

# UMIR

Raport z projektu "zdjęcia dronów"

Zespół 8:

**Kazimierz Roman**

**Artur Romaniuk**

**Karol Duszczyk**

13 stycznia 2025

## Spis treści

<b>1</b>	<b>Dane</b>	<b>2</b>
1.1	Dataset . . . . .	2
1.2	Data augmentation . . . . .	2
<b>2</b>	<b>Zadanie 1: Uczenie klasyfikatora</b>	<b>2</b>
<b>3</b>	<b>Zadanie 2: Uczenie sieci głębokiej</b>	<b>2</b>
3.1	Przeprowadzić uczenie ostatniej warstwy splotowej wraz z częścią klasyfikującą .	2
3.2	Wytrenować całą sieć dla zadanych danych . . . . .	3
3.3	Uprościć strukturę sieci wytrenowanej w zadaniu 2c (np. poprzez usunięcie jednej lub więcej końcowych warstw splotowych, usunięcie warstw regularyzujących itp.) i ponowić uczenie . . . . .	4
3.4	Zanalizować wyniki 2 abc . . . . .	4
<b>4</b>	<b>Zadanie 3: Wizualizacja</b>	<b>4</b>
<b>5</b>	<b>Podsumowanie</b>	<b>4</b>

# 1 Dane

## 1.1 Dataset

W projekcie użyliśmy zbioru zdjęć *Drone Detection*. Zbiór ten zawiera 4 klasy: 0 - samolot, 1 - dron, 2 - helikopter, 3 - ptak.

## 1.2 Data augmentation

W każdym zadaniu zastosowano augmentację danych. Zastosowano następujące transformacje:

- Random Rotation - obrót o losowy kąt z zakresu  $(-54^\circ, +54^\circ)$  (factor=0.15)
- Random Translation - przesunięcie o losową wartość z zakresu  $(-10\%, +10\%)$  (height\_factor=0.1, width\_factor=0.1)
- Random Flip - losowe odbicie w poziomie lub pionie
- Random Contrast - losowa zmiana kontrastu z zakresu  $(-10\%, +10\%)$  (factor=0.1)

## 2 Zadanie 1: Uczenie klasyfikatora

## 3 Zadanie 2: Uczenie sieci głębokiej

### 3.1 Przeprowadzić uczenie ostatniej warstwy splotowej wraz z częścią klasyfikującą

Ostatnią warstwą splotową w sieci EfficientNetB0 jest warstwa top\_conv, która jest trzecią warstwą od góry. Z tego powodu zamrożono wszystkie warstwy sieci poza trzema ostatnimi. Jako wagi początkowe zastosowano wagi imagenet. Następnie przeprowadzono uczenie z wykorzystaniem zbioru treningowego. W trakcie uczenia zastosowano optymalizator Adam, funkcję straty sparse categorical crossentropy oraz metrykę accuracy. Przetestowano różne wartości współczynnika uczenia, ostatecznie wybrano wartość  $1e-5$ . Po 40 epokach uczenia osiągnięto na zbiorze testowym **accuracy na poziomie 0.74, loss na poziomie 0.66** oraz macierz pomyłek przedstawioną w tabeli 1. Wyniki uczenia w czasie przedstawiono na rysunku 1. Dalsze uczenie nie przynosiło poprawy wyników.

klasa	0	1	2	3
0	32	18	29	49
1	0	229	4	1
2	1	4	105	1
3	4	19	24	76

Tabela 1: Macierz pomyłek dla zadania 2a



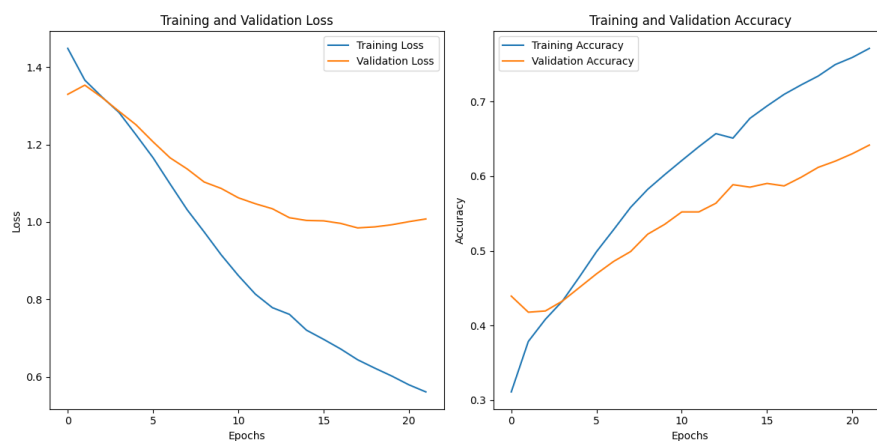
Rysunek 1: Wyniki uczenia dla zadania 2a

### 3.2 Wytrenować całą sieć dla zadanych danych

W tym zadaniu odmrożono wszystkie warstwy EfficientNetB0 i nadano im wagi początkowe imagenet. Następnie przeprowadzono uczenie z wykorzystaniem zbioru treningowego. W trakcie uczenia zastosowano optymalizator Adam, funkcję straty sparse categorical crossentropy oraz metrykę accuracy. Przetestowano różne wartości współczynnika uczenia, ostatecznie wybrano wartość  $1e-5$ . Podczas uczenia zastosowano mechanizm early stopping, który zatrzymywał uczenie jeśli przez 4 epoki nie następowała poprawa wyników. Uczenie zatrzymało się po 22 epokach. Ostatecznie osiągnięto na zbiorze testowym **accuracy na poziomie 0.63, loss na poziomie 1.01** oraz macierz pomyłek przedstawioną w tabeli 2. Wyniki uczenia w czasie przedstawiono na rysunku 2.

klasa	0	1	2	3
0	56	17	55	0
1	0	234	0	0
2	22	1	87	1
3	9	36	78	0

Tabela 2: Macierz pomyłek dla zadania 2b z wagami imagenet



Rysunek 2: Wyniki uczenia dla zadania 2b z wagami imagenet

Próbowano także wytrenować całą sieć bez zadanych wag początkowych, jednak w tym przypadku wyniki były dużo gorsze (accuracy na poziomie 0.38). Może to wynikać z faktu, że zbiór treningowy jest stosunkowo mały, a wagi początkowe z imagenet są lepsze niż losowe.

**3.3 Uprościć strukturę sieci wytrenowanej w zadaniu 2c (np. poprzez usunięcie jednej lub więcej końcowych warstw splotowych, usunięcie warstw regularyzujących itp.) i ponowić uczenie**

**3.4 Zanalizować wyniki 2 abc**

**4 Zadanie 3: Wizualizacja**

**5 Podsumowanie**