Prepoznavanje lica korišćenjem veštačkih neuronskih mreža

Seminarski rad u okviru kursa
Računarska inteligencija
Matematički fakultet

Anđelković Dragica, Mandić Igor
andjelkovic.dragica96@gmail.com, igormandic996@gmail.com

24. maj 2019

Sažetak

Sadržaj

1 Uvod

U ovom radu opisan je program za automatsko prepoznavanje lica. Cilj je da se na osnovu slike lica, prepozna identitet osobe, pri čemu slike lica te osobe već postoje u bazi. Program je zasnovan na mašinskom učenju, gde je glavni zadatak je naći dobar klasifikacioni model. Sastoji od tri faze. U prvoj fazu se vrši prikupljanje i obrada podataka koji će predstavljati ulaz. Nakon toga, ulazni skup podataka se deli na dva dela, za trening i za test. Skup za trening je veći, oblično obuhvata 80 procenata ulaza, dok ostatak pripada test skupu. Učenje modela se vrši na trening podacima, da bi se na kraju taj model testirao na podacima za test. U implementaciji ovog programa korišćene su veštačke neuronske mreže.

2 Opis problema

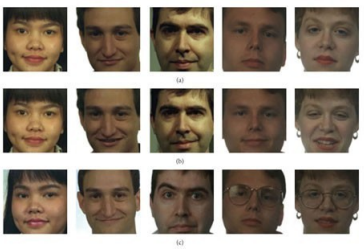
Svi ljudi imaju različita lica, odnosno lice predstavlja jedinstveni identifikator za svakog čoveka. Za ljude razlikovanje lica predstavlja lak zadatak, čak i kada se ono nalazi u različitim položajima ili se posmatra iz različitih uglova. Međutim, za računar ovo predstavlja veoma složen zadatak. Razlog toga je što postoji veliki broj različitih oblika u kojima se isto lice može pojaviti, a nije moguće sve te slučajeve isprogramirati.

2.1 Opis baze

Baza podataka se sastoji od 50 različitih osoba muškog i ženskog pola. Folderi su numerisani brojevima 0-49 i u svakom folderu se nalazi 20-50 slika jedne osobe na osnovu kojih neuronska mreža uči da prepozna osobu. U ovom seminarskom radu korišćene su neke od poznatijih baza kao što su FERET database, The Extended Yale Face Database B i Our database koje su detaljnije opisane u sledećim poglavljima.

2.1.1 FERET database

Baza podataka FERET je prikupljena u 15 sesija između avgusta 1993. godine i jula 1996. godine. Baza sadrži 1564 skupa slika sa ukupno 14126 slika koje uključuju 1199 pojedinaca i 365 duplikata skupova slika (slika 1). Skup duplikata je drugi skup slika osobe koja se već nalazi u bazi, ali su slike fotografisane drugog dana. Za neke pojedince, period između prvog i drugog fotografisanja je bio duži od dve godine, što je omogućilo istraživačima da prvi put prouče promene u izgledu pojedinca koje nastaju nakon perioda dužeg od godinu dana [4].



Slika 1: Primeri slika iz baze FERET

2.1.2 The Extended Yale Face Database B

Baza sadrži 16128 slika 28 različitih osoba u 9 poza i 64 uslova osvetljenja (slika 2) [3][2].

Slika 2: Primeri slika iz baze The Extended Yale Face Database B

2.1.3 Our database

Baza sadrži 6660 slika 90 različitih osoba. Svaka osoba ima 74 slika, pri čemu je 37 slika snimnjeno na svakih 5 stepeni okretanja osobe od desnog profila (definisano kao +90°) do levog profila (definisano kao -90°). Preostalih 37 slika su generisane od postojećih 37 slika pomoću softvera za obradu komercijalnih slika koji ih okreće horizontalno [5].

Slika 3: Primeri slika iz baze Our database

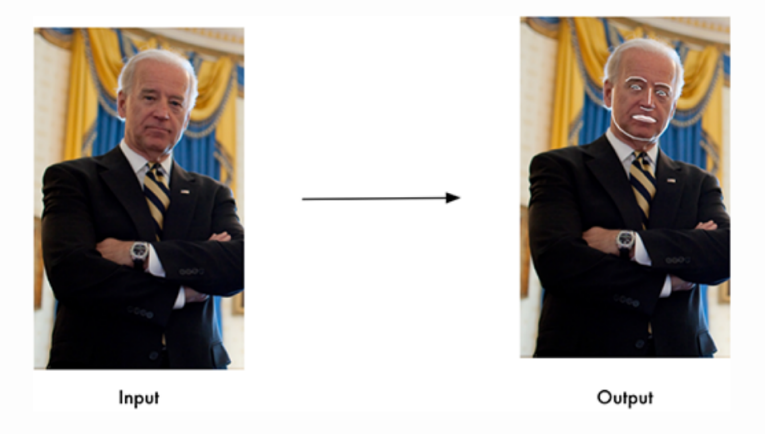
3 Opis rešenja

Problem prepoznavanja lica može se rešiti korišćenjem neuronskih mreža. Program se sastoji iz tri faze:

- Pretprocesiranje
- Učenje modela na trening podacima
- Testiranje modela

3.1 Pretprocesiranje

Program za ulaz dobija slike lica, i u prvoj fazi potrebno je obraditi te slike i napraviti ulaz pogodan za neuronsku mrežu. Na početku se skaliraju slike, tako da sve budu iste veličine. Nakon toga je potrebno detektovati lice na slici, i to se postiže korišćenjem biblioteke `face_recognition`. Pomoću funkcije `load_image_file` vrši se učitavanje slike, a funkcijom `face_landmarks` dobija se lista svih lica koja se nalaze na učitanoj slici (slika 4). Svim licima sa slike je moguće pristupiti, i za svako lice se čuvaju kooridinate tačaka koje ga čine.



Slika 4: Detekcija lica

Kao što je već napomenuto, svi ljudi imaju različito lice, a jedna od razlika je i dužina određenih delova lica. U ovom projektu u obzir su uzete sledeće dužine:

- širina lica
- dužina nosa
- širina očiju
- širina usne
- širina obrva
- rastojanje između očiju
- rastojanje između obrva

Za računanje dužine i širine delova lica korišćeno je euklidsko rastojanje. Kod za obradu slika nalazi se na slici 1.

```
id = os.path.basename(os.path.dirname(putanja)).replace(" ", "-").
    lower()

slika = PIL.Image.open(putanja)

obradjena = slika.resize((290, 320))
obradjena.save('/home/igor/Desktop/RI_PROJEKAT/za_obradu.jpg')

image = face_recognition.load_image_file(
    "/home/igor/Desktop/RI_PROJEKAT/za_obradu.jpg")

listaLica = face_recognition.face_landmarks(image)
```

```
if(listaLica != []):
    for lice in listaLica:
        ids.append(id)
        lice = collections.OrderedDict(sorted(lice.items(), key=
    lambda t: t[0]))
        for crteLica in lice.keys():
            if (crteLica != "right_eye" and crteLica != "left_eye"
     and crteLica != "top_lip" and
                    crteLica != "bottom_lip"):
                rezultat.append(duzinaISirina(lice[crteLica]))
            else:
                if (crteLica != "bottom_lip"):
                    rezultat.append(duzinaISirinaUstaIOciju(lice[
    crteLica]))

        rezultat.append(euklidskoRastojanje(lice["left_eye"][len(
    lice["left_eye"]) - 1],
                                            lice["right_eye"][0]))
        rezultat.append(euklidskoRastojanje(lice["left_eyebrow"][
    len(lice["left_eyebrow"]) - 1],
                                            lice["right_eyebrow"
    ][0]))
```

Listing 1: Pretprocesiranje

3.2 Učenje modela na trening podacima

Nakon faze pretprocesiranja sledi učenje modela na trening podacima pomoću veštačke neuronske mreže (poglavlje 3.2.1). Mreža se sastoji od četiri sloja, jednog ulaznog, dva skrivena sloja i jednog izlaznog sloja. Ulaz u veštačku neuronsku mrežu su vektori veličine 10, koji predstavljaju izračunate parametre za svaku osobu. Izlaz je vektor veličine 50, koji predstavlja verovatnoću da slika priada određenoj klasi tj. osobi. Kao aktivaciona funkcija ulaznog sloja i skrivenih slojeva korišćena je Ispravljena linearna jedinica (poglavlje 3.2.2), koja se zbog svoje nelinearnosti dobro pokazala u praksi. S obzirom da se u ovom radu bavimo klasifikacijom lica za aktivacionu funkciju izlaznog sloja je korišćena Softmax (poglavlje 3.2.3). Kod za učenje modela se nalazi na slici 2.

```
model = Sequential([
Dense(20, input_dim=10),
Activation('relu'),
Dense(50),
Activation('relu'),
Dense(70),
Activation('relu'),
Dense(10),
Activation('softmax'),
])

model.compile(optimizer= 'adadelta',
          loss='categorical_crossentropy',
          metrics=['accuracy'])

trainLabels = keras.utils.to_categorical(y_train)

model.fit(x_train, trainLabels, epochs=1000)
```

Listing 2: Učenje modela

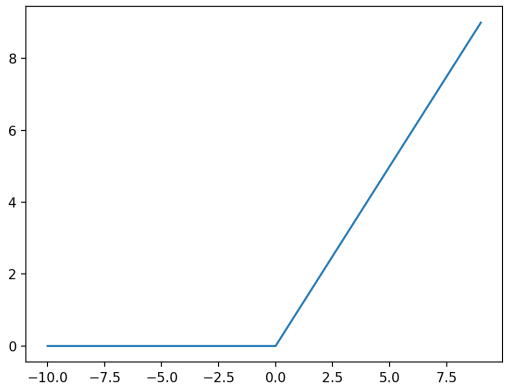
3.2.1 Neuronske mreže

3.2.2 Ispravljena linearna jedinica

Ispravljena linearna jedinica (eng. *ReLU - Rectified Linear Unit*) je danas najpopularniji izbor za aktivacionu funkciju. Ova aktivaciona funcija nije ograničena, ni diferencijabilna, ali je nelinearna. Definiše se na sledeći način:

$$ReLU(x) = max(0, x)$$

gde je x ulaz u neuron. Funkcija je jednaka identitetu za pozitivne vrednosti, a nuli za negativne vrednosti (slika 5).



Slika 5: ReLU aktivaciona funkcija

ReLU aktivaciona funkcija omogućava brže i delotvornije učenje neuronske mreže na velikim i složenijim bazama podataka. Račun je jeftin jer nema komplikovane matematike, zbog čega je modelu potrebno manje vremena za učenje. Konvergira brže i nema problem sa malim gradijenatima, kao što to imaju sigmoidna i tangetna funkcija.

3.2.3 Softmax

Softmax se uglavnom koristi kod problema klasifikacije podataka u N kategorija. Izlazi poslednjeg sloja se prosledjuju softmax funkciji,koja vraća vektor verovatnoća pripadnosti ulaza svakoj od klasa, odnosno vraća vektor *y'*, gde su $y_i' = P(y = i|x)$, i = 1,...,n. Svi y_i' imaju vrednost između 0 i 1, tako da je suma svih y_i' jednaka 1. Funkcija je definisana je na sledeći način:

$$softmax(x) = \left(\frac{e^{x_1}}{\sum_{i=1}{}_N e^{x_i}}, ..., \frac{e^{x_N}}{\sum_{i=1} + N e^{x_i}} \right)$$

3.2.4 AdaDelta

Za optimizaciju se koristi metoda AdaDelta. Ova metoda se dinamički prilagođava tokom vremena i spada u optimizacione algoritme prvog reda (eng. *first order algorithm*). Ne zahteva ručno podešavanje stope učenja, nije osetiljiva na hiperparametre, deli dinamičku stopu učenja po dimenzijama, robusna je na šumoviti gradijent i izbor arhitekture [6].

Cilj mnogih metoda mašinskog učenja je ažuriranje skupa parametara x sa ciljem da se optimizuje ciljna funkcija f(x). Ovo često uključuje i iterativnu proceduru koja primenjuje promene na parametre Δx u svakoj iteraciji algoritma. Označavajući parametar x_t u t-oj iteraciji, pravilo za ažuriranje je sledeće:

$$x_{t+1} = x_t + \Delta x_t$$

Algoritam AdaDelta je poboljšana verzija algoritma gradijentnog spusta koji pokušava da optimizuje ciljnu funkciju prateći smer spuštanja negativnog gradijenta g_t. Ovaj opšti pristup se može primeniti za ažuriranje bilo kojih parametara na kojima se može primeniti derivacija:

$$\Delta x_t = -\eta g_t$$

gde je g_t gradijent parametara u t-oj iteraciji $\frac{\partial f(x_t)}{\partial x_t}$ i η je stopa učenja koja kontroliše veličinu koraka u smeru negativnog gradijenta.

Adadelta je dodatak na Adagrad algoritam koji ne akumulira sumu kvadrata svih prethodnih gradijenata kao što radi Adagrad, već akumulira prethodnih w gradijenata. [1] Budući da je skladištenje prethodnih w kvadrata gradijenata neefikasno, akumulacija se implementira kao eksponencijalno opadajući prosek kvadrata gradijenata (eng. *exponen- tially decaying average of the squared gradients*). U vremenu t trenutni prosek $E[g^2]_t$ se računa:

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho) g_t^2$$

gde je ρ konstanta koja označava stopu propadanja. S obzirom da se u algoritmu traži kvadratni koren ove vrednosti kod ažuriranja parametara, kada se kvadrira, postaje srednja kvadratna greška (eng. *RMS - Root Mean Square*) prethodnih kvadrata gradijenata do vremena t:

$$RMS[g]_t = \sqrt{E[g^2]_t + \epsilon}$$

gde je ϵ konstanta. Rezultujući parametar ažuriranja je:

$$\Delta x_t = -\frac{\eta}{RMS[g]_t} g_t$$

Algoritam 1 Računanje ADADELTA ažuriranja u trenutku t

Require: brzina propadanja ρ, konstanta ϵ
Require: početni parametar x_1
1: Inicijalizovati akumulirane promenljive $E[g^2]_0 = 0, E[\Delta x^2]_0 = 0$
2: **for** *t=1:T* **do**
3: Izračunati gradijent: g_t
4: Akumulirati gradijent: $E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho) g_t^2$
5: Izračunati ažuriranje: $\Delta x_t = -\frac{\eta}{RMS[g]_t} g_t$
6: Akumulirati ažuriranja: $E[\Delta x^2]_t = \rho E[\Delta x^2]_{t-1} + (1-\rho)\Delta x_t^2$
7: Ažurirati: $x_{t+1} = x_t + \Delta x_t$
8: **end for**

3.3 Testiranje modela

U poslednjoj fazi, vrši se testiranje dobijenog modela. Prvo se vrši testiranje na trening podacima, a zatim i na test. Trening podaci obuhvataju 80 procenata ulaza, a test podaci 20 procenata. Kod za test se nalazi na slici 3.

```
1000
y_pred_train = model.predict_classes(x_train)
1002 y_pred_test = model.predict_classes(x_test)

1004 matricaKonfuzije = confusion_matrix(y_train, y_pred_train)
matricaKonfuzijeTest = confusion_matrix(y_test, y_pred_test)
```

Listing 3: Testiranje modela

Funkciom `predict_classes` se određuje kojoj klasi svaki ulazni podatak pripada. Prvo je primenjena na trening podacima, a zatim i na test. Od dobijenih rezultata, napravljena je matrica konfuzije.

4 Eksperimentalni rezultati

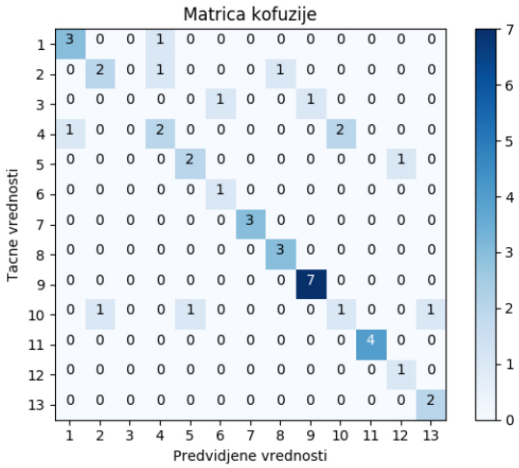
Testiranjem modela, dobijeni su različiti rezultati. Do njih smo došli eksperimentisanjem sa parametrima funkcija neuronske mreže, kao što su broj epoha, broj slojeva, različite funkcije aktivacije. Pored toga, rezultati zavise i od baze na koju se primenjuje neuronska mreža.

4.1 Rezultati za bazu FERET

Kao ulaz u neuronsku mrežu uzet je uzorak od 13 osoba. Dobijeni su sledeći rezultati:

- preciznost na trening skupu je 99%
- preciznost na test skupu je 76%

Matrica konfuzije se nalazi na slici 6.



Slika 6: Matrica konfuzije

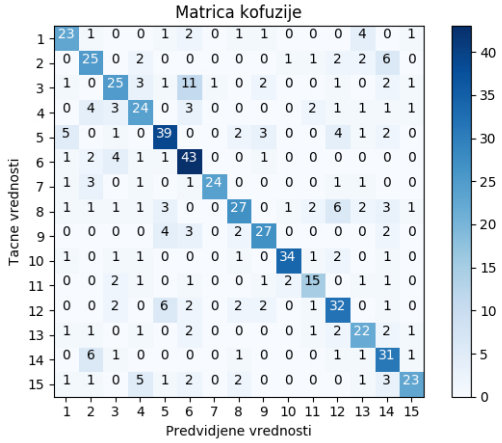
Parametar broj epoha je postavljen na 1000, i sa povaćanjem ovog parametra, preciznost ostaje približno ista. Ako se broj epoha smanji, rezultati dosta variraju. Sa povećanjem broja slika, rezultati opadaju, i sve više variraju.

4.2 Rezultati za bazu The Extended Yale Face Database B

Kao ulaz u neuronsku mrežu uzet je uzorak od 15 osoba. Dobijeni su sledeći rezultati:

- preciznost na trening skupu je 98%
- preciznost na test skupu je 72%

Matrica konfuzije se nalazi na slici 7.



Slika 7: Matrica konfuzije

Kao i za prethodnu bazu, broj epoha je postavljen na 1000. Sa povaćanjem ovog parametra, preciznost ostaje približno ista. Ako se broj epoha smanji, rezultati dosta variraju. Ukoliko se broj slika poveća, rezultati su lošiji i nestabilniji.

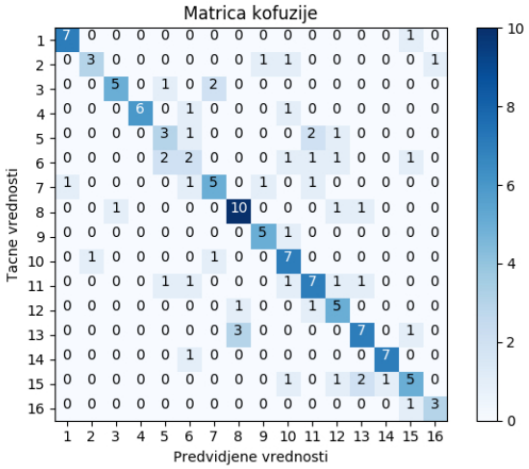
4.3 Rezultati za bazu Our database

Kao ulaz u neuronsku mrežu uzet je uzorak od 15 osoba. Dobijeni su sledeći rezultati:

- preciznost na trening skupu je 100%
- preciznost na test skupu je 60%

Matrica konfuzije se nalazi na slici 8.

Kao i za prethodnu bazu, broj epoha je postavljen na 1000. Sa povaćanjem ovog parametra, preciznost ostaje približno ista. Ako se broj epoha smanji, rezultati dosta variraju. Ukoliko se broj osoba poveća, rezultati su lošiji i nestabilniji. Za ovu bazu, rezultati su dosta lošiji. Razlog tome je što se u ovoj bazi nalaze slike osoba iz Azije, a njihovi delovi lica imaju sličnu dužinu.



Slika 8: Matrica konfuzije

5 Zaključak

Literatura

[1] Overview of artificial neural network technologies, author = Tin Krambeger, Bojan Nožica, Ivica Dodig, Davor Cafuta, year=2019.

[2] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[3] University of California San Diego. The Extended Yale Face Database B. on-line at: http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html/.

[4] National Institute of Standards and Technology. Color FERET Database . on-line at: https://www.nist.gov/itl/iad/image-group/color-feret-database/.

[5] Department of Computer Science Robotics Laboratory and National Cheng Kung University Information Engineering. Databases for Face Detection and Pose Estimation. on-line at: http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm/.

[6] Matthew D. Zeiler. AdaDelta: An adaptive learning rate method, 2012.