

PREPOZNAVANJE OSOBA U SLICI ZASNOVANO NA DUBOKOM UČENJU

Budak, Luka

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:321699>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-20**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**PREPOZNAVANJE OSOBA U SLICI
ZASNOVANO NA DUBOKOM UČENJU**

Završni rad

Luka Budak

Osijek, 2019

Sadržaj

1. UVOD	1
1.1 Zadatak završnog rada	1
2. DUBOKO UČENJE	2
2.1. Umjetna inteligencija, strojno učenje, duboko učenje	2
2.2 Primjena dubokog učenja u obradi digitalne slike	4
2.3 Primjena dubokog učenja za prepoznavanje osobe u digitalnoj slici	4
3. KONVOLUCIJSKE NEURONSKE MREŽE	6
3.1. Općeniti pojmovi	6
3.2 Načini optimizacije rada neuronskih mreža i mogući postupci za njihov bolji rad	9
4. MODEL ZA PREPOZNAVANJE OSOBE U SLICI	12
4.1 Opis baze podataka	12
4.2 Opis programskog okruženja	14
4.3 Odabir modela za prepoznavanje osoba	15
4.4 Postupak treniranja modela	16
5. VREDNOVANJE PERFORMANSI MODELA ZA PREPOZNAVANJE OSOBA U SLICI	23
6. ZAKLJUČAK	26
LITERATURA	27
SAŽETAK	29
ŽIVOTOPIS	31
PRILOZI	32

1. UVOD

Prepoznavanje osoba u slici je postalo neophodno u današnje vrijeme. Prepoznavanje osoba u slici ima raznoliku primjenu, od društvenih mreža, verifikacije identiteta, ciljanog marketinga pa do implementacije u zdravstvo i sigurnosne sustave.

Prepoznavanje osobe u slici je u svojoj osnovi uspoređivanje odabranih značajki lica sa slika lica koje se nalaze u bazi podataka. Budući da je u posljednjih desetak godina došlo do značajnog povećanja računalne moći s pojavom grafičkih kartica visokih performansi i sve većim javno dostupnim bazama slika, klasične tehnike i metode obrade mirne slike zamijenili su algoritmi zasnovani na strojnom učenju. Prije svega koriste se principi podgrupe strojnog učenja zvanog duboko učenje.

Arhitekture dubokog učenja, kao što je konvolucijska neuronska mreža, koriste mnogostruke slojeve kako bi postupno izdvojili bitne značajke iz neobrađenih podataka. Upotreba konvolucijske neuronske mreže uklanja potrebu za ručni biranjem bitnih značajki jer ih mreža sama nauči odabrati.

U ovom radu objašnjavaju se principi dubokog učenja i konvolucijskih mreža te se opisuje primjena prenesenog učenja na postojećem modelu za prepoznavanje osoba u slici. Rad se sastoji od šest poglavlja. U drugom poglavlju opisana su osnovna načela dubokog učenja uz nabrojane primjene dubokog učenja za obradu slike i primjene za prepoznavanje osoba u slici. Poglavlje broj tri objašnjava konvolucijske neuronske mreže i njene gradivne elemente kao i metode optimizacije mreže za bolji rad. U četvrtom poglavlju predstavlja se baza podataka slika korištenih za treniranje mreže, opisano je programsko okruženje, objašnjena je arhitektura već postojeće konvolucijske neuronske mreže te njeno treniranje. Peto poglavlje prikazuje rezultate dobivenog modela za prepoznavanje osoba u slici. Šesto, ujedno i posljednje poglavlje donosi zaključak.

1.1 Zadatak završnog rada

U radu potrebno je opisati probleme vezane uz prepoznavanje osoba u slici, dati sažeti pregled postojećih rješenja za ovaj problem i razviti vlastito rješenje koje prepoznaje osobe u slici koristeći duboko učenje. Rješenje je potrebno evaluirati na jednoj od javno dostupnih baza slika.

2. DUBOKO UČENJE

2.1. Umjetna inteligencija, strojno učenje, duboko učenje

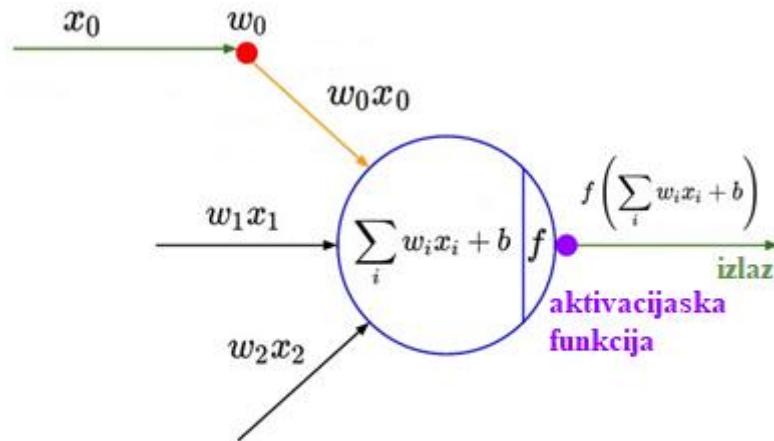
Umjetna inteligencija je pojam koji obuhvaća sve tehnike koje omogućuju računalima da oponašaju ljudska ponašanja kao što su učenje i rješavanje problema. Mogućnosti računala koje spadaju pod umjetnu inteligenciju podrazumijevaju uspješno razumijevanje ljudskog govora, autonomno upravljanje vozilima, natjecanje u visoko kompetitivnim strateškim igrama (šah, Go), robotika, računalni vid kao i mnogi drugi [1].

Strojno učenje je primjena umjetne inteligencije koja statističkim metodama omogućuje računalu automatsko učenje i napredak s iskustvom. To su algoritmi koji prave matematičke modele na uzorku podataka, tzv. trening podacima, kako bi radili predikcije ili donosili odluke bez da su eksplicitno isprogramirani za izvođenje tog zadatka. Cilj je dozvoliti računalu da ono samo uči te da bez ljudskog upliva usklađuje radnje [1].

S obzirom na pristup razlikuje se više tipova algoritama za učenje, nenadzirano, nadzirano i podržano strojno učenje. Algoritmi nenadziranog strojnog učenja uzimaju samo setove ulaznih podataka te traže skrivenu strukturu u tim podacima, npr. grupiranjem (engl. *grouping*) ili uskupljavanjem (engl. *clustering*) podatkovnih točaka. Algoritmi nadziranog strojnog učenja grade modele na setu podataka koji sadrži ulazni kao i željeni izlazni podatak. Algoritmi nadziranog učenja obavljaju klasifikaciju i regresiju. Klasifikacija se koristi kada su nam izlazni podaci ograničeni na određeni niz diskretnih vrijednosti, dok se regresija koristi kad izlazni podaci mogu poprimiti bilo koju numeričku vrijednost na nekom području. Podržano učenje je učenje optimalne strategije na temelju pokušaja s odgođenom nagradom. Ova se metoda učenja, zbog svoje općenitosti, koristi u mnogim drugim disciplinama, primjerice teorija igara, teorija vođenja, operacijska istraživanja, teorija informacije i genetski algoritam [1].

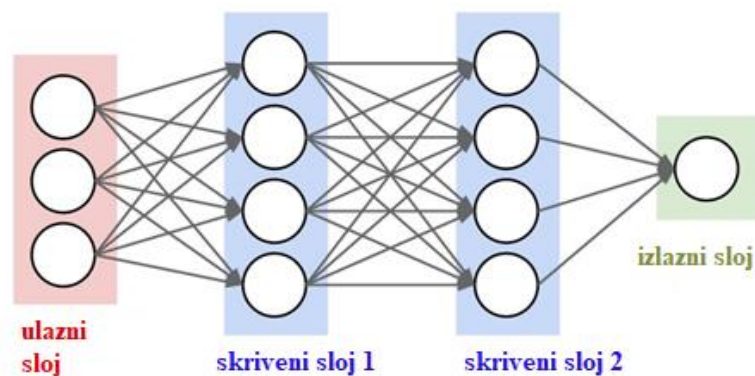
Duboko učenje je podgrupa strojnog učenja koja se bavi algoritmima inspirirani strukturom i funkcijom mozga zvana umjetna neuronska mreža. Umjetna neuronska mreža se sastoji od kolekcije povezanih čvorova zvanih umjetni neuroni. Pojedini umjetni neuron prima jedan ili više ulaza koje obradi i šalje dalje na ostale neurone koji su povezani s njime. Ulazi su brojevi te se svaki od njih množi s drugim brojem zvanim težina koji se može mijenjati tokom treninga. Izlaz iz umjetnog neurona je suma svih ulaza pomnoženih s težinama, provedena kroz nelinearnu funkciju poznatu kao aktivacijska funkcija. Primjer jednog takvog umjetnog neurona dan je na

slici 2.1. gdje su x_0, x_1, x_2 ulazi, w_0, w_1, w_2 su njihove pripadajuće težine, b je nagib, a f je aktivacijska funkcija [2].



Slika 2.1. Matematički model neurona [2]

Neuroni su grupirani u slojeve gdje različiti slojevi mogu izvoditi različite transformacije nad svojim ulazima. Neuroni jednog sloja povezani su samo s neuronima u prethodnom i sljedećem sloju. Sloj koji na svome ulazu dobiva neobrađene podatke, ujedno i prvi sloj, nazivamo ulaznim slojem. Zadnji se sloj u umjetnoj neuronskoj mreži zove izlazni sloj te nam daje konačan rezultat. To je razlog zašto neuroni izlaznog sloja nemaju aktivacijsku funkciju. Između ulaznog i izlaznog sloja se mogu nalaziti nijedan ili više skrivenih slojeva (Slika 2.2.). Vezu između dva sloja gdje se svaki neuron u jednom sloju povezuje sa svim neuronima sljedećeg sloja nazivamo potpuno poveznom vezom [2].



Slika 2.2. Primjer umjetne neuronske mreže s dva skrivena sloja [2]

Tako povezani neuroni tvore usmjereni aciklički graf te su takve mreže poznate pod nazivom klasične neuronske mreže (engl. *feedforward networks*).

2.2 Primjena dubokog učenja u obradi digitalne slike

Duboko učenje ima veliku primjenu u obradi digitalne slike. Pomoću nadzornog učenja, metode dubokog učenja uklanjaju potrebu za ručnim traganjem značajki (engl. *feature engineering*) jer svaka razina mreže samostalno nauči optimalni raspored značajki po razinama i transformira svoj ulaz u apstraktne i složene reprezentacije.

Primjene uključuju LeNet mrežu [3] treniranu na MNIST bazi podataka [4] za prepoznavanje ručno napisanih brojki i AlexNet mreža [5] koja je prva popularizirala korištenje konvolucijske neuronske mreže za računalni vid te je na ImageNet [6] ILSVRC 2012 natjecanju pobijedila s najmanjom pogreškom (top-5 pogreška u iznosu od 16 posto) raspoznavanja tisuću različitih kategorija.

2.3 Primjena dubokog učenja za prepoznavanje osobe u digitalnoj slici

Jedna od prvih primjena dubokog učenja za prepoznavanje osobe je DeepFace [7] kojeg su napravili istraživači *Facebook*-a. Oni su koristili 3D modeliranje lica kako bi primijenili po dijelovima afinu transformaciju te tako izvukli lice koje gleda prema naprijed i pomoću duboke neuronske mreže, koja ima devet slojeva, odradili prepoznavanje. Mreža ima preko 120 milijuna parametara i organizirana je kao sijamska mreža, tj. koristi iste težine dok u tandemu radi na dva različita ulaza kako bi se proračunao usporediv izlaz. Jedan izlaz je od prije izračunat pa čini osnovicu s kojom se uspoređuje drugi izlaz. Model je na Labeled Faces in the Wild [8] bazi podataka za prepoznavanje osoba postigao točnost od 97.35 posto.

VGGFace [9] je rješenje Sveučilišta u Oxfordu za prepoznavanje 2622 jedinstvenih pojedinaca koje koristi konvolucijsku mrežu VGGNet-16 [10]. VGGNet-16 je model koji je na ImageNet [6] ILSVRC 2014 natjecanju postigao točnost od 92.7 posto. Model se sastoji od 11 blokova, svaki sadrži linearni operator kojeg prati jedna ili više nelinearnost kao što su ispravljena linearna jedinica (engl. *rectified linear unit*) i *max* sažimanje. Prvih osam blokova su konvolucijski jer je linearni operator serija linearnih filtera (linearna konvolucija). Zadnja tri bloka su potpuno povezani, gdje prva dva sloja imaju izlaze dimenzije 4096 i zadnji sloj ima izlaz dimenzije 2622. Model je na Labeled Faces in the Wild [8] bazi podataka postigao točnost od 98.95 posto.

FaceNet [11] je rješenje *Google*-ovog istraživačkog tima. To je model koji direktno nauči preslikavanje iz slika lica u Euklidski prostor gdje udaljenosti odgovaraju mjeri sličnosti lica. Čim

se taj prostor proizvede, prepoznavanje osoba se implementiraju standardnim tehnikama koristeći FaceNet-ove ugradnje kao značajke. Te se ugradnje optimiziraju dubokom konvolucijskom mrežom pomoću triplet funkcije gubitaka. Triplet funkcija gubitaka radi na način da minimizira L2 udaljenost (Euklidsku normu) između slika s istom osobom i maksimizira L2 udaljenost za slike s različitim licima (osobama). Benefit ovog modela jest veća efikasnost zbog toga što za svako lice koristi 128 bajtova koliko je potrebno za mapiranje Euklidskog prostora. FaceNet ima točnost od 99.63 posto na Labeled Faces in the Wild [8] bazi podataka.

Ring loss [12] je jedna od novijih rješenja za prepoznavanje osoba u slici s kojim augmentiraju standardne funkcije gubitka kao što su *Softmax*. Ovo rješenje je na Labeled Faces in the Wild [8] bazi podataka postigao točnost od 99.5 posto.

Tablica 2.1. Točnost raznih modela za Labeled Faces in the Wild [8] bazu podatka [13]

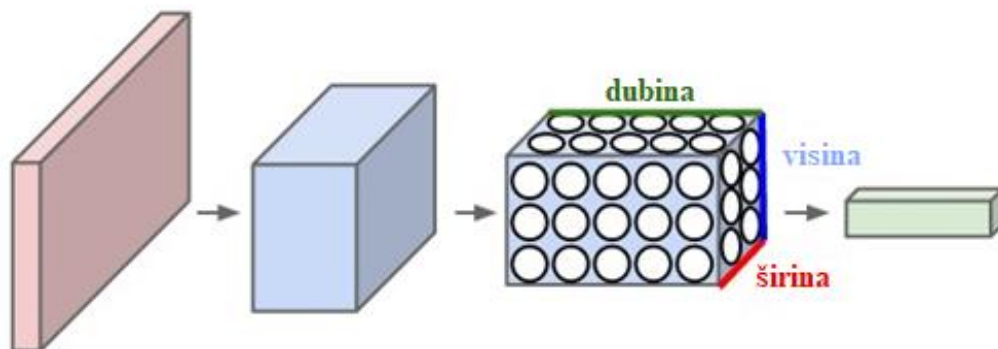
Metoda	Vrijeme objave	Funkcija gubitka	Arhitektura	Broj mreža	Trening podaci	Točnost ± devijacija
DeepFace [7]	2014	<i>softmax</i>	Alexnet	3	Facebook	97.35±0.25
FaceNet [11]	2015	triplet	GoogleNet-24	1	Google	99.63±0.09
VGGface [9]	2015	triplet	VGGNet-16	1	VGGface	98.95
Ring loss [12]	2018	prsten	ResNet-64	1	MS-Celeb-1M	99.50

3. KONVOLUCIJSKE NEURONSKE MREŽE

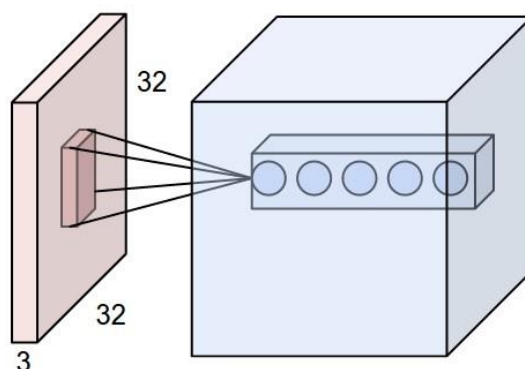
3.1. Općeniti pojmovi

Konvolucijske neuronske mreže su jako slične umjetnim neuronskim mrežama, sastoje se od neurona koji imaju parametre težine i nagiba koje je može trenirati. Svaki neuron prima ulaz, nad njim provodi skalarni produkt te ga prati nelinearnost. Cijela mreža isto funkcionira, za svake neobrađene podatke na svome ulazu daje klasu rezultata na izlazu. Jedina razlika jest eksplicitna pretpostavka da su nam ulazi slike, što nam omogućuje da kodiramo određena svojstva u arhitekturu mreže. Time se dobije lakša implementacija i smanjena količina parametara u mreži [1].

Konvolucijska neuronska mreža raspoređuje svoje neurone u tri dimenzije: širinu, visinu, dubinu (za sliku širina i visina odgovara dimenzijama slike, a dubina je tri jer predstavlja kanale za crvenu, zelenu i plavu boju). Svaki sloj mreže transformira 3D volumen ulaza u 3D volumen izlaza neurona (Slika 3.1.) [2]. Također, neuroni su povezani samo s malom površinom sloja prije, tzv. receptivno polje (engl. *receptive field*), umjesto da su svi neuroni potpuno povezani (Slika 3.2.) [2]. Tako se uzima u obzir prostorna struktura podatka te se elementi slike koji su jako udaljeni neće tretirati jednako kao elementi slike koji su blizu. Ti lokalno povezani slojevi jamče da naučeni filtri proizvedu najveći odziv na prostorno lokalni ulazni uzorak. Slaganjem više takvih slojeva vodi ka nelinearnom filtru koji postaje sve više globalan kako bi mreža prvo stvorila reprezentacije manjih dijelova ulaza te iz njih složila reprezentaciju većih područja. Bitna značajka konvolucijske neuronske mreže je i dijeljenje težina. Dijeljenje težina podrazumijeva da svi neuroni u danom konvolucijskom sloju reagiraju na istu značajku unutar određenog područja. Značajke se zato mogu detektirati bez obzira na njihovu poziciju unutar vizualnog polja. Ova nabrojana svojstva omogućuju konvolucijskoj mreži bolju generalizaciju za vizualne probleme naspram umjetnih neuronskih mreža te manje memorijske zahtjeve prilikom pokretanja mreže [2].



Slika 3.1. 3D volumen neurona konvolucijske neuronske mreže [2]



Slika 3.2. Veza neurona samo s lokalnim područjem prostornog ulaznog volumena (Svaki neuron konvolucijskog sloja je povezan samo s lokalnim područjem prostornog ulaznog volumena, ali u punu dubinu, tj. pet neurona kroz dubinu „gledaju“ na isto područje) [2]

Konvolucijska se neuronska mreža sastoji od ulaznog i izlaznog sloja, kao i mnogih skrivenih slojeva. Postoji tri glavna tipa skrivenih slojeva kojima se gradi arhitektura konvolucijske neuronske mreže: konvolucijski sloj, sloj sažimanja (engl. *pooling*) i potpuno povezani sloj [2].

Konvolucijski sloj je temeljna gradivna komponenta konvolucijske mreže i radi većinu računskih operacija. Parametri se ovog sloja sastoje od niza filtera koji se mogu naučiti. Svaki je filter prostorno mali, ali se proteže duž cijele dubine ulaznog volumena. Tijekom propagacije, svaki se filter kliže (konvulira) uzduž visine i širine ulaznog volumena i računa skalarni produkt maske filtra i ulaza za svaku poziciju. Kako se filter provuče kroz cijelu visinu i širinu ulaznog volumena, tako se proizvede 2D aktivacijska mapa koja nam daje odziv filtra na svakoj prostornoj poziciji. Intuitivno, mreža nauči filtre koji se aktiviraju kada vide neki tip vizualne značajke kao što su rubovi, orijentacije ili mrlju neke boje na prvome sloju ili složenije uzorke na višim slojevima mreže. Slažući aktivacijske mape svih filtera uzduž dimenzije dubine dobiva se cijeli izlazni volumen konvolucijskog sloja. Tri parametra kontroliraju izlazni volumen: dubina (engl. *depth*), korak (engl. *stride*) i dopunjavanje (engl. *padding*). Dubina je broj filtera koji se koriste, korak je broj za koliko elemenata slike pomičemo filter, a dopunjavanje kod konvolucijskih mreža je veličina koja definira koliko su puta popunjeni rubovi ulaznog podatka s nulama [14].

Do dimenzije izlaznog volumena (W_{izlaz}) dolazi se preko izraza danog formulom (3-1) gdje je W_{ulaz} veličina ulaznog volumena koja je opisana veličinama elementima slike za širinu i visinu ulaza.

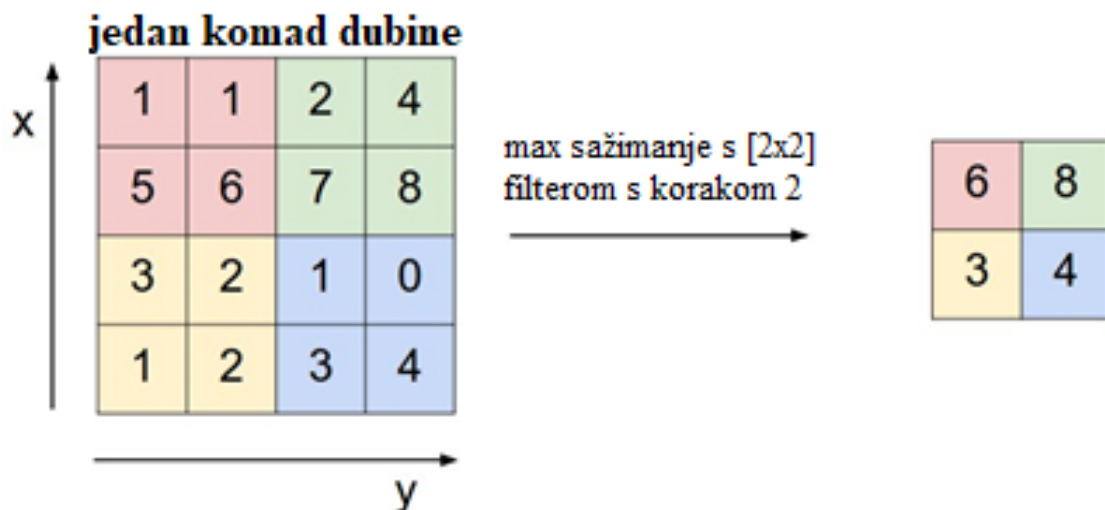
Nadalje, K je veličina receptivnog polja sa dimenzijama izraženim u elementima slike za širinu i

visinu, P je vrijednost dopunjavanja, a S je korak, odnosno broj elemenata slike za koje se pomiče filtar [2].

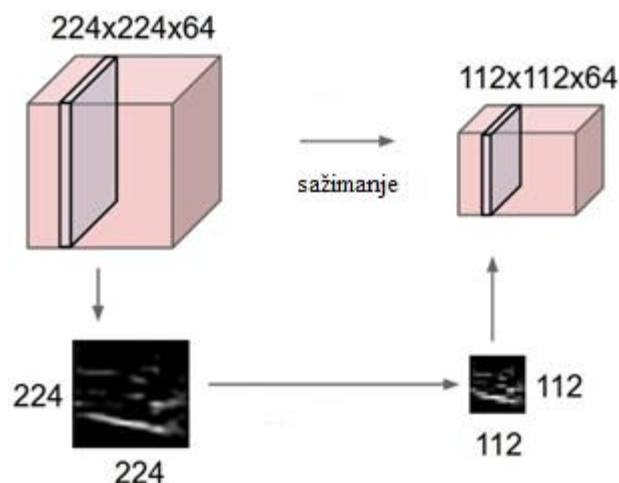
$$W_{izlaz} = \frac{W_{ulaz} - K + 2P}{S} + 1 \quad (3-1)$$

Primjerice, prvi konvolucijski sloj AlexNet-a [3] ima dubinu od $D = 96$, a na svome ulazu prima slike dimenzije $[227 \times 227 \times 3]$ ($W_{ulaz} = 227$) uz korištenje receptivnog polja veličine $[11 \times 11]$ ($K = 11$), koraka $S = 4$ i bez dopunjavanja ($P = 0$). Koristeći formulu (3-1) za izlaznu se veličinu širine i visine dobije $W_{izlaz} = 55$ zbog jednakih dimenzija ulazne širine i visine. Izlazni volumen ima oblik $[W_{izlaz} \times W_{izlaz} \times D]$, tj. veličinu $[55 \times 55 \times 96]$.

Sloj sažimanja ima funkciju nelinearnog izuzimanja. Najpoznatija takva funkcija je *max* sažimanje. Ona postupno smanjuje prostornu veličinu reprezentacije u svrhu smanjenja broja parametara i računanja u mreži te isto kontrolira pojavu prenaučenosti (engl. *overfitting*). Sloj sažimanja radi operaciju za svaki komad dubine ulaza posebno i smanjivanje obavlja pomoću *max* operacije. Najučestaliji oblik je sloj sažimanja sa filterima veličine $[2 \times 2]$ (širina i visina) i korakom dva izuzima svaki komad dubine u ulaza za dva uzduž širine i visine, izuzimajući tako 75 posto aktivacija. U ovom slučaju se svaka *max* operacija izvodi nad četiri broja (Slika 3.3.). Treba napomenuti da dimenzija dubine ostaje nepromijenjena (Slika 3.4.) [2].



Slika 3.3. *Max* sažimanje s filtrom veličine $[2 \times 2]$ i korakom dva za pojedini komad dubine [2]



Slika 3.4. Primjer izuzimanja ulaznog volumena veličine $[224 \times 224 \times 64]$ filterom veličine dva, korakom dva da se dobije izlazni volumen veličine $[112 \times 112 \times 64]$ [2]

Nakon nekoliko konvolucijskih i slojeva sažimanja dolazi potpuno povezani sloj koji izvodi visoko-razinske zaključke neuronske mreže. Neuroni u potpuno povezanom sloju imaju vezu sa svim izlazima prijašnjeg sloja, kao i kod umjetne neuronske mreže. Dodatno ima sloj gubitka koji precizira kako treniranje penalizira devijacije predviđenog izlaza od prave oznake izlaza. Ta se zadaća obavlja koristeći funkcije gubitka, kao što je *Softmax* funkcija koja se koristi za prepoznavanje jedne klase od n međusobno različitih klasa.

3.2 Načini optimizacije rada neuronskih mreža i mogući postupci za njihov bolji rad

Pri izbjegavanju pojave prenaučivosti, kada model na trening podacima ima visoku točnost i nema dobru generalizaciju na drugim podacima, koriste se razne regulacije i optimizacije rada neuronske mreže.

Dropout [15] je metoda koja u svakoj fazi treniranja za pojedine čvorove računa vjerojatnost hoće li se odbaciti ili ostaviti u mreži. Takva reducirana mreža odradi treniranje u toj fazi te vraća nazad odbačene čvorove s njihovim težinama. Izbjegavajući treniranje svih čvorova na svim trening podacima *dropout* metoda smanjuje prenaučivost i uvelike povećava brzinu treniranja.

Stohastičko sažimanje koristi stohastičku proceduru gdje se svaka aktivacija unutar područja sažimanja bira nasumično u skladu s multi nominalno distribucijom. Stohastičko sažimanje je ekvivalentan *max* sažimanju, ali s više kopija ulazne slike koja svaka ima malu lokalnu deformaciju [16].

Problem prenaučivosti se može izbjeći i s dodavanjem još trening podataka. Dodatni podaci se mogu dobiti augmentacijom postojeće baze podataka gdje se za svaku sliku radi neka od transformacija (zrcaljenje, rotacija pod određenim kutom, relativno pomicanje slike u nekom smjeru, zumiranje).

Jedna od jednostavnijih metoda prevencije prenaučivosti mreže je zaustavljanje treninga prije nego što nastupi prenaučivost. Redovitim uspoređivanjem točnosti na testnim i validacijskim podacima može se izbjeći trenutak kada točnosti na validacijskim podacima kreće opadati dok točnost na testnim podacima nastavi rasti, tj. točan trenutak nastajanja prenaučivosti.

Također se može mijenjati stopa učenja (engl. *learning rate*). Stopa učenja je parametar koji određuje do koje će mjere nove težine zamijeniti stare težine [17]. Stopa učenja je uvijek broj manji od jedan inače učenje neće nikada konvergirati ka konačnom rješenju za koje funkcija gubitka ima globalni minimum. Pravilnim odabirom stope učenja možemo izbjeći predugo trajanje učenja ili zapinjanje u nepoželjnom lokalnom minimumu.

Kako bi se računala ažuriranja težina, potrebna nam je iterativna, rekurzivna i efikasna metoda. U tu se svrhu koristi učenje s unatražnim rasprostiranjem (engl. *backpropagation*) koje koristi optimizaciju baziranu na gradijentima. *Backpropagation* je iskorišten za optimizaciju stohastičkog spusta gradijenta kako bi se podesile težine neurona računajući gradijent od funkcije gubitka. Funkcija gubitka za analizu regresije je srednja kvadratna pogreška, a za klasifikaciju je kategorička ukrštena entropija (engl. *categorical crossentropy*). *Backpropagation* izračuna gradijente, a stohastički spust gradijenta koriste te gradijente za treniranje modela optimizirajući ga. Jedan takav algoritam za optimizaciju je Adam (engl. *Adaptive Moment Estimation*) [18]. Adam za svoje optimiziranje koristi srednje vrijednosti gradijenta i adaptivne stope učenja. Te dvije vrijednosti predstavljaju procjenu prvog i drugog momenta gradijenta.

Batch size, često zvan i *mini-batch*, je parametar koji određuje koliko će se podataka provesti kroz mrežu odjedanput. Treba napomenuti da *batch size* nije isto što epoha (prolazak svih trening podataka kroz mrežu). Veličina *batch*-a određuje koliko brzo će model odraditi svaku epohu. Optimalnim će se izborom te veličine ubrzati učenje i izbjeći situacija da zbog prevelikog *batch size*-a model ne bude mogao dobro generalizirati nad podacima koje nije vidio [19].

Odabirom ispravljene linearne jedinice (*ReLU*) za aktivacijsku funkciju u modelu se sve negativne vrijednosti unutar aktivacijske mape stavljaju na nulu i ona nema utjecaja na receptivna polja konvolucijskih slojeva. Češće se koristi od drugih aktivacijskih funkcija jer se njome treniranje mreža brže provodi bez izraženog negativnog utjecaja na točnost generalizacije [5].

4. MODEL ZA PREPOZNAVANJE OSOBE U SLICI

U ovom poglavlju opisan je praktični dio rada. Odabrana je baza podataka *Aligned Face Dataset For Face Recognition* [20] koja se proširila na umjetan način provodeći razne transformacije kako bi se dobio veći skup slika za treniranje. VGGNet-16 [10] je odabrani model konvolucijske neuronske mreže za treniranje. Treniranje je provedeno na izvornoj bazi podataka [20] kao i na proširenom skupu podataka. U narednim potpoglavljima nalaze se detaljan opis svega navedenog.

4.1 Opis baze podataka

Model se trenirao na bazi podataka skinute s *Kaggle*-a [20]. Baza se sastoji od 1255 slika za osam različitih poznatih osoba (Alexandra Daddario, Amber Heard, Anne Hathaway, Emilia Clarke, Gal Gadot, Emma Watson, Kiernan Shipka i Maisie Williams). Slike su dimenzija [300x300x3] te su sve slike odrezane na takav način da je na slici samo lice osobe (Slika 4.1.).



Slika 4.1. Primjer slika u bazi podataka

Podaci su pomoću *OpenCV* biblioteke prvo učitani i sveli na dimenzije [224x224x3] te su se formatirale u matrični zapis (Slika 4.2.). Promjena veličina slika se morala odraditi kako bi se slike mogle predati ulaznom sloju mreže koji je također dimenzija [224x224x3].

```

import numpy as np
import cv2
import os
IMG_SIZE = 224
DATADIR = "/content/drive/My Drive/data"
CATEGORIES = ["alexandra_daddario", "amber_heard", "anne_hathaway", "emilia_clarke", "emma_watson", "gal_gadot", "kiernan_shipka", "maisie_williams"]
training_data = []

def create_training_data():
    for category in CATEGORIES:
        path = os.path.join(DATADIR, category)
        class_num = CATEGORIES.index(category)
        for img in os.listdir(path):
            img_array = cv2.imread(os.path.join(path, img), cv2.IMREAD_COLOR)
            new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
            training_data.append([new_array, class_num])

create_training_data()

```

Slika 4.2. Programski kod za obradu podataka

Nadalje, matricama koje reprezentiraju slike se moraju dodati oznake koje nam govore o kojoj je osobi riječ. Budući da su se koristili *Python* i *Keras* [21], oznake i matrice su se formatirale na način opisan slikom 4.3.

```

import random

random.shuffle(training_data)

X = []
y = []

for features, label in training_data:
    X.append(features)
    y.append(label)

X = np.array(X).reshape(-1, IMG_SIZE, IMG_SIZE, 3)
y = tensorflow.keras.utils.to_categorical(y, num_classes=8, dtype='int32')

```

Slika 4.3. Programski kod za formatiranje podataka prilagođenih *Keras*-u

Zbog relativno male količine podataka i želje da se izbjegne prenaučenos, potrebna je bila dodatna augmentacija skupa slika. Skup slika je augmentiran na način da se obavio niz transformacija nad slikama. Transformacije su: horizontalno zrcaljenje, rotacije u rasponu od 45 stupnjeva, pomicanje po smjeru dužine i visine u rasponu od 0.1 (omjer cijele dimenzije i količine elementa slike za koje se izvodi pomicanje), košenje slika (na način da se širina i visina slike sužavanjem za neki faktor u rasponu od 0.3 iz pravokutne forme prevede u oblik romba) i zumiranje slika po širini i visini za neki faktor u rasponu od 0.05. Nakon augmentacije se dobilo osam tisuća slika te za svaku osobu postoji tisuću različitih slika. Programski kod za generiranje dodatnih slika se nalazi na slici 4.4.


```

import tensorflow
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rotation_range=45, horizontal_flip=True, width_shift_range=0.1, height_shift_range=0.1,
                                   shear_range=0.3, zoom_range=0.05)

train_generator=train_datagen.flow_from_directory('/content/drive/My Drive/data/', target_size=(299,299),
                                                save_to_dir = '/content/aug/', shuffle=False,
                                                batch_size = 1272)

Found 1255 images belonging to 8 classes.

i=0
for batch in train_generator:
    i +=1
    if(i>=7):
        break

# sedam puta se poziva generator kako bi se napravilo 1000 slika po klasi

```

Slika 4.4. Programski kod za augmentaciju skupa slika

Kompletna baza slika korištenih u ovom radu za treniranje, validaciju i testiranje nalazi se na elektroničkom prilogu ovog rada u prilogu P.4.1.

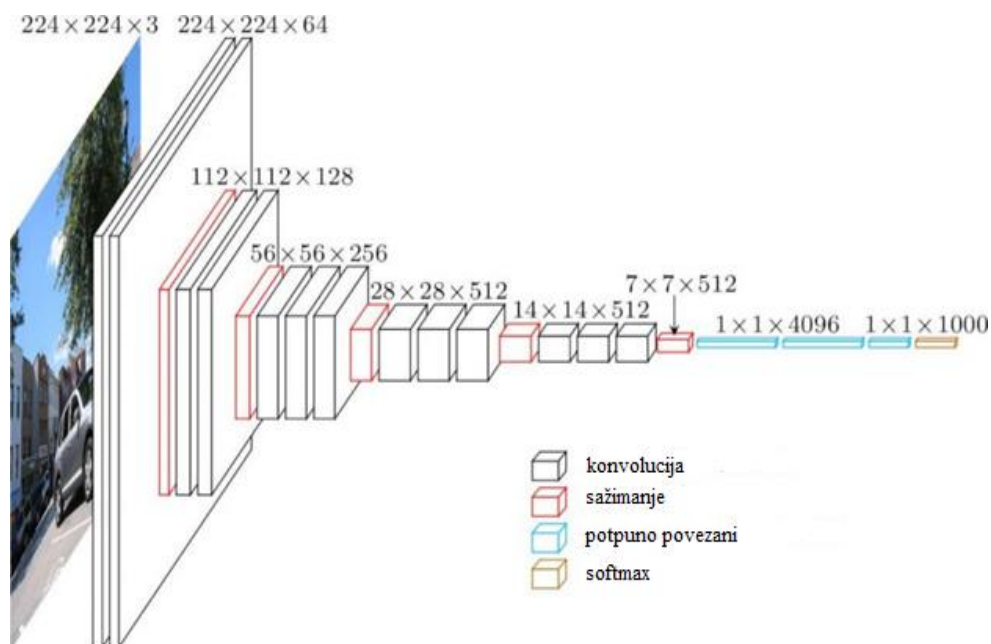
4.2 Opis programskog okruženja

Ovaj rad je napravljen u programskom okruženju *Google-a* zvanog *Colaboratory*. To je usluga na oblaku koja omogućuje izradu aplikacija dubokog učenja i besplatno korištenje njihovih grafičkih kratica (*Tesla K80 GPU*). Kod se piše u programskom kodu *Python* u okruženju *online* servisa *Jupyter Notebook*. *Colaboratory* omogućuje uvoz drugih bitnih biblioteka kao što su *numpy*, *pandas*, *matplotlib*, *OpenCV*, *scikit-learn*, *Keras* [21] i *Tensorflow* [22]. Također se može povezati s *Google Drive*-om za pohranu svih podataka i modela dobivenih u *Colaboratory*-ju.

Keras [21] je visoko razinska biblioteka otvorenog pristupa za duboko učenje pisana u programskom jeziku *Python* te koristi *Tensorflow* [16] kao jednu od glavnih osnovica u svome radu. *Tensorflow* [22] je *Google*-ova besplatna programska biblioteka otvorenog pristupa koja se koristi za tok podataka i diferencijalno programiranje. To je simbolička matematička biblioteka koja se koristi u svrhu pravljenja aplikacija strojnog učenja.

4.3 Odabir modela za prepoznavanje osoba

U radu je za prepoznavanje osobe u slici izabran model VGGNet-16 [10]. Model je izabran zbog svoje visoke točnosti na Imagenet [6] natjecanju te dobro izvlači kompleksne značajke u slikama zbog svojih naslaganih konvolucijskih slojeva. Arhitektura mreže je dana slikom 4.5.



Slika 4.5. Arhitektura odabrane mreže za prepoznavanje osobe u slici u sklopu ovog završnog rada [23]

Ulazni sloj prima sliku dimenzija [224x224x3] te se slika provodi kroz stog od trinaest konvolucijskih slojeva, čiji su filtri veličine [3x3] s korakom jedan i uvijek koriste isto dopunavanje ulaznog podatka nulama veličine jednog elementa slike u objema dimenzijama. Mreža ima pet slojeva *max* sažimanja koji prate samo neke konvolucijske slojeve i provodi svoje operacije na prozoru veličine [2x2] s korakom dva. Raspored konvolucijskih slojeva i slojeva *max* sažimanja, se može iščitati sa slike 4.5.

Tri potpuno povezana sloja dolaze nakon stoga konvolucijskih slojeva te dva imaju istu dimenziju dubine od 4096, dok zadnji ima dimenziju dubine od 1000.

Kako bi se model prilagodio za zadatak ovog rada izbačen je zadnji potpuno povezani sloj te je na mjesto njega stavljen potpuno povezani sloj s dimenzijama [1x1x8] koji pomoću *Softmax*-a obavlja zadaću klasifikacije slike za osam odabranih osoba (Slika 4.6.).

```

vgg16_model = tensorflow.keras.applications.VGG16(include_top=True, input_shape=(224,224,3), weights='imagenet')
out = Dense(8, activation='softmax', name='predictions')(vgg16_model.layers[-2].output)
custom_vgg_model = Model(vgg16_model.input, out)
for layer in custom_vgg_model.layers[:-1]:
    layer.trainable = False

```

Slika 4.6. Programski kod za mijenjanje zadnjeg potpunog poveznog sloja u traženi oblik zajedno sa zamrzavanjem težina u slojevima zaduženim za izdvajanje značajki

Modelu su uvezene težine originalno korištene za Imagenet natjecanje i sve težine osim zadnjeg sloja su mu zamrznute. To znači da pri treniranju će model mijenjati isključivo težine zadnjeg potpuno povezanog sloja koji se posebno kreirao. Ostatak mreže služi za izdvajanje značajki te ih dovodi na ulaz zadnjeg sloja koji će obaviti klasifikaciju. Ovaj princip rada se naziva preneseno učenje (engl. *transfer learning*) [2].

Također se isprobalo treniranje težina svih potpuno povezanih slojeva koje je polučilo boljim rezultatima (Slika 4.7.).

```
for layer in custom_vgg_model.layers[:-3]:  
    layer.trainable = False
```

Slika 4.7. Omogućavanje treniranja svih potpuno povezanih slojeva

4.4 Postupak treniranja modela

Prije početka treniranja podaci se moraju promiješati kako bi se izbjeglo da model pri treniranju prolazio kroz slike kako su poredane po klasama, tj. prenaučenosť. Također je dobra praksa odraditi normalizaciju podataka kako bi se računalu olakšale operacije. Budući da su podaci slike, normalizacija se odradila na način da su sve vrijednosti unutar polja matrica podijeljene s maksimalnom vrijednosti koju element slike može poprimiti (255).

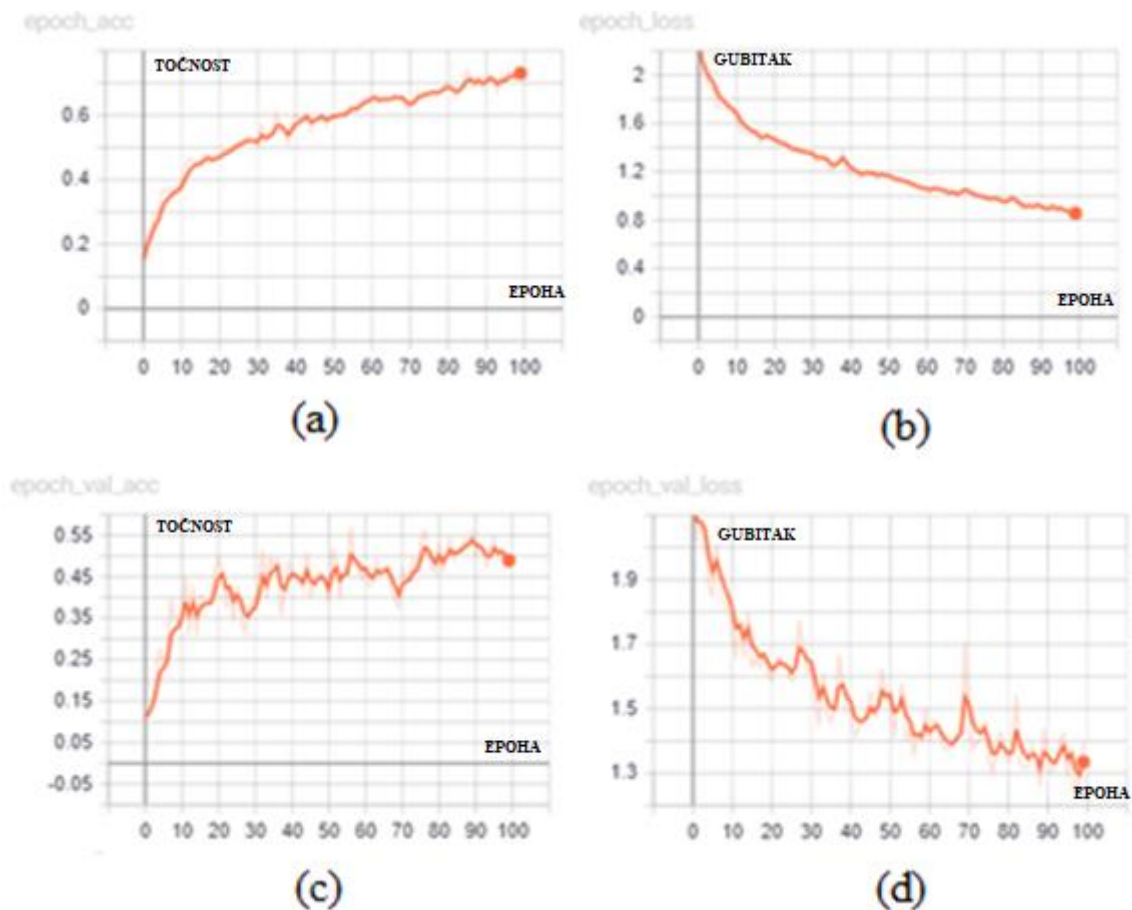
Od ukupnog broja podataka (8000) se odvojio skup od 6400 slika (80 posto), koji se dalje podijelio na skup za treniranje i skup za validaciju, i skup od 1600 slika koji će se koristiti za testiranje mreže te ih mreža prije samog testiranja neće „vidjeti“.

Treniranje se provelo nad 5760 slika (što je 90 posto od 6400 odabranih slika za treniranje i validaciju), dok se skup za validaciju sastoji od 640 slika (10 posto od 6400 odabranih za treniranje i validaciju). Pri treniranju se koristio Adam optimizator zajedno s kategoričkom ukrštenom entropijom kao funkcija gubitka (Slika 4.8.).

```
custom_vgg_model.compile(Adam(lr=0.001) , loss='categorical_crossentropy', metrics=['accuracy'])  
custom_vgg_model.fit(X_train, y_train, batch_size=50, epochs=100, validation_split=0.1, shuffle=True,  
                    callbacks=[tensorboard, checkpoint])
```

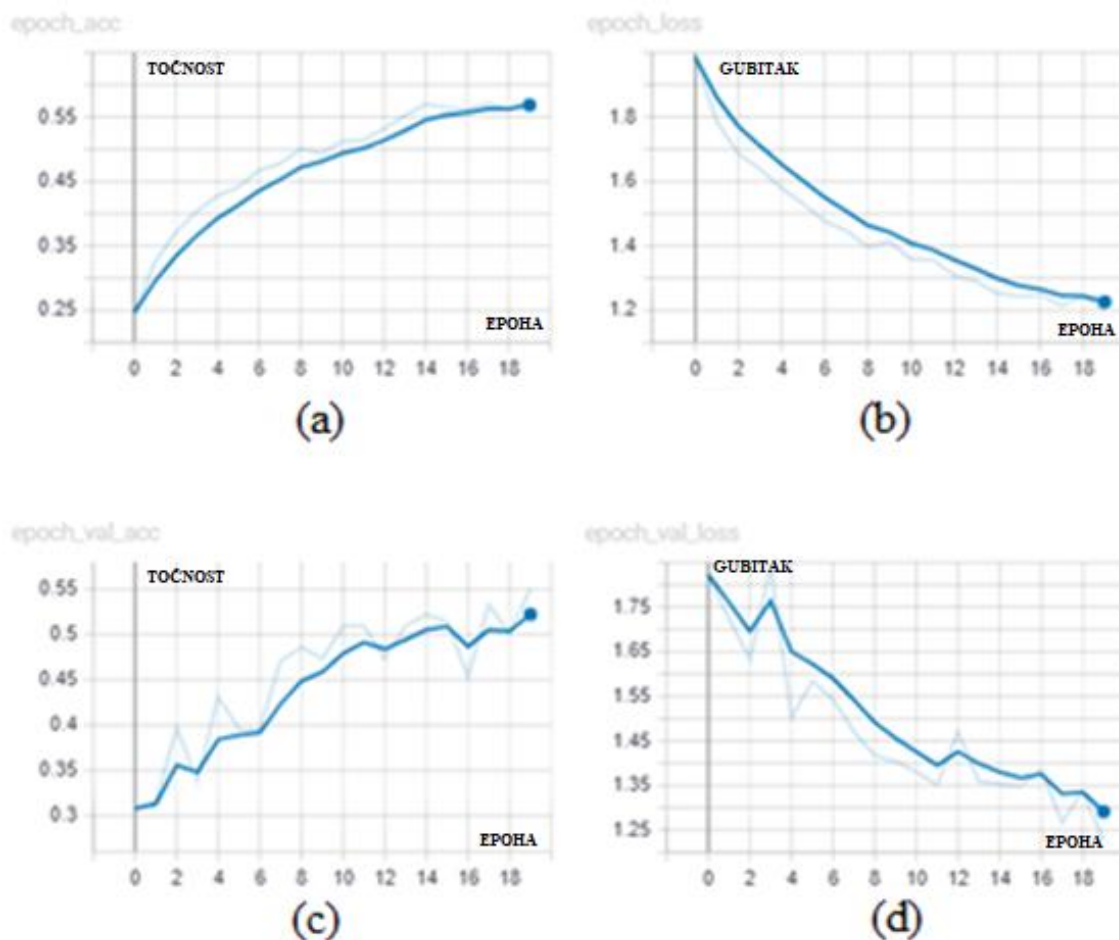
Slika 4.8. Programski kod za sastavljanje i pokretanje treniranja modela

Prvi pokrenuti ciklus treniranja je bio na izvornoj bazi slika [20] te nije polučio dobrim rezultatima. Zadano mu je bilo 100 epoha s *batch size*-om od 50 i stopom učenja 0.001. Svi ti parametri su rezultirali s modelom koji na validacijskom skupu slika ima točnost od 57.1 posto. Iz slike 4.9. se vidi da model dosegao svoju maksimalnu točnosti za izvornu bazu podataka te da je nastavljeno treniranje bi nastupila prenaučenost.



Slika. 4.9. Grafovi rezultata treniranja na izvornoj bazi podataka s 100 epoha;
 (a) točnost za trening podatke, (b) gubitci za trening podatke,
 (c) točnost za validacijske podatke, (d) gubitci za validacijske podatke

Drugi ciklus trenirana je pokrenut s augmentiranom bazom podataka. Model je za treniranje bio sastavljen od 20 epoha, *batch size*-a od 50 i stopom učenja 0.001. Nakon treniranja dobivala se točnost od 55 posto na skupu za validaciju. Primjećuje se u da je ovom ciklusu bilo zadano premalo epoha za treniranje te je zato model imao lošiju točnost od modela treniranog na izvornoj i manjoj bazi slika (Slika 4.10.).



Slika. 4.10. Grafovi rezultata treniranja na augmentiranoj bazi podataka s 20 epoha; (a) točnost za trening podatke, (b) gubitci za trening podatke, (c) točnost za validacijske podatke, (d) gubitci za validacijske podatke

Treći ciklus treniranja je pokrenut na spremljenom modelu iz prijašnjeg ciklusa uz jedinu promjenu broja epoha koja se povećala na 100. Tako istreniran modela daje točnost od 69.7 posto na skupu slika predodređenu za validaciju. Ako se obrati pozornost na graf za gubitke za skup validacije (Slika 4.11.(d)), može se uočiti da je model lagano zakoračio u prenaučenost.

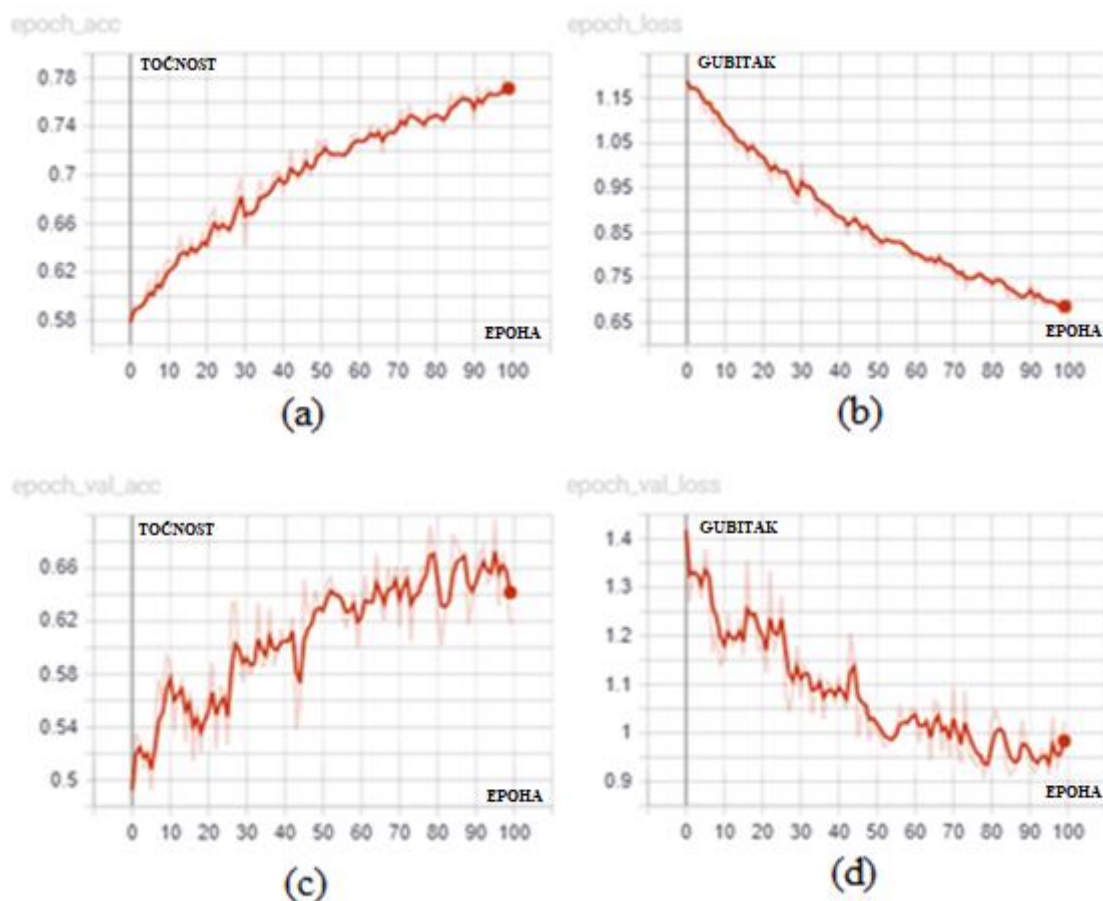
Četvrti ciklus treniranja se proveo na modelu kojemu je promijenjen samo parametar stope učenja na 0.007 te je pokrenut na novim težinama. Točnost na skupu validacije je 65.9 posto uz veliki gubitak od 1.173 (Slika 4.12.).

Peti ciklus se također mijenjala stopa učenja, ovoga put se postavila na vrijednost od 0.01. Mijenjanjem stope učenja u ovom i prijašnjem ciklusu se htjelo postići brži spust gradijenta ka minimalnoj vrijednosti gubitka. U petom ciklusu treniranja s novim težinama model postiže točnost od 73.44 posto i gubitak od 0.8778 za skup slika validacije. Kada se usporedi slika 4.12.(d)

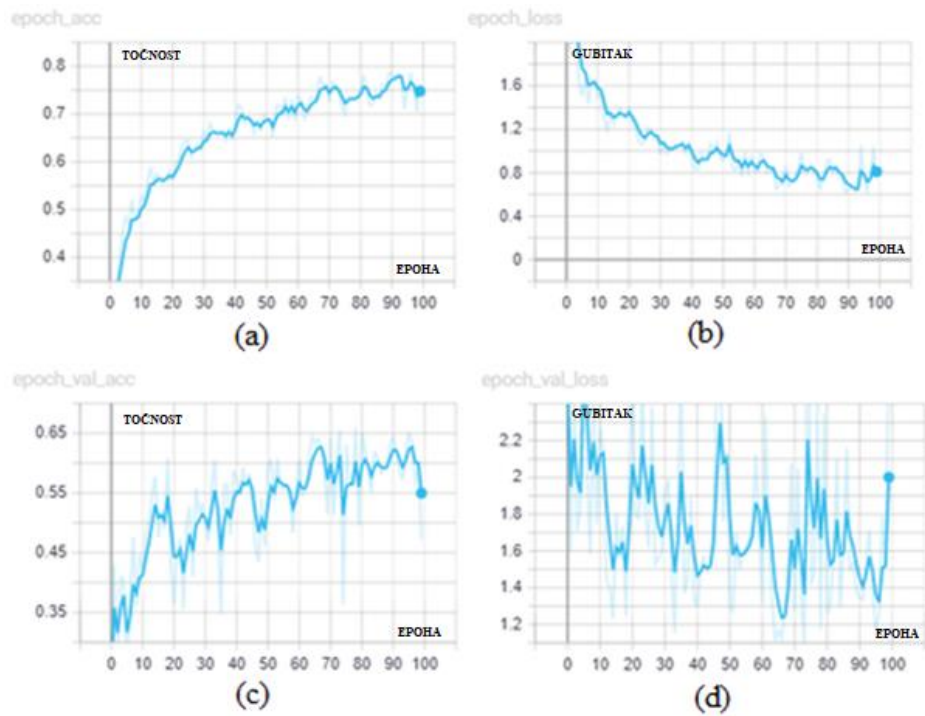
i slika 4.13.(d) uočava se, iako su u oba slučaja grafovi šiljasti i upućuju na prenaple spustove gradijenta, da je model u četvrtom ciklusu imao premalu stopu učenja te se nije uspio približiti minimumu kao model u petom ciklusu.

Treba uzeti u obzir da su dosadašnji ciklusi treniranja mijenjali isključivo težine unutar zadnjeg potpuno povezanog sloja. Ukupni broj težina koji se trenirao bio je oko 32 tisuće. Ako se želi povećati točnost modela, moraju se uključiti parametri iz još nekoliko slojeva mreže. U šestom se ciklusu zato omogućilo treniranje sva tri potpuno povezana sloja te se tako treniralo oko 120 milijuna težina. Modelu se zadalo 80 epoha, *batch-size* od 50 i stopa učenja od 0.001. Rezultati ovog ciklusa su najbolji, model ima točnost od 85.47 posto uz gubitak od 0.8626 na skupu slika za validaciju (Slika 4.14.).

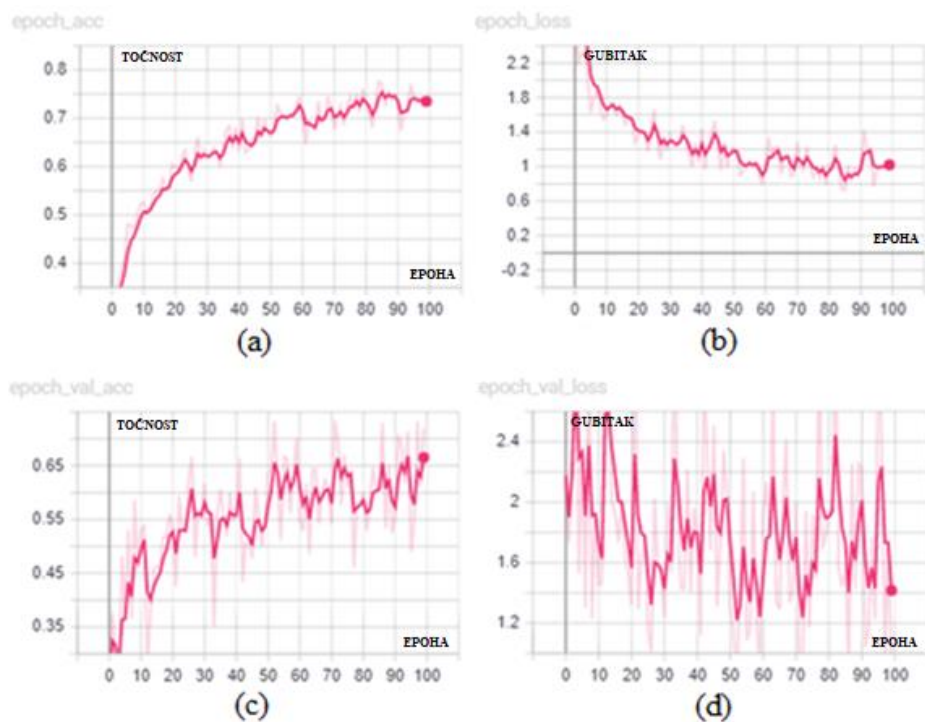
Kompletan programski kod rješenja ovog rada nalazi se na elektroničkom prilogu ovog rada u prilogu P.4.2.



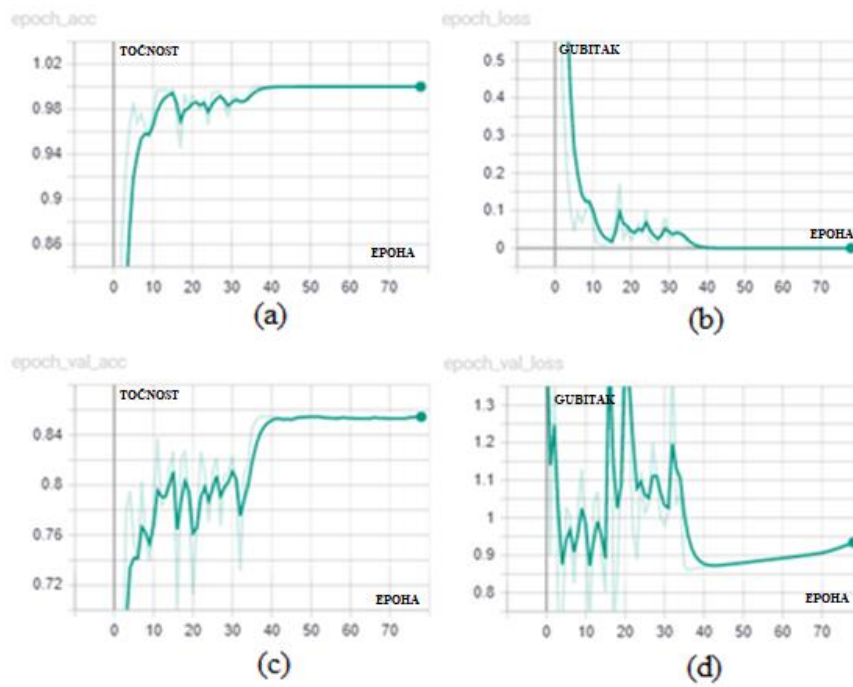
Slika. 4.11. Grafovi rezultata modela dotreniranog s 100 epoha na augmentiranoj bazi podataka; (a) točnost za trening podatke, (b) gubitci za trening podatke, (c) točnost za validacijske podatke, (d) gubitci za validacijske podatke



Slika. 4.12. Grafovi rezultata modela treniranog s 100 epoha na augmentiranoj bazi podataka uz stopu učenja 0.007; (a) točnost za trening podatke, (b) gubitci za trening podatke, (c) točnost za validacijske podatke, (d) gubitci za validacijske podatke



Slika. 4.13. Grafovi rezultata modela treniranog s 100 epoha na augmentiranoj bazi podataka uz stopu učenja 0.01; (a) točnost za trening podatke, (b) gubitci za trening podatke, (c) točnost za validacijske podatke, (d) gubitci za validacijske podatke



Slika. 4.14. Grafovi rezultata modela koji treniran težine sva tri potpuno povezana sloja u 80 epoha sa stopom učenja od 0.001; (a) točnost za trening podatke, (b) gubitci za trening podatke, (c) točnost za validacijske podatke, (d) gubitci za validacijske podatke

5. VREDNOVANJE PERFORMANSI MODELA ZA PREPOZNAVANJE OSOBA U SLICI

Za vrednovanje modela se koristila skupina slika koje model pri treniranju nije vidio te se sastoji od 1600 slika (200 slika za svaku osobu). Pozivom evaluacijske metode u *Keras*-u (Slika 5.1.) dobiva se podataka kolika je prosječna točnost predikcije modela u iznosu od 83.33 posto uz gubitak od 0.8005. Modelu je bilo potrebno 19 sekundi da obavi predikciju na skupu od 1600 slika, gdje mu je po slici trebalo 12 milisekundi.

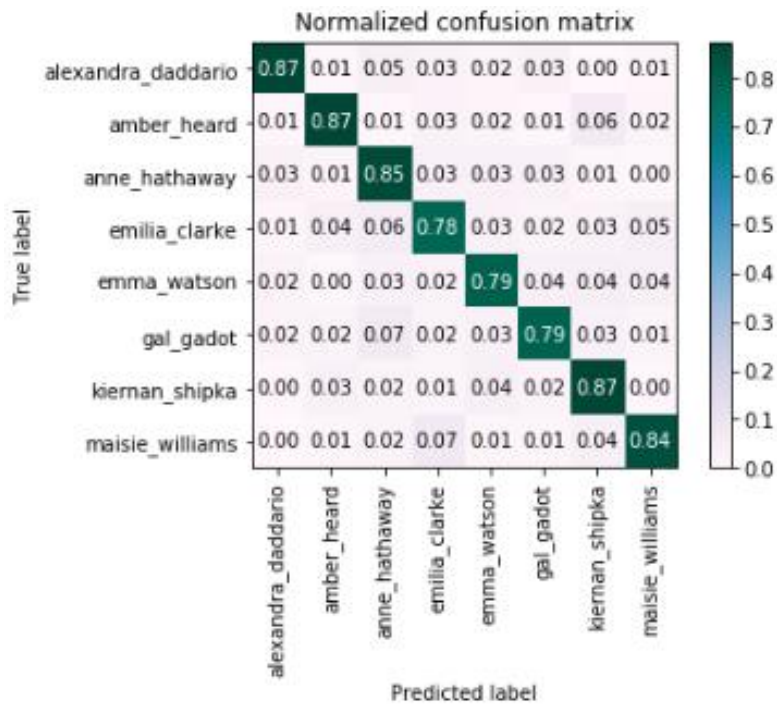
```
scores = custom_vgg_model.evaluate(X_test, y_test, verbose=1)
1600/1600 [=====] - 19s 12ms/sample - loss: 0.8005 - acc: 0.8331
```

Slika 5.1. „Programski kod za poziv metode *evaluate* u *Keras*-u“

Prilikom vrednovanja performansi dobivenog modela za prepoznavanje osoba u slici zasnovanom na dubokom učenju koristila se matrica zbunjenosti (engl. *confusion matrix*). Matrica zbunjenosti je posebni tablični raspored koji pruža vizualizaciji rezultata modela dobivenog nadzornim učenjem. Svaki red matrice predstavlja instancu stvarne klase, a svaki stupac predstavlja instancu predviđene klase. Izgled matrice zbunjenosti za konačni model (model dobiven u šestom ciklusu treniranja) dana je na slici 5.2. u matričnom obliku i slici 5.3. u tabličnom obliku.

```
Normalized confusion matrix
[[0.8693 0.005  0.0452 0.0251 0.0151 0.0302 0.      0.0101]
 [0.0053 0.867  0.0053 0.0266 0.016  0.0053 0.0585 0.016 ]
 [0.0274 0.0137 0.8539 0.032  0.032  0.0274 0.0091 0.0046]
 [0.0102 0.0355 0.0558 0.7766 0.0305 0.0203 0.0254 0.0457]
 [0.0245 0.0049 0.0294 0.0245 0.7941 0.0392 0.0441 0.0392]
 [0.0237 0.0237 0.0664 0.0237 0.0284 0.7915 0.0284 0.0142]
 [0.0047 0.028  0.0187 0.014  0.0374 0.0187 0.8738 0.0047]
 [0.      0.0119 0.0179 0.0714 0.0119 0.0119 0.0357 0.8393]]
```

Slika 5.2. Matrica zbunjenosti u matričnom obliku



Slika 5.3. Matrica zbunjenosti u tabličnom obliku

Prateći dijagonalu matrice (Slika 5.3.) vidljiva je točnost modela kada pravilno pogodi osobu na slici koju je analizirao.

Model za pet klasa ima visoku točnost (preko 84 posto) i nešto manju točnost za ostale tri klase (oko 80 posto). Gledajući pogrešne predikcije uočava se da su one reda veličine 7 posto i manje, gdje u najgorim slučajevima imamo po tri netočne predikcije koja svaka iznosi oko 6 posto.

Budući da se baza podataka sastoji od slika osam žena bijele rasne skupine koje iz tog razloga dijele više sličnih značajki lica, razumljiva je pogreška prilikom prepoznavanja osobe. Primjerice, vidi se da šest žena dijeli slični oblik brade, na većini slika nose crveni ruž te da ih petero ima tamniju boju kose. Također, oblik frizure, dob pojedine žene i šminka utječu na crte lica kao i različiti uvjeti osvjetljenja. Kriva predviđanja proizlaze iz tih sličnosti. Najveća pogreška prepoznavanja uočena je između Maisie Williams i Emilia Clarke koja iznosi 7.14 posto. Mogući razlozi tog pogrešnog prepoznavanja su sličnosti oblika lica, veličine i boje očiju, oblik obrva te slična dob.

Model je najpouzdaniji prilikom prepoznavanja Kiernan Shipka-e s točnosti od 87.38 posto, što se objašnjava činjenicom da specifična obilježja (izraz lica, tena i veličina usana) se uvelike razlikuju od ostalih žena.

Analizom dobivenih rezultata se zaključuje da model češće ispravno prepozna o kojoj je osobi riječ. Nadalje, ovaj bi se model mogao primijeniti u stvarnom okruženju i od njega se može očekivati točno prepoznavanje osobe na temelju njenog lica.

Performanse modela se mogu poboljšati proširenjem baze raznovrsnijim slikama za treniranje čime će se temeljitije naučiti značajke pojedine osobe. Kako bi performanse modela bile bolje, moguće je korištenje triplet funkcije gubitka [9]. Provođenje treniranja nad svim težinama mreže također bi moglo polučiti boljim rezultatima.

6. ZAKLJUČAK

Ovaj rad primjenjuje principe dubokog učenja za rješavanje problema prepoznavanja osobe u digitalnoj slici temeljem njenoga lica. Odabrana je konvolucijska neuronska mreža po uzoru na arhitekturu mreža VGGNet-16 i VGGFace. Koristeći javno dostupno bazu slika te povećavanjem broja slika augmentacijom trenirala se konvolucijska mreža koja je trebala obaviti klasifikaciju slika za osam identiteta. Treniranje se postiglo metodama prenesenog učenje, gdje se slojevi za izdvajanje značajki nisu trenirali već su se samo trenirala tri potpuno povezani slojevi. Općenito, ovakvim pristupom se dobilo relativno dobro rješenje za prepoznavanje osoba u slici na temelju lica. Točnost predikcije modela bi sigurno bila veća kada bi se dodalo još slika, koristila triplet funkcija gubitka i dodatno trenirale sve težine u konvolucijskim slojevima.

LITERATURA

- [1] T. M. Mitchell, Machine Learning, McGraw Hill, 1997
- [2] CS231n: Convolutional Neural Networks for Visual Recognition, <http://cs231n.github.io/> (kolovoz, 2019)
- [3] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, No. 11, Vol. 86, pp. 2278-2324, November 1998
- [4] THE MNIST DATABASE of handwritten digits, <http://yann.lecun.com/exdb/mnist/> (rujan, 2019)
- [5] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Network, University of Toronto, Kanada, 2012, dostupno na: <https://www.cs.toronto.edu/~fritz/absps/imagenet.pdf>
- [6] Imagenet, <http://www.image-net.org/> (rujan, 2019)
- [7] Y. Taigman, M. Yang, M. A. Ranzato, L. Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, IEEE Conference on Computer Vision and Pattern Recognition, 2014, dostupno na: https://zfpascal.net/cvpr2014/Taigman_DeepFace_Closing_the_2014_CVPR_paper.pdf
- [8] Labeled Faces in the Wild, <http://vis-www.cs.umass.edu/lfw/> (rujan, 2019)
- [9] O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, BMVC, Vol. 1, pp. 6, 2015, dostupno na: <https://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf>
- [10] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, University of Oxford, September 2014, dostupno na: <https://arxiv.org/abs/1409.1556>
- [11] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering, CVPR, pp. 815–823, 2015, dostupno na: <https://arxiv.org/abs/1503.03832>
- [12] Y. Zheng, D. K. Pal, M. Savvides, Ring loss: Convex Feature Normalization for Face Recognition, February 2018, dostupno na: <https://arxiv.org/abs/1803.00130>

- [13] M. Wang, W. Deng, Deep Face Recognition: A Survey, April 2018, dostupno na: <https://arxiv.org/abs/1804.06655>
- [14] Convolutional Neural Networks (CNNs): An Illustrated Explanation, <https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/> (rujan, 2019)
- [15] S. Nitish, C. G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from overfitting, Journal of Machine Learning Research, No. 1, Vol. 15, pp. 1929–1958, 2014
- [16] M. D. Zeiler, R. Fergus, Stochastic Pooling for Regularization of Deep Convolutional Neural Networks, January 2013, dostupno na: <https://arxiv.org/abs/1301.3557>
- [17] H. Zulkifli, Understanding Learning Rates and How It Improves Performance in Deep Learning, Towards Data Science, January 2018, dostupno na: <https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10>
- [18] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, December 2014, dostupno na: <https://arxiv.org/abs/1412.6980>
- [19] Batch Size in a Neural Network explained, deeplizard, November 2017, dostupno na: <https://deeplizard.com/learn/video/U4WB9p6ODjM>
- [20] Aligned Face Dataset For Face Recognition, <https://www.kaggle.com/frules11/pins-face-recognition> (kolovoz, 2019)
- [21] Keras Documentation, <https://keras.io/> (rujan, 2019)
- [22] Tensorflow, <https://www.tensorflow.org/> (rujan, 2019)
- [23] VGG16 – Convolutional Network for Classification and Detection, Neurohive, November 2018, <https://neurohive.io/en/popular-networks/vgg16/>

SAŽETAK

U ovom radu objašnjeni su principi rada dubokog učenja za rješavanje problema prepoznavanja osobe u digitalnoj slici. Općenito se za rješavanje problema u računalnom vidu koriste konvolucijske neuronske mreže. Ovaj rad se bavi prilagođavanjem postojeće konvolucijske neuronske mreže koja će obavljati prepoznavanje osoba. Tehnikom prenesenog učenja na VGGNet-16 modelu je izgrađena mreža koja može prepoznati osam jedinstvenih identiteta. Treniranje je obavljeno na javno dostupnoj bazi podataka slika lica čiji se broj slika dodatno povećao augmentacijom skupa tih slika. Dio slika iz baze podataka se izdvojio od trening podataka za kasnije vrednovanje dobivenog modela. Vrednovanjem modela na tim slikama dobije se prosječna točnost prepoznavanja od 83 posto.

KLJUČNE RIJEČI:

umjetna inteligencija, strojno učenje, duboko učenje, konvolucijske neuronske mreže, prepoznavanje osoba

RECOGNIZING PERSONS IN AN IMAGE BASED ON DEEP LEARNING

Abstract

This paper explains the principles of deep learning work to solve the problem of recognizing a person in a digital image. Generally, convolutional neural networks are used to solve computer vision problems. This paper deals with the adjustment of an existing convolutional neural network that will perform recognition of persons. Using transfer learning on VGGNet-16 model a network has been built that can recognize eight unique identities. The training was performed on a publicly available database of facial images whose number of images was further increased by augmenting a set of these images. Part of the images from the database have been sorted out from the training data for later evaluation of the made model. Evaluating the models with these images yields an average recognition accuracy of 83 percent.

KEYWORDS:

artificial intelligence, machine learning, deep learning, convolutional neural networks, person recognition

ŽIVOTOPIS

Luka Budak rođen je 20.kolovoza 1994. godine u Vinkovcima. Završio je osnovnu školu Ivana Gorana Kovačića u Vinkovcima. Nakon završene osnovne škole upisao je Gimnaziju Matije Antuna Reljkovića smjer prirodoslovno-matematički u Vinkovcima. Maturirao 2013. godine i upisao preddiplomski sveučilišni studij Elektrotehnike i informacijskih tehnologija na Fakultetu elektrotehnike i računarstva u Zagrebu. Godine 2017. upisao je preddiplomski sveučilišni studij Elektrotehnike i informacijskih tehnologija na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

PRILOZI

P.4.1. Baza slika korištenih u ovom radu za treniranje, validaciju i testiranje

P.4.2. Programski kod rješenja

Priložene stavke su na CD-u uz ovaj rad.