

## LAB 03 - WPROWADZENIE DO GŁĘBOKIEGO UCZENIA - ROZPOZNAWANIE TWARZY

**Karol Działowski**

nr albumu: 39259  
przedmiot: Uczenie Maszynowe 2

Szczecin, 21 stycznia 2021

### Spis treści

<b>1 Cel laboratorium</b>	<b>1</b>
<b>2 Zaproponowana sieć</b>	<b>2</b>
<b>3 Wpływ szumu na jakość klasyfikacji</b>	<b>3</b>
<b>4 Wpływ jasności na jakość klasyfikacji</b>	<b>3</b>
<b>5 Wnioski</b>	<b>4</b>

### 1 Cel laboratorium

Celem laboratorium było zapoznanie się z funkcjonalnością pakietów *TensorFlow* i *Keras*. W tym celu zaimplementowano prosty model sieci splotowej o zadaniu rozpoznawania twarzy.

Zadaniem głównym było zbudowanie modelu sieci splotowej o dowolnej architekturze, który ma służyć do rozpoznawania obrazów twarzy. Należało zbadać wpływ rozdzielczości obrazu wejściowego na jakość uczenia oraz przetestować model na obrazach zaszumionych i o zmienionym poziomie jasności oraz określić wpływ poziomu szumu i jasności na jakość klasyfikacji.

Skorzystano z zbioru danych Olivetti Faces, który zawiera 400 zdjęć i 40 unikalnych klas (osób). Rozmiar każdego obrazu to 112x92 pikseli.

## 2 Zaproponowana sieć

Zaproponowana sieć opiera się na poprzednio zaimplementowanych architekturach. Wykorzystano następującą strukturę:

- sieć splotowa (1) o głębokości 32 z rozmiarem kernela  $3 \times 3$
- warstwa MaxPooling o rozmiarze  $2 \times 2$
- sieć splotowa (2) o głębokości 64 z rozmiarem kernela  $3 \times 3$
- warstwa MaxPooling o rozmiarze  $2 \times 2$
- sieć splotowa (3) o głębokości 128 z rozmiarem kernela  $3 \times 3$
- warstwa MaxPooling o rozmiarze  $2 \times 2$
- Warstwa gęsta o 256 neuronach
- Warstwa BatchNormalization
- Warstwa Dropout odrzucająca 25% połączeń.
- Warstwa gęsta o 40 neuronach z aktywacją softmax

### Kod źródłowy 1: Implementacja modelu

Źródło: Opracowanie własne

```
1 def create_model(img_shape, filters=(32, 64, 128)):
2     inputShape = (img_shape[0], img_shape[1], 1)
3
4     chanDim = -1
5
6     inputs = Input(shape=inputShape)
7
8     for (i, f) in enumerate(filters):
9
10        if i == 0:
11            x = inputs
12            # CONV => RELU => BN => POOL
13            x = Conv2D(f, (3, 3), padding="same")(x)
14            x = Activation("relu")(x)
15            x = MaxPooling2D(pool_size=(2, 2))(x)
16
17
18        x = Flatten()(x)
19        x = Dense(256)(x)
20        x = Activation("relu")(x)
21        x = BatchNormalization(axis=chanDim)(x)
22        x = Dropout(0.25)(x)
23        x = Dense(40)(x)
```

```

24     x = Activation("softmax")(x)
25
26     model = Model(inputs, x)
27
28     opt = SGD(lr=1e-3, momentum=1e-3/200)
29
30     model.compile(optimizer=opt, loss="categorical_crossentropy",
31                  metrics=["accuracy"])
32
33     return model

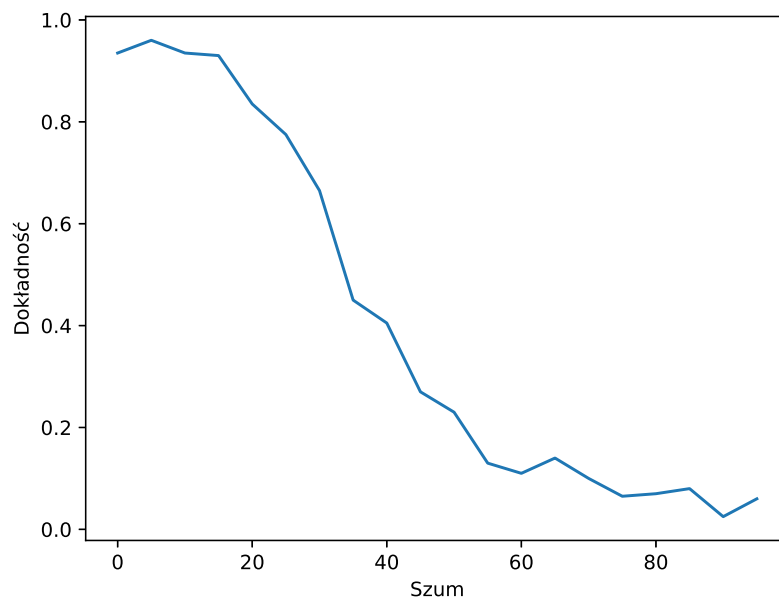
```

Model nauczone na 1000 epokach z rozmiarem batch 128, krokiem uczenia 0.001 oraz z optyimizatorem *SGD*.

Uzyskano dokładność na zbiorze testowym wynoszącą 93%.

### 3 Wpływ szumu na jakość klasyfikacji

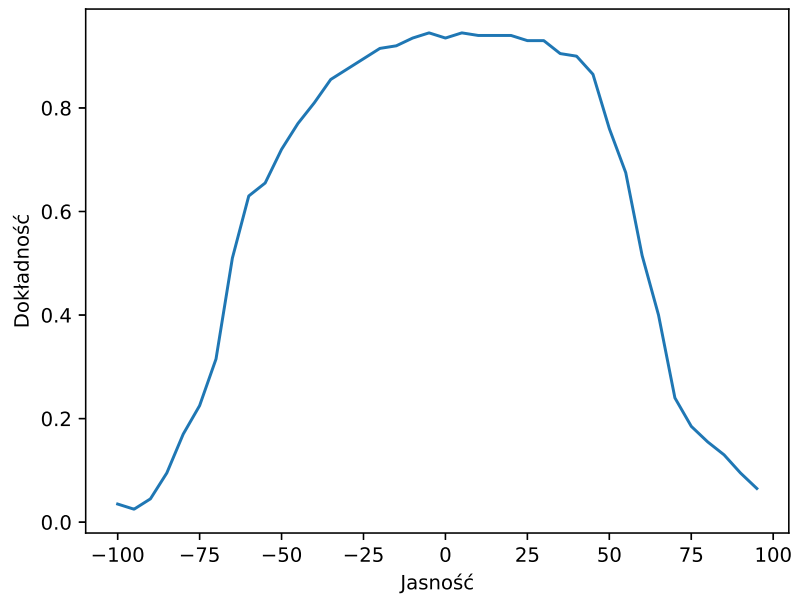
Przetestowano wpływ szumu na jakość klasyfikacji. Wykonano testy na zbiorze testowym dodając szum o losowych wartościach od 0 do 100. Wynik zaprezentowano na poniższym wykresie zależności szumu od dokładności.



**Rysunek 1:** Zależność dokładności od szumu

### 4 Wpływ jasności na jakość klasyfikacji

Przetestowano wpływ zmiany jasności na jakość klasyfikacji. Wykonano testy na zbiorze testowym dodając wartości z przedziału od -100 (przyciemnienie) do 100 (dodanie jasności). Wyniki zaprezentowano na poniższym wykresie zależności jasności od dokładności.



**Rysunek 2:** Zależność dokładności od zmiany jasności

## 5 Wnioski

Zbudowano model, którego dokładność klasyfikacji dla zbioru Olivetti Faces wynosi 93%. Przetestowano wpływ szumu i jasności na jakość klasyfikacji.

Dodając szum obserwujemy szybki spadek dokładności modelu. Największe zmiany dokładności występują w zakresie zmian od 20 do 55 szumu.

Zwiększając lub zmniejszając jasność również obserwujemy spadek dokładności. Zależność dokładności od zmiany jasności nie jest symetryczna. Dodając jasność dłużej obserwujemy zadowalającą dokładność, w przeciwieństwie do przyciemniania obrazu, gdzie spadek dokładności jest szybszy.