych. Kryteriami dobrego klasyfikatora będą:

- Wysoka dokładność,
- Niska strata.

Baza danych

Jest to kolekcja 14 000 zdjęć obejmujących 100 różnych dyscyplin sportowych, podzielona na część do trenowania (13 500 zdjęć) i testowania (500 zdjęć). Zdjęcia mają format jpg w rozdzielczości 224x224x3 (RGB). Baza posiada nieliczne błędy jak np. Wstawiony folder zamiast zdjęcia lub zdjęcie niezwiązane ze sportem.



Preprocessing – KNN i MLP

Nie każdy klasyfikator przyjmuje dane w ten sam sposób. Oto techniki jakie zostały użyte do przetworzenia danych dla klasyfikatorów KNN i MLP:

- Resizing – aby zapewnić jednolity rozmiar skalujemy wszystkie obrazy do tego samego rozmiaru
- Normalizacja – skaluje wartości RGB z 0-255 na 0-1
- Spłaszczanie - modele te nie potrafią interpretować przestrzennej struktury obrazu, więc dane są spłaszczane

Preprocessing – CNN

Klasyfikator CNN z biblioteki tensorflow oferuje wiele gotowych narzędzi do preprocessingu obrazów. Oto jakie narzędzia zostały użyte:

- Normalizacja
- Augmentacja danych (tylko zbiór treningowy) - seria losowych transformacji, które tworzą zmodyfikowane wersje obrazów w celu uniknięcia przeuczenia, a w tym:
 - losowy obrót,
 - losowe przybliżenie,
 - odbicie lustrzane,
 - losowe przesunięcie w pionie i poziomie.
- Puste piksele po augmentacji (np. Po obrocie) zostają wypełnione na podstawie sąsiadujących pikseli

Preprocessing – przykład augmentacji

Przed



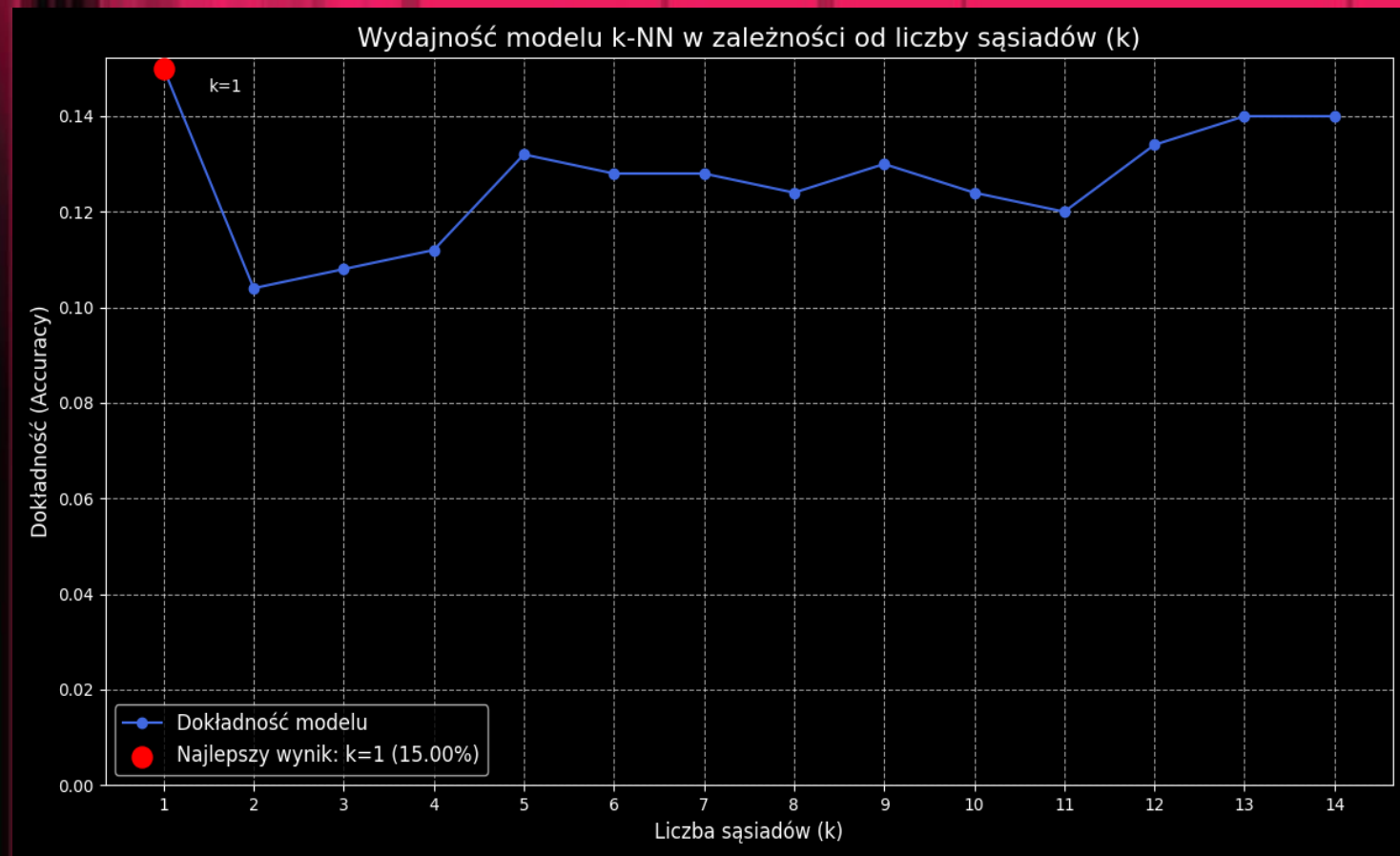
Po



Klasyfikacja KNN

KNN jest najszybszym, ale zarazem najmniej dokładnym klasyfikatorem z tego zestawienia.

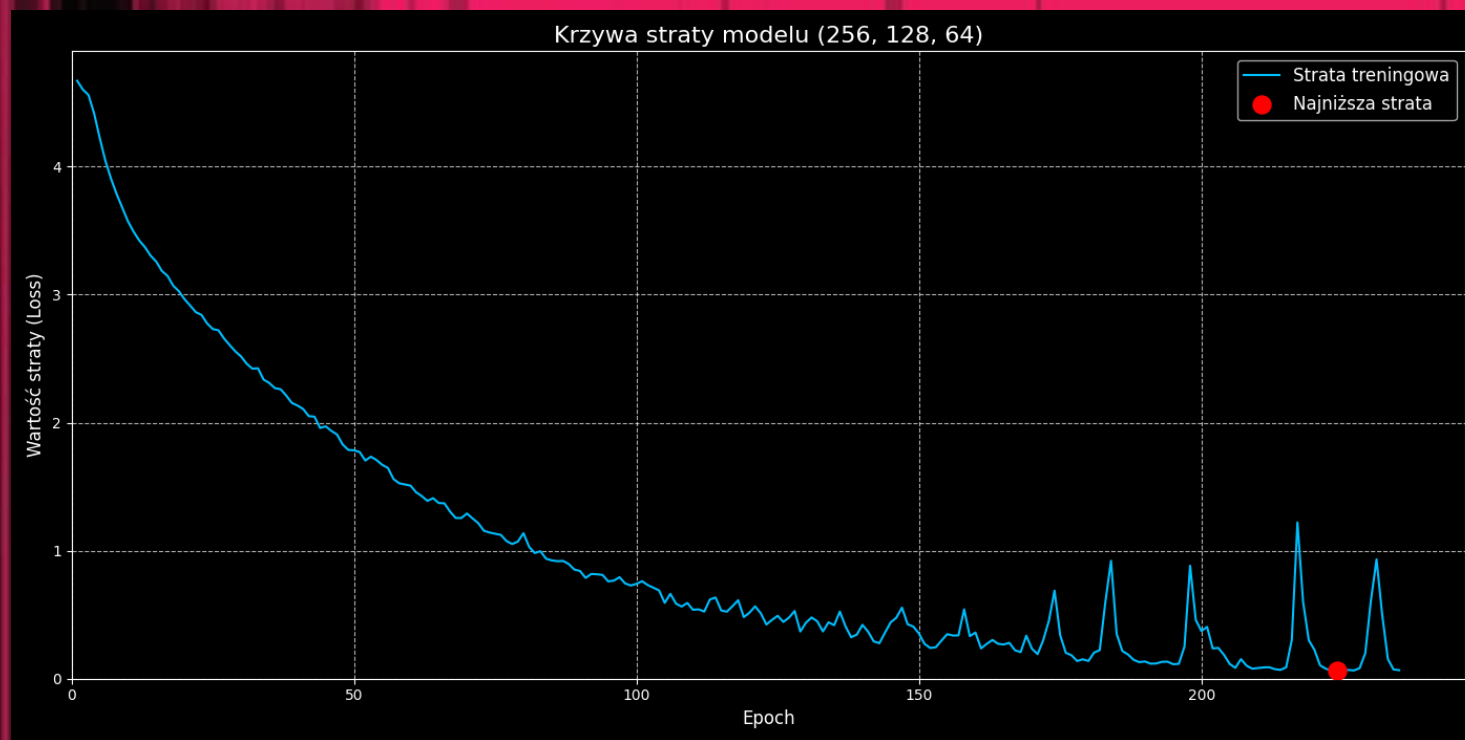
Wykonuje się on zaledwie kilka sekund, lecz traci na dokładności wynoszącej w najlepszym przypadku 15%.



Klasyfikacja MLP

Konfiguracje (ukryte warstwy):

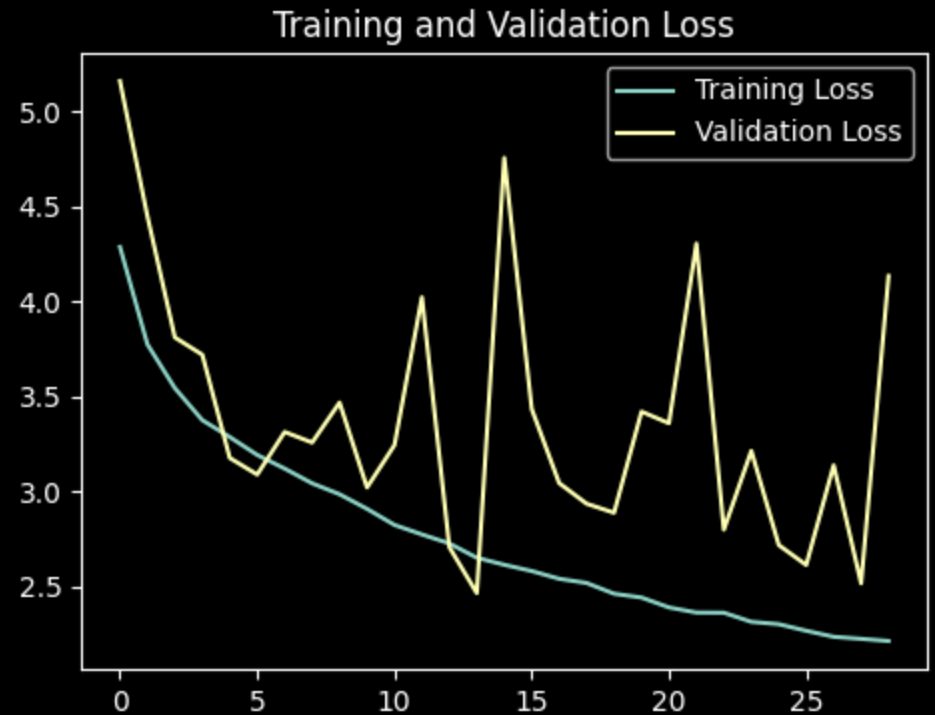
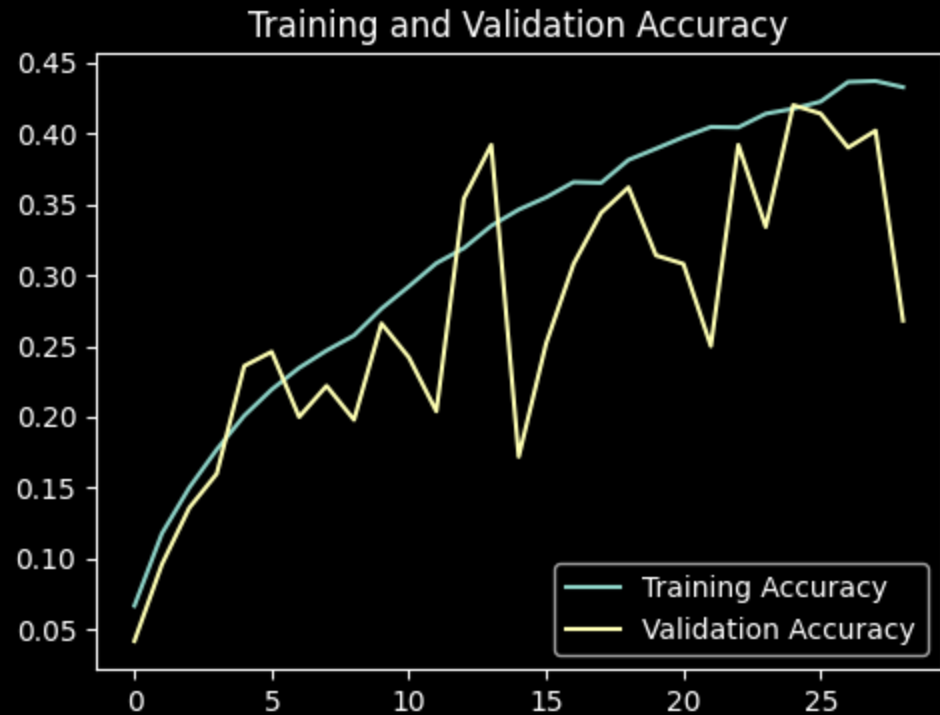
- 128 i 64
 - o Strata: ~0.6
 - o Dokładność: ~19%
 - o Czas: 15min
- 256, 128 i 64
 - o Strata: ~0.06
 - o Dokładność: ~21%
 - o Czas: 30min
- 128, 128 i 64
 - o Strata: ~0.4
 - o Dokładność: ~21%
 - o Czas: 25min



Klasyfikacja CNN – konfiguracja #1

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 32)	896
batch_normalization (BatchNormalization)	(None, 224, 224, 32)	128
activation (Activation)	(None, 224, 224, 32)	0
conv2d_1 (Conv2D)	(None, 224, 224, 32)	9,248
batch_normalization_1 (BatchNormalization)	(None, 224, 224, 32)	128
activation_1 (Activation)	(None, 224, 224, 32)	0
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
dropout (Dropout)	(None, 112, 112, 32)	0
conv2d_2 (Conv2D)	(None, 112, 112, 64)	18,496
batch_normalization_2 (BatchNormalization)	(None, 112, 112, 64)	256
activation_2 (Activation)	(None, 112, 112, 64)	0
conv2d_3 (Conv2D)	(None, 112, 112, 64)	36,928
batch_normalization_3 (BatchNormalization)	(None, 112, 112, 64)	256
activation_3 (Activation)	(None, 112, 112, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
dropout_1 (Dropout)	(None, 56, 56, 64)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 64)	0
dense (Dense)	(None, 512)	33,280
batch_normalization_4 (BatchNormalization)	(None, 512)	2,048
activation_4 (Activation)	(None, 512)	0
dropout_2 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 100)	51,300
Total params: 152,964 (597.52 KB)		
Trainable params: 151,556 (592.02 KB)		
Non-trainable params: 1,408 (5.50 KB)		

Klasyfikacja CNN – konfiguracja #1

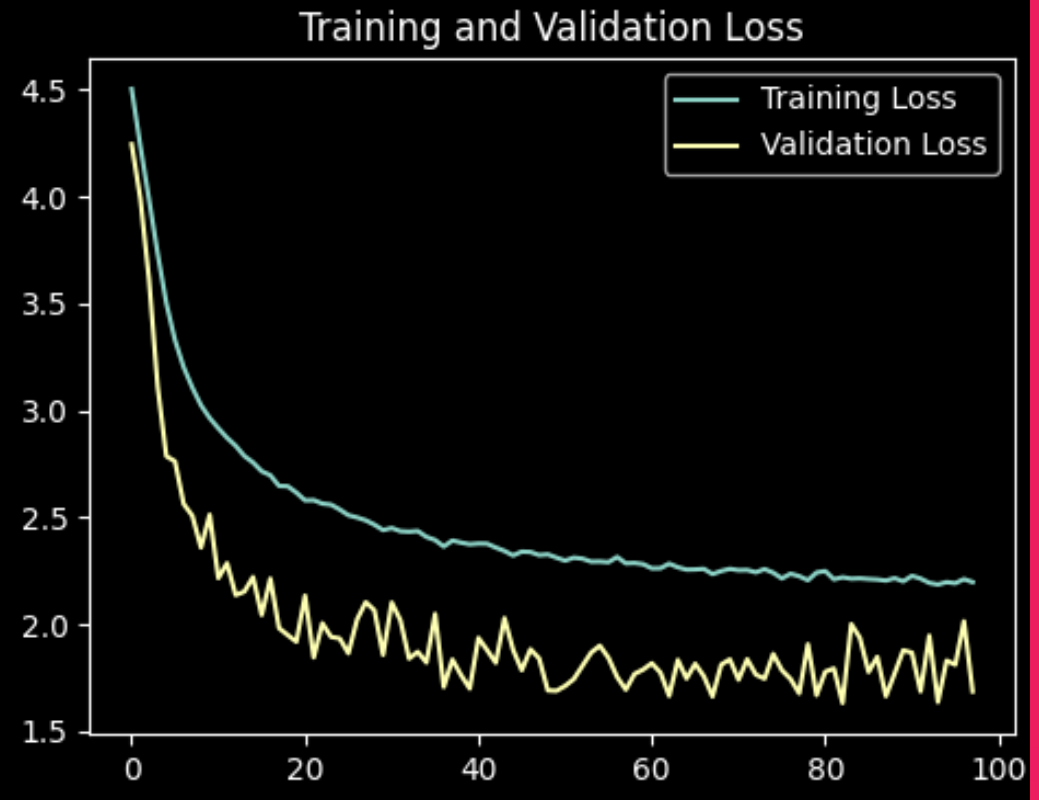
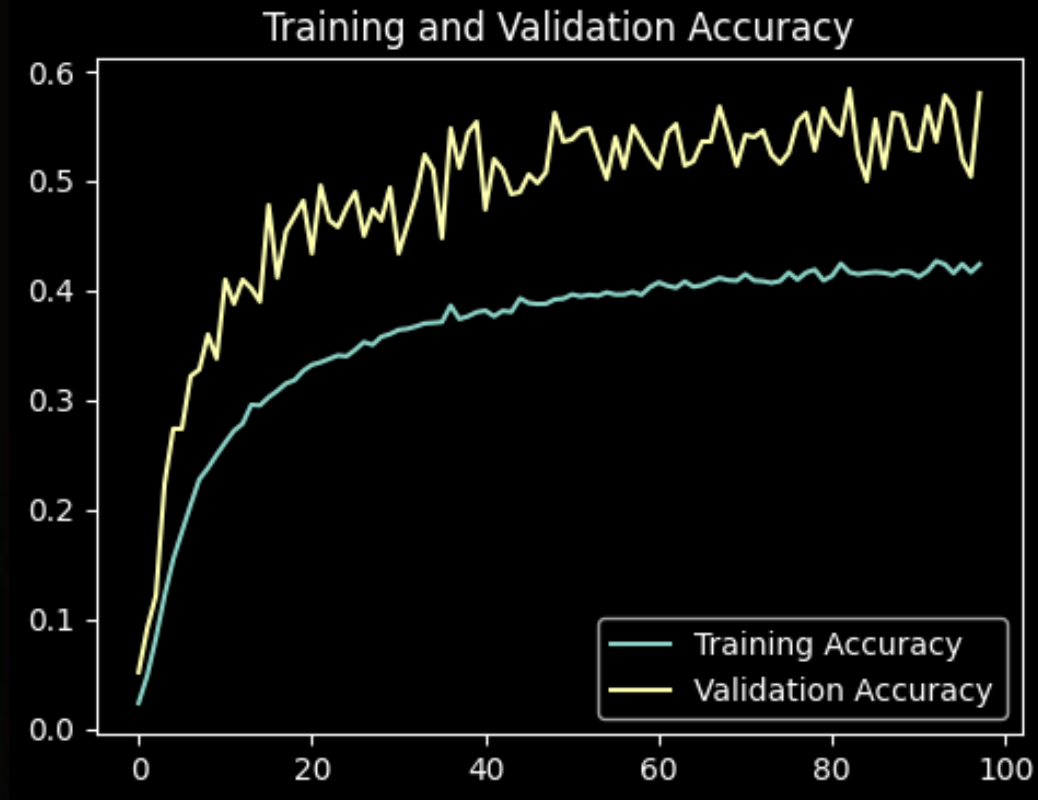


Klasyfikacja CNN – konfiguracja #2

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 512)	44,302,848
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 100)	51,300

Total params: 44,447,396 (169.55 MB)
Trainable params: 44,447,396 (169.55 MB)
Non-trainable params: 0 (0.00 B)

Klasyfikacja CNN – konfiguracja #2



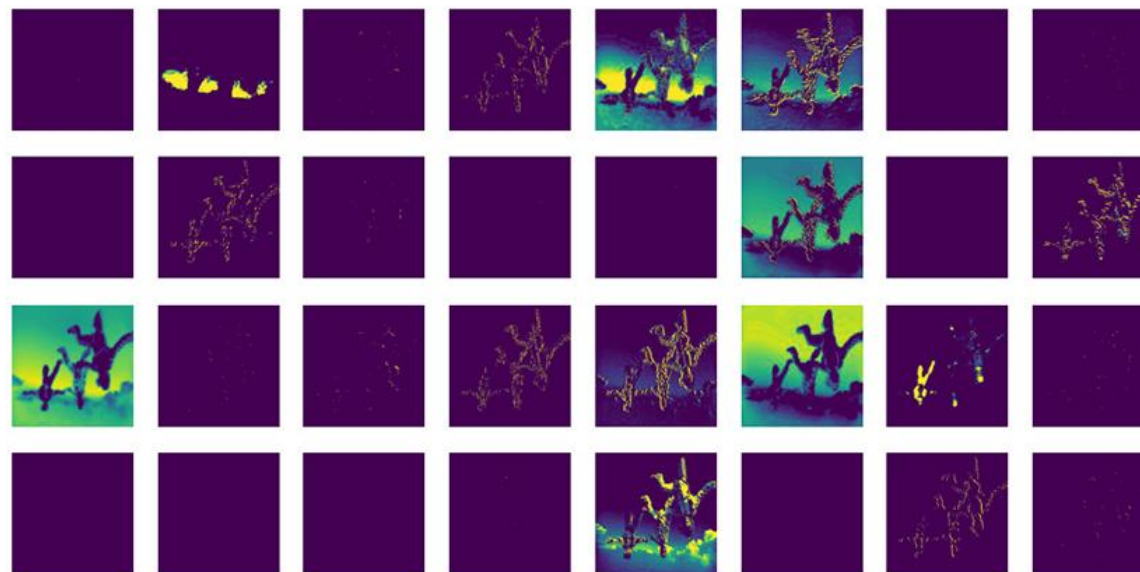
CNN – warstwa splotowa i łącząca



Conv2d

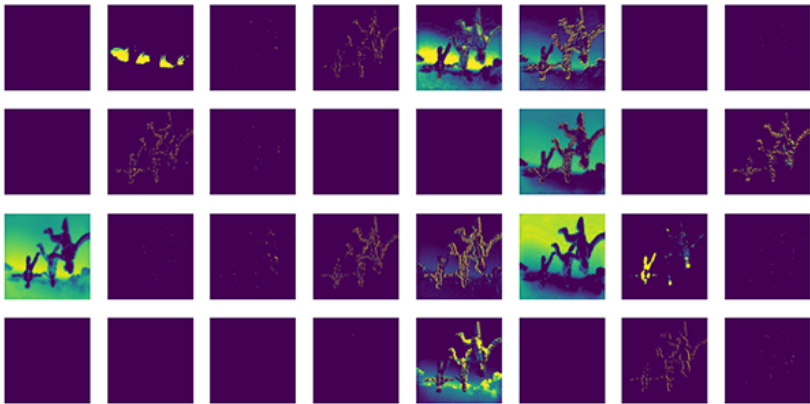


Wynik warstwy: conv2d
Kształt: (1, 222, 222, 32)



CNN – warstwa splotowa i łącząca

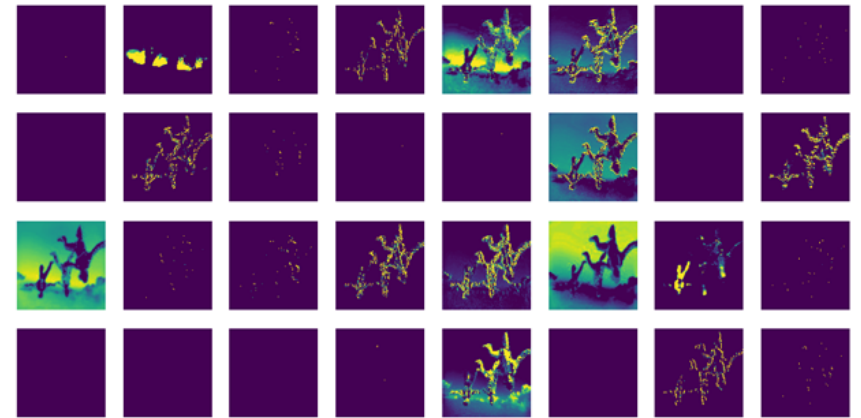
Wynik warstwy: conv2d
Kształt: (1, 222, 222, 32)



MaxPooling

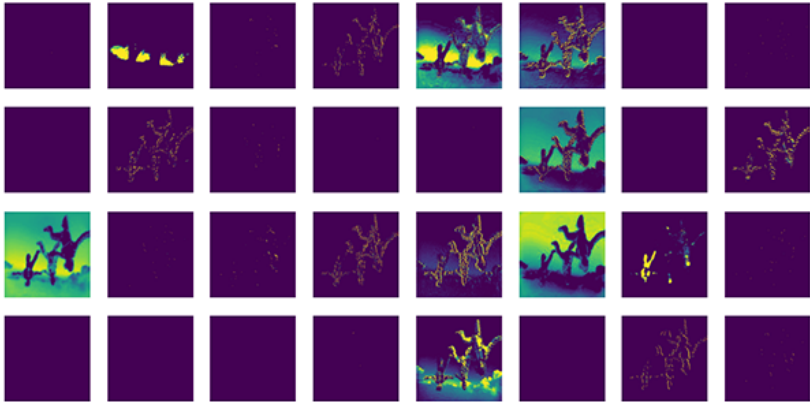


Wynik warstwy: max_pooling2d
Kształt: (1, 111, 111, 32)



Klasyfikacja CNN – przykład działania

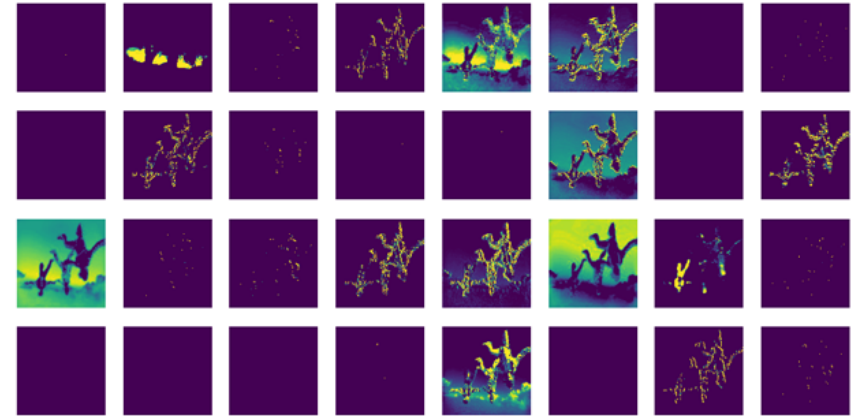
Wynik warstwy: conv2d
Kształt: (1, 222, 222, 32)



MaxPooling

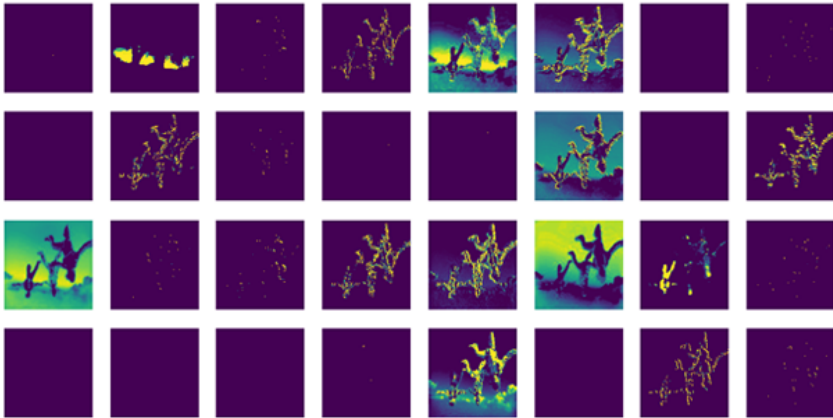


Wynik warstwy: max_pooling2d
Kształt: (1, 111, 111, 32)



CNN – warstwa splotowa i łącząca

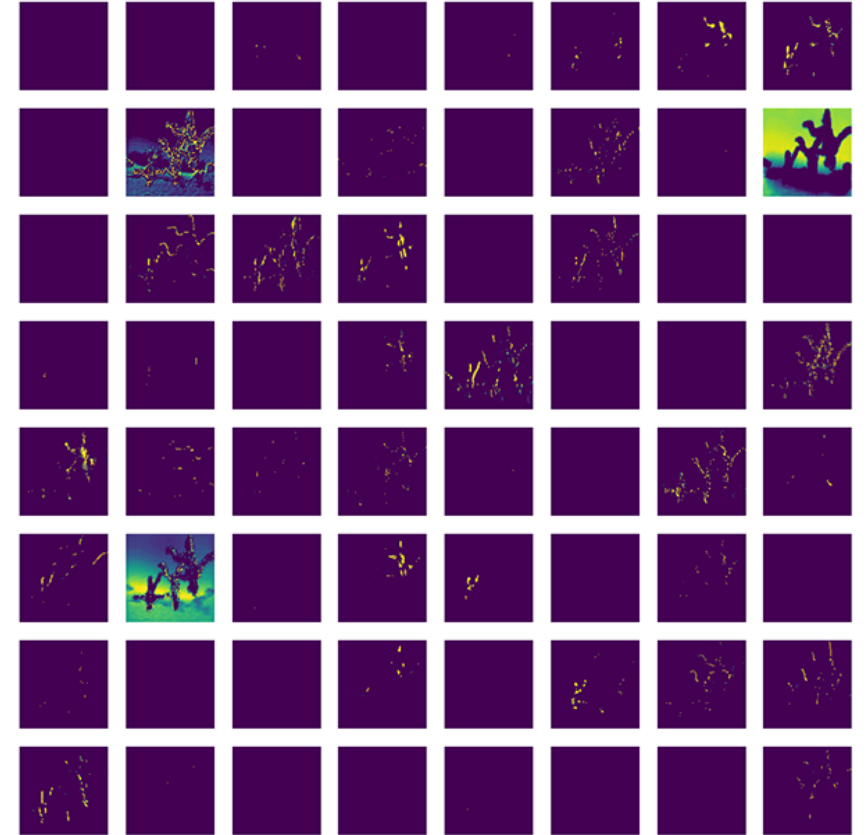
Wynik warstwy: max_pooling2d
Kształt: (1, 111, 111, 32)



Conv2d

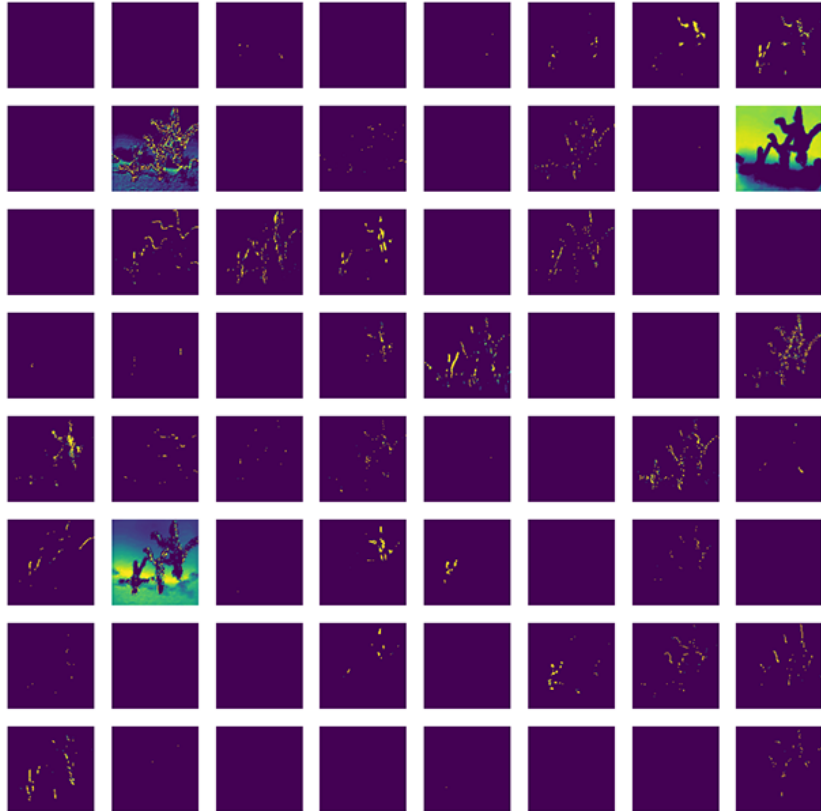


Wynik warstwy: conv2d_1
Kształt: (1, 109, 109, 64)



CNN – warstwa splotowa i łącząca

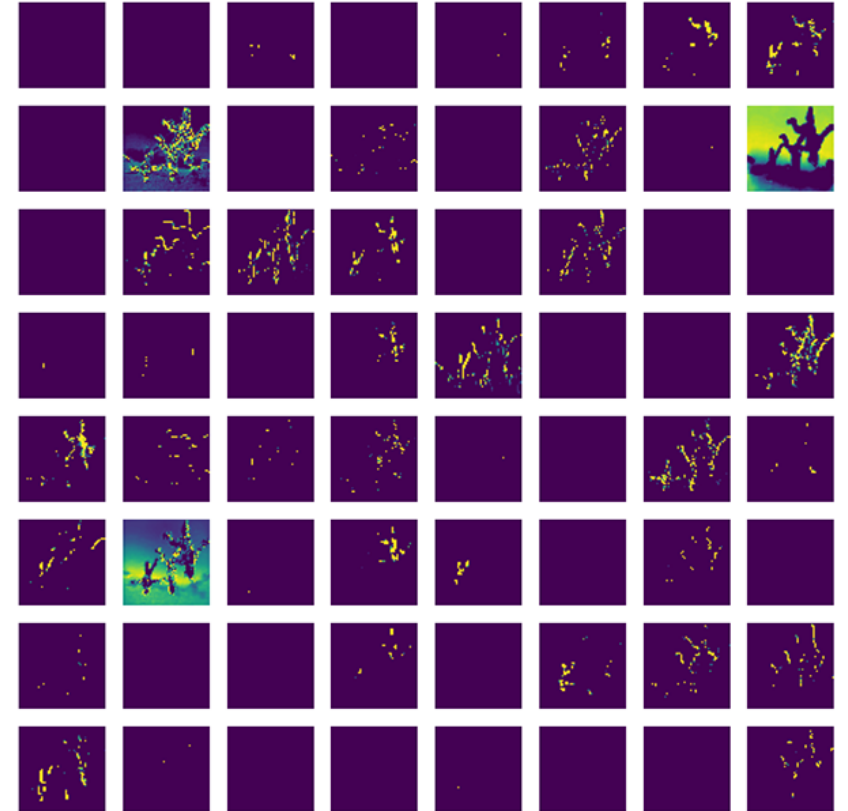
Wynik warstwy: conv2d_1
Kształt: (1, 109, 109, 64)



MaxPooling

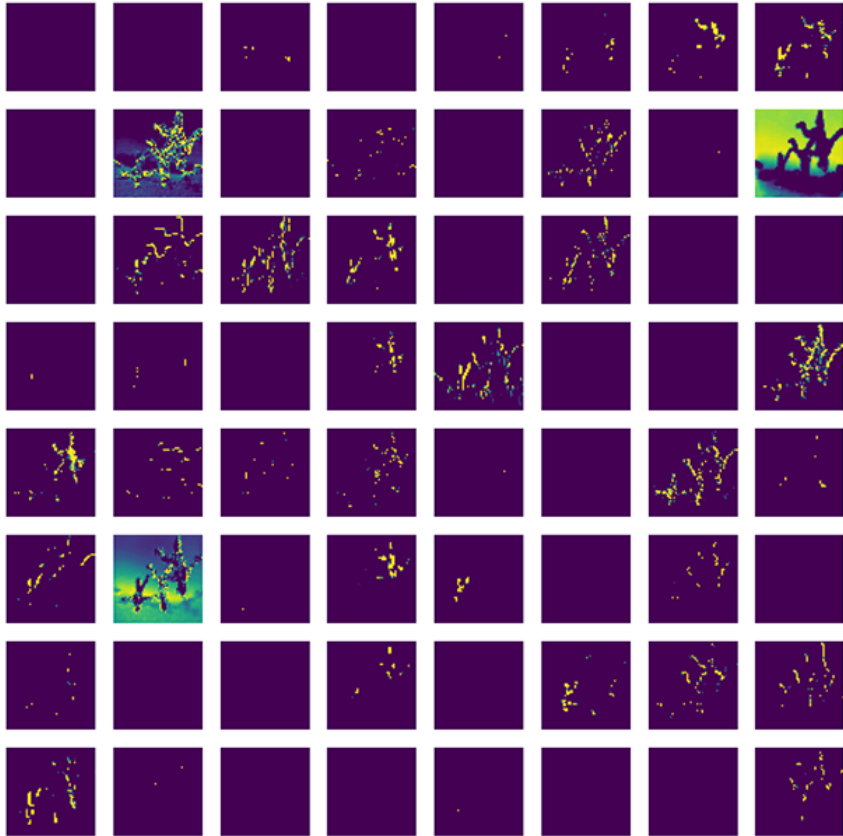


Wynik warstwy: max_pooling2d_1
Kształt: (1, 54, 54, 64)



CNN – warstwa splotowa i łącząca

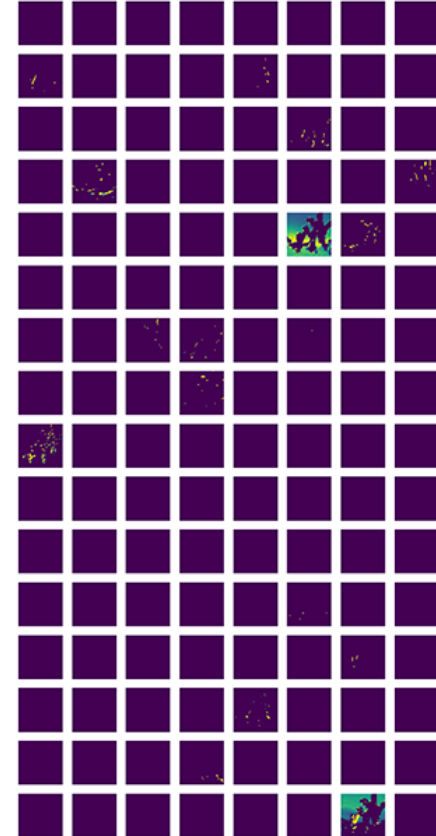
Wynik warstwy: max_pooling2d_1
Kształt: (1, 54, 54, 64)



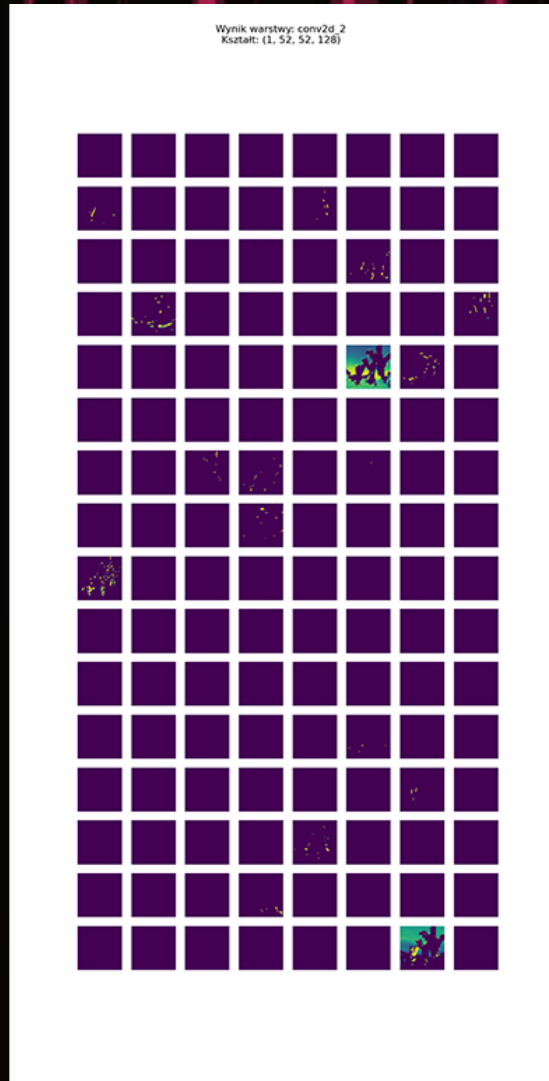
Conv2d



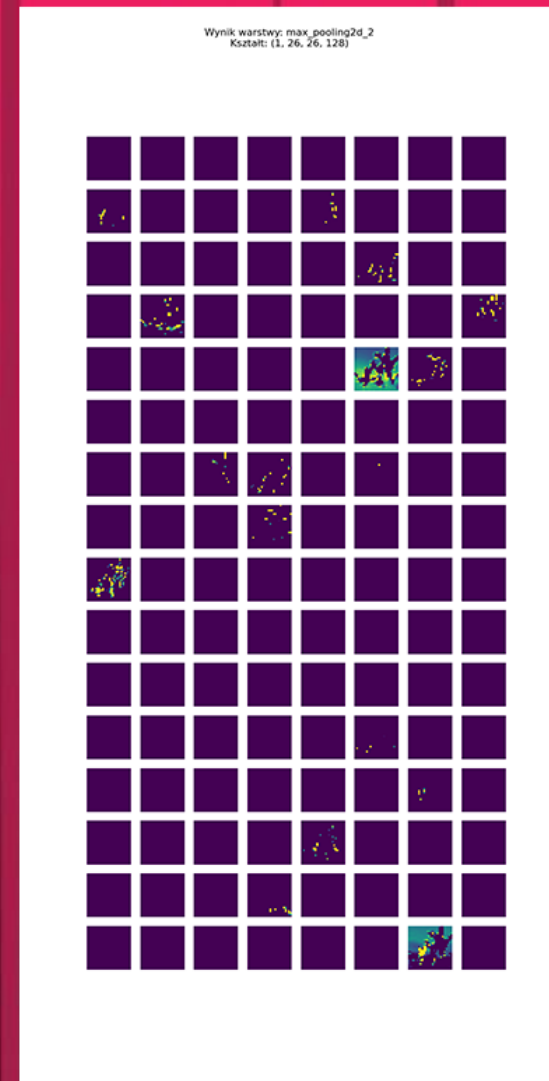
Wynik warstwy: conv2d_2
Kształt: (1, 52, 52, 128)



CNN – warstwa splotowa i łącząca



MaxPooling



Wnioski

Klasyfikator CNN zdecydowania sprawuje się najlepiej w kontekście klasyfikacji obrazów. W testach porównawczych jego skuteczność była nieporównywalnie wyższa od alternatywnych modeli, które miały trudności z poprawną klasyfikacją znaczącej części danych testowych.

Koniec