

# Identifikacija ušes s pomočjo konvolucijskih nevronske mreže (CNN) z TensorFlow in Keras

Assignment #3

Image Based Biometrics 2020/21, Faculty of Computer and Information Science, University of Ljubljana

Nejc Bevk

**Abstract**—Implementacija preprostega modela za prepoznavanje ušes s pomočjo konvolucijskih nevronske mreže (CNN) z ogrodjema TensorFlow in Keras. Model je dostopen na Github repozitoriju: [https://github.com/nebev/ear\\_recognition](https://github.com/nebev/ear_recognition)

## I. UVOD

Sistemi za identifikacijo ušes po razvoju sledijo ostalim panogam računalniškega vida in za zajemanje značilik uporabljajo vedno naprednejše metode, kot je na primer uporaba konvolucijskih nevronske mreže. Za spopadanje s tovrstno problematiko na spletu zasledimo nekaj zanimivih rešitev [1], [2]. V omenjenih primer za identifikacijo ušes s pomočjo CNN uporabljajo platformo TensorFlow in visoko nivojsko ogrodje Keras, ki nam omogoča hitro implementacijo rešitev za reševanje problemov strojnega učenja s poudarkom na uporabi modernih metod globokega učenja.

## II. METODOLOGIJA

Za učenje modela sem uporabil podatkovno zbirko slik ušes AWE [3], [4]. Podatkovna zbirka vsebuje 1000 anotiranih slik ušes, ki pripadajo 100 različnim osebam. Odločil sem se pripraviti model za klasifikacijo oseb po slikah ušes po predlogih iz [1], [2]. Pri implementaciji sem sledil korakom dokumentacije TensorFlow.

Za začetek sem podatkovno bazo razdelil v zbirki za validacijo modela in učenje nevronske mreže. Delež slik se deli na 20% za validacijo in 80% za učenje. Med deljenjem podatkovne baze se opravi tudi preprosta obdelava slik, ki vse slike ušes raztegne na nastavljeni velikost. Za optimizacijo sem uporabil TensorFlow funkcijo za nalaganje podatkovne baze vnaprej `tf.data.AUTOTUNE`. Kljub težavam z namestitvijo vseh pravih komponent, sem postopek učenja pohitрил z uporabo procesorja grafične kartice.

Pri sestavljanju modela sem prvih nekaj slojev namenil obdelovanju slik. To nam omogočajo Keras sloji za pred-procesiranje. S tem se povečamo zbirko slik, ki so namenjene učenju in izboljšamo natančnost modela. Uporabil sem obdelave kot so: naključna rotacija, naključno približevanje naključen kontrast in naključne preslikave.

Sloje prvega modela sem sestavil po predlogu [1], kjer so uporabili dva konvolucijska sloja, dva "pooling" sloja in en polno povezan sloj.

Drugi model sem preizkusil po predlogu [2], kjer za klasifikacijo ušes uporabijo že vnaprej naučeno mrežo Xception. Pri drugem modelu, sem zaradi razlik pri implementaciji ponovno uvozil podatkovna seta, ampak na nekoliko drugačen način kot pri prvem modelu. Z drugo implementacijo sem imel nekaj težav in nisem dobil korektnih rezultatov.

## III. REZULTATI

Pri bom preizkušal različne nastavitve ob želji pridobiti čim večjo natančnost (Accuracy) za klasifikacijo oseb po slikah ušes.

Table I  
REZULTATI TESTOV

	Osnovni model	Dodana augmentacija	Augmentacija in Dropout sloji
Avg. test loss	8.5	4.4	4.1
Avg. test accuracy	0.21	0.15	0.19

Pri prvi implementaciji sem najprej preizkusil nek osnovni model, ki pa sem ga nekoliko optimiziral. Kot omenjeno prej sem najprej dodal sloj za augmentacijo podatkovne baze z nekaj osnovnimi obdelavami slik. Dodatno sem za izboljšanje točnosti dodal dva "dropout" sloja in tako nekoliko izboljšal "overfitting". Razlika med osnovnim in optimiziranim modelom je prikazana na 1 in 2, kjer se po optimizaciji opazi veliko boljše prilaganje, ki pa še vedno ni zadovoljivo.

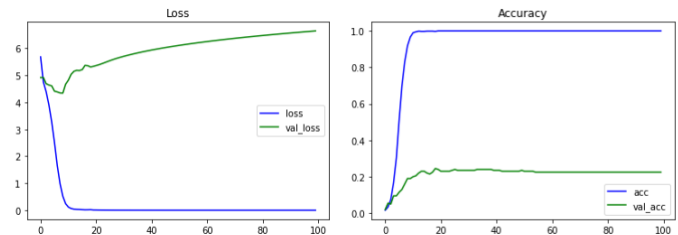


Figure 1. Graf točnosti in izgube v originalnem modelu

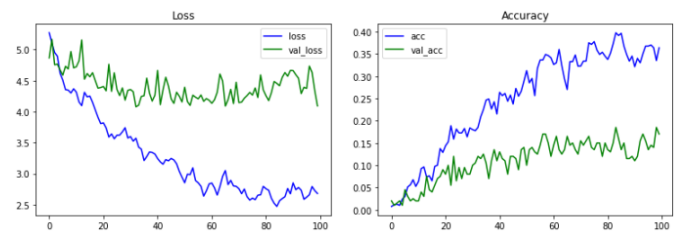


Figure 2. Graf točnosti in izgube v optimiziranem

Razlika med izgubo in točnostjo modela skozi nadgradnje je prikazana v 1. Povprečne vrednosti točnosti in izgube sem pridobil skozi 5 testov za posamezno nastavitve. V Učenje je v vsakem primeru šlo skozi 100 iteracij (epochs).

Pri implementaciji drugega modela z uporabo arhitekture Xception sem sledil implementaciji v [2]. Zaradi podobnosti podatkovnih setov sem pričakoval manj zapletov in boljše rezultate, ampak mi tega ni uspelo doseči. Primer rezultata natančnosti in izgube lahko vidimo v 3.

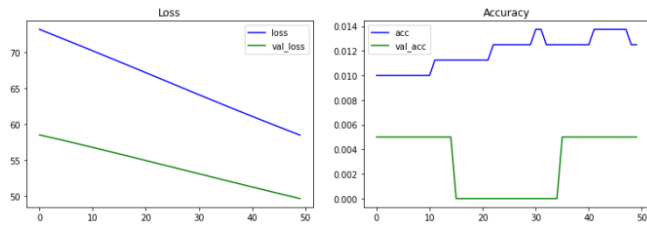


Figure 3. Graf točnosti in izgube v optimiziranem

#### IV. RAZPRAVA

Sama implementacija modela za učenje konvolucijskih nevronske mreže je z uporabo platforme TensorFlow in Keras je razmeroma preprosta in hitra. Kot prvi točko za doseganje boljših rezultatov bi izpostavil velikost podatkovnih setov za učenje, z večjim setom bi bilo učenje bolj temeljito in s tem klasifikacija bolj točna. Prepričan sem, da bi ta problem poleg agumentacije podatkovnega seta precej izboljšali tudi z uporabo naprej vnaprej naučenih značilk arhitekture Xception. Kot drugo točko pa bi izpostavil boljše razumevanje delovanja posameznih slojev nevronske mreže in njihovega vpliva na postopek učenja. Za boljše prilaganje bi moral uporabiti naprednejše metode namesto klasičnih, ki jih priporoča platforma.

#### REFERENCES

- [1] Z. Jiang, "Ear-recognition-with-keras," <https://github.com/TTTheo/Ear-Recognition-with-Keras>, 2019.
- [2] R. Balasubramanian, "Ear-recognition-challenge," <https://github.com/RamjiB/Ear-Recognition-Challenge>, 2019.
- [3] Ž. Emeršič, V. Štruc, and P. Peer, "Ear recognition: More than a survey," *Neurocomputing*, vol. 255, pp. 26–39, 2017.
- [4] Ž. Emeršič, B. Meden, P. Peer, and V. Štruc, "Evaluation and analysis of ear recognition models: performance, complexity and resource requirements," *Neural computing and applications*, pp. 1–16, 2018.