

# **Sprawozdanie**

Komunikacja Człowiek–Komputer

Marek Kamiński, 136732

Grzegorz Bryk, 136686

Klasyfikator wybranych środków transportu przedstawionych  
na zdjęciach oparty o sieć neuronową

## Spis treści

1 Wstęp.....	3
1.1 Cele ćwiczenia.....	3
1.2 Obrany temat.....	3
1.3 Sposób realizacji.....	3
1.3.1 Wykorzystane narzędzia.....	3
1.3.2 Zarys implementacji.....	3
1.3.3 Wykorzystany zbiór danych.....	4
1.4 Sposób oceny skuteczności.....	4
2 Realizacja.....	4
2.1 Wyniki.....	5
2.1.1 Macierze błędów.....	5
2.1.2 Wykresy przebiegu treningu.....	6
3 Wnioski.....	8
3.1 Popełnione błędy.....	8
3.2 Wyciągnięte wnioski.....	8

# 1 Wstęp

## 1.1 Cele ćwiczenia

Celem ćwiczenia było zdobycie i pogłębienie wiedzy o przetwarzaniu obrazów w systemach komputerowych.

## 1.2 Obrany temat

W ramach projektu postanowiliśmy zrealizować klasyfikator rozróżniający od siebie wybrane środki lokomocji, tj.:

- samoloty,
- samochody,
- łodzie motorowe,
- łodzie żaglowe,
- motocykle.

## 1.3 Sposób realizacji

Jako podstawę dla klasyfikatora postanowiliśmy wykorzystać sztuczną sieć neuronową.

### 1.3.1 Wykorzystane narzędzia

Do implementacji rozwiązania wykorzystaliśmy poniższe technologie:

- Python 3.7
- Keras
- Tensorflow z CUDA
- OpenCV

### 1.3.2 Zarys implementacji

Wykorzystaliśmy gotowy projekt sieci – był to zmniejszony wariant sieci VggNet16, przygotowany przez Adriana Rosebrocka. Sieć ta została przygotowana specjalnie do rozpoznawania obrazów. Ma 16 warstw, na co wskazuje nazwa. Korzysta z relatywnie małych filtrów konwolucyjnych (3x3), i daje dobre wyniki w zastosowaniach związanych z rozpoznawaniem obrazu.<sup>1</sup> Wadą tej sieci jest stosunkowo długi proces uczenia.

---

<sup>1</sup> <https://arxiv.org/abs/1409.1556>

### 1.3.3 Wykorzystany zbiór danych

Do treningu powyższej sieci potrzebowaliśmy zbioru zdjęć poszczególnych pojazdów. Uzyskaliśmy je wykorzystując API serwisu Flickr. Zbiór treningowy składał się z następującej liczby obrazów podzielonych wg kategorii:

- samochody: 829 zdjęć,
- motocykle: 801 zdjęć,
- łodzie motorowe: 619 zdjęć,
- łodzie żaglowe: 755 zdjęć,
- samoloty: 914 zdjęć.

Wszystkie pobrane obrazy zostały przekształcone do rozmiaru 96x96 pikseli, z trzeba ośmiobitowymi kanałami koloru.

Zbiór walidacyjny składał się z od 17 do 25 obrazów z każdej kategorii.

## 1.4 Sposób oceny skuteczności

Za wskaźnik oceny skuteczności sieci obraliśmy liczbę poprawnie sklasyfikowanych obrazów ze zbioru walidacyjnego.

## 2 Realizacja

Sieć uczyliśmy 3 razy. W ten sposób powstały 3 modele: AUG1000, AUG200 i AUG100 bez podejścia. Do porównania wykorzystaliśmy też model trenowany na mniejszym zbiorze – po 100 obrazków z każdej kategorii, który nazywamy Water. Liczby w nazwach modelu AUG oznaczają liczbę epok przez którą trenowano model. W uczeniu modeli AUG wykorzystaliśmy wspomaganie uczenia poprzez rotowanie obrazków do 25° w każdą stronę na wejściu sieci.

## 2.1 Wyniki

### 2.1.1 Macierze błędów

RANKING	car	motorbike	motorboat	plane	sailboat
car	17	0	2	1	0
motorbike	0	21	0	0	0
motorboat	1	0	15	4	0
plane	0	0	0	25	0
sailboat	0	0	0	3	14

Tabela 1: Macierz błędów modelu Water, w wierszach obiekt faktycznie przedstawiony na obrazie, w kolumnach wynik klasyfikacji

Dla modelu Water wykonaliśmy macierz błędów. Okazało się, że najczęstszą pomyłką sieci jest klasyfikowanie łodzi motorowej jako samolotu. Postanowiliśmy zastosować proces uczenia wspomaganego (stąd nazwa **augmented**), co poprawiło wyniki. Poniżej prezentujemy analogiczne macierze dla modeli AUG200 i AUG1000:

RANKING	car	motorbike	motorboat	plane	sailboat
car	20	0	0	0	0
motorbike	0	21	0	0	0
motorboat	1	0	18	0	1
plane	2	2	2	19	0
sailboat	0	0	1	0	16

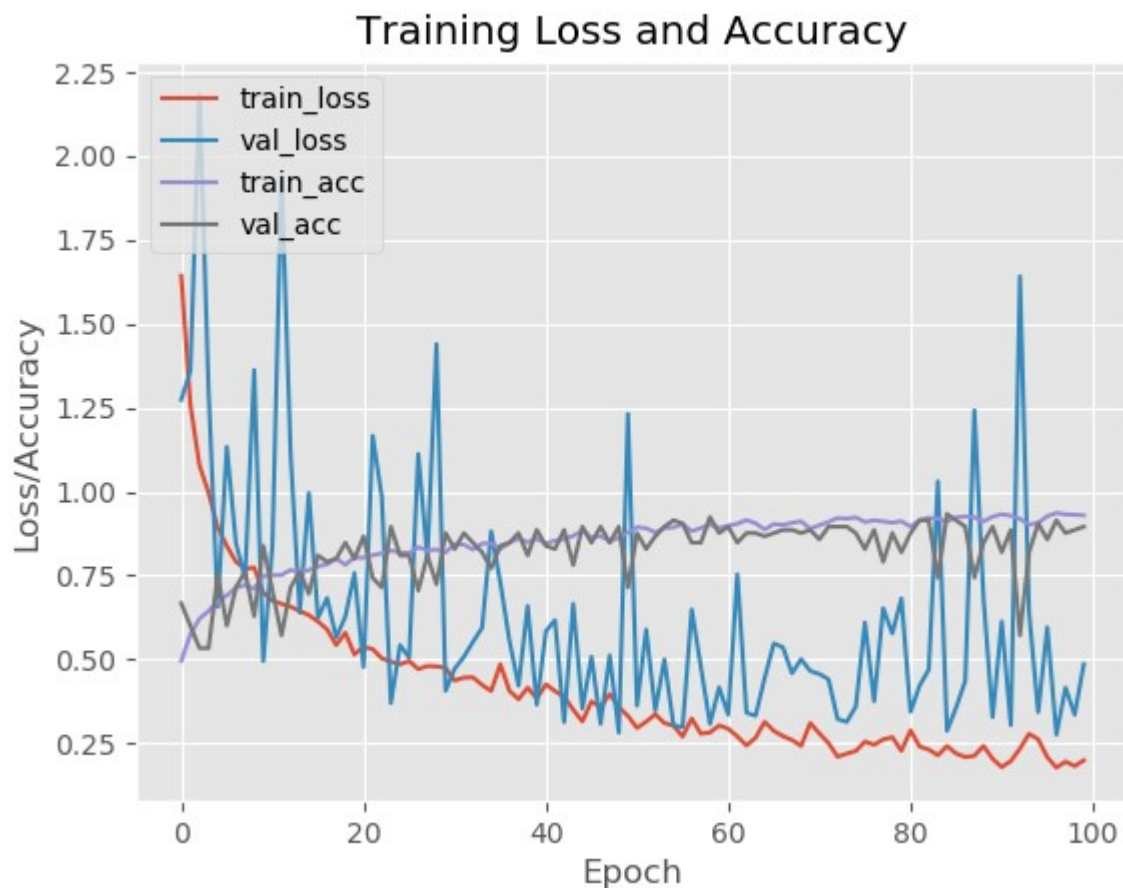
Tabela 2: Macierz błędów dla AUG200

RANKING	car	motorbike	motorboat	plane	sailboat
car	19	1	0	0	0
motorbike	0	21	0	0	0
motorboat	1	0	19	0	0
plane	0	3	2	19	1
sailboat	0	0	1	0	16

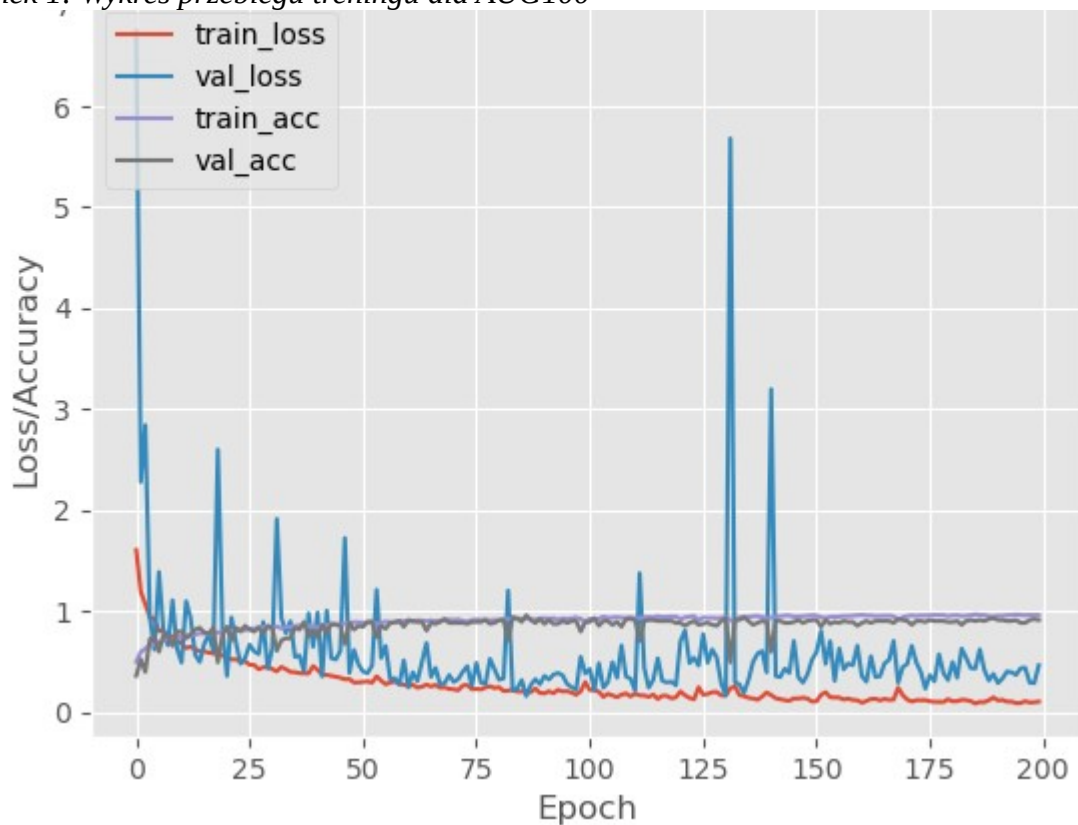
Tabela 3: Macierz błędów dla AUG1000

Jak widać, wyniki są zbliżone, z niewielką przewagą modelu AUG1000 nad krócej trenowanym konkurentem.

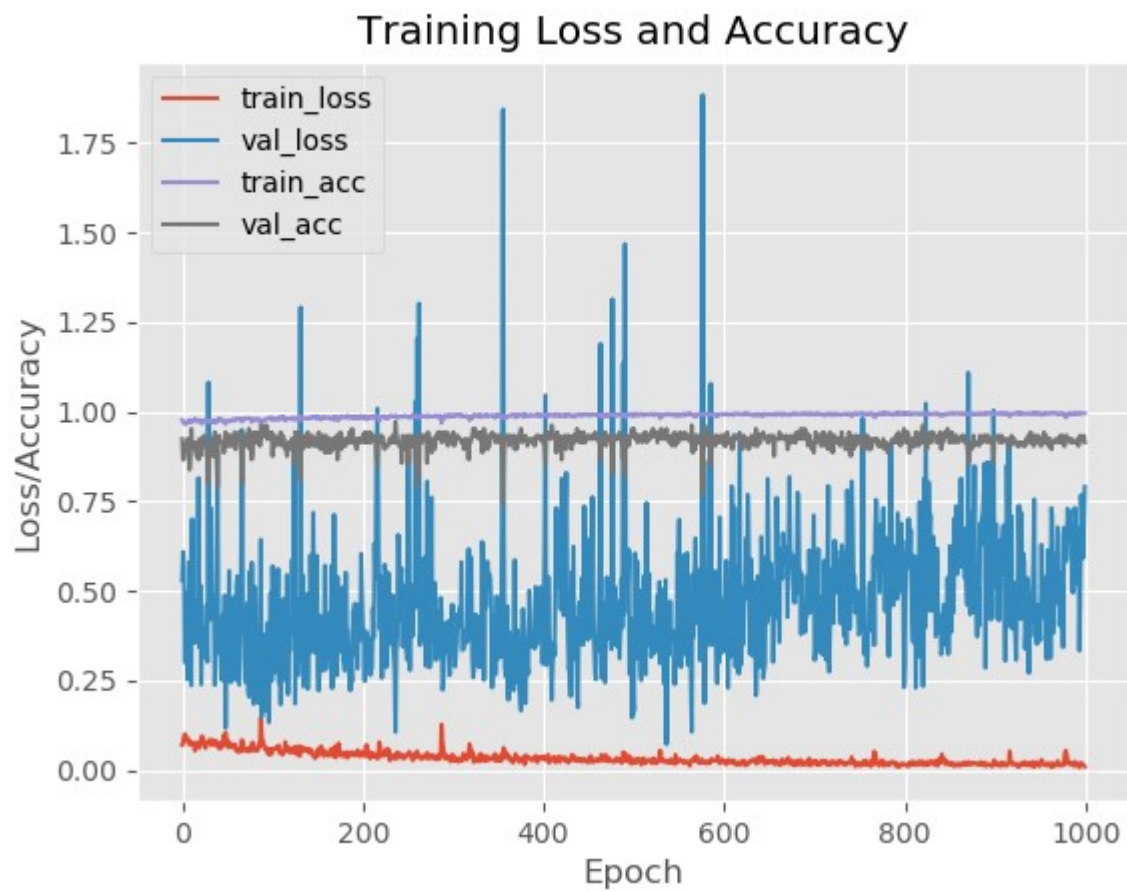
## 2.1.2 Wykresy przebiegu treningu



Rysunek 1: Wykres przebiegu treningu dla AUG100



Rysunek 2: Wykres przebiegu treningu dla AUG200



Rysunek 3: Wykres przebiegu treningu dla AUG1000

## **3 Wnioski**

### **3.1 Popelnione błędy**

- Dobraliśmy zbyt mały zbiór walidacyjny, co uniemożliwiło dokładną i poprawną ocenę działania sieci.
- W zbiorze treningowym znalazły się zdjęcia które nie przedstawiały całości obiektów, a np. widok skrzydła z wewnątrz samolotu. To mogło zaburzyć proces uczenia się sieci.
- Zbyt późno zaimplementowaliśmy etykietowanie, co uniemożliwiło ocenę działania sieci na początkowym etapie prac.

### **3.2 Wyciągnięte wnioski**

- Za pomocą sieci neuronowych w stosunkowo łatwy i szybki sposób można zrealizować zadanie rozpoznawania obiektów na obrazach.
- Sieci neuronowe nadają się do zastosowań w których nie jest wymagana 100-procentowa precyzja.
- Sieci neuronowe bardzo dobrze radzą sobie z typowymi przypadkami problemów do których się je aplikuje, ale przypadki brzegowe sprawiają im problemy.



## 4 Śmieszny obrazek na koniec:



Output

**motorboat 99.99%**

