

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

Sveučilišni studij

**PREPOZNAVANJE LICA POMOĆU MODELA
PRETRENIRANE DUBOKE NEURONSKE MREŽE**

Obrada slike i računalni vid

Ivan Jakab

Osijek, 2019. godina

1. UVOD

Prepoznavanje lica jedno je od složenijih, ali i istraživanih problema u računarstvu danas. Praktične primjene su jako velike, od potvrde identiteta bez unosa lozinki, pa do sigurnosnih sustava. No, problematika je poprilično složena, jer se slike mogu slikati iz različitih kutova, ljudi mogu promijeniti šminku ili kosu, ili imati dodatak kao sunčane naočale i sl. Zbog ovoga bi ovaj problem bio gotovo nemoguć za riješiti bez neuronskih mreža.

U seminaru se prikazuje mogućnost prepoznavanja lica pomoću modela istrenirane neuronske mreže. Izrađen je program koji uspoređuje sliku s nekog dokumenta, poput osobne ili vozačke dozvole, sa trenutno uslikanom slikom. Za te dvije slike, program će pokušati utvrditi jeli prikazana ista osoba.

2. PREGLED PODRUČJA

Duboke neuronske mreže jedne su od najpopularniji tehnika na polju strojnog učenja. Osobito su korisne na poljima obrade slike, klasifikacije objekata na slikama, robotskom vidu, prepoznavanju govornog jezika te prepoznavanju lica.

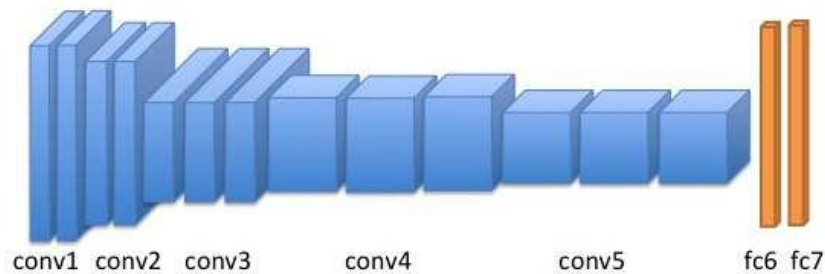
¹Neke od najpoznatijih kompanija koje se bave ovim područjem su Google's Deep Mind i OpenAI. Osim njihovih poznatih projekata u kojima mreže pobjeđuju najbolje ljudske igrače u računalnim igrama, razvij (Github gist, 2020) (Researchgate, n.d.)aju i mreže za gore navedena područja. Vjerojatno najbolja mreža za prepoznavanje lica je Google's FaceNet, koji je treniran i testiran na stotinama milijuna slika i ima preko 99 postotnu točnost.

Za samu izradu neuronskih mreža trenutno je najpopularniji jezik python, sa svojom bibliotekom TensorFlow. Ona olakšava izradu neuronskih mreža, te omogućava treniranje istih na grafičkoj kartici (GPU), što znatno ubrzava proces učenja. Osim GPU, velike kompanije danas imaju FPGA sklopove izrađene posebno za TensorFlow (npr. TPU – tensor processing unit). Često je korištena biblioteka Keras, koja daje dodatnu razinu apstrakcije iznad TensorFlowa, te omogućuje još lakšu izradu neuronskih mreža.

U ovome seminaru koristi se mreža VGGFace2. To je konvolucijska neuronska mreža, izrađena na sveučilištu u Oxfordu. Jednostavna je za koristiti, te se može koristiti u Kerasu. Koristi se kako bi dobili vektore značajki lica – koji se kasnije mogu usporediti sa vektorom

¹ (Balaban, 2019)

značajki sa druge slike. ²Sastoji se od ulaznog sloja (ulazna slika), 5 konvolucijskih slojeva sa povećavajućim brojem neurona (64, 128, 256, 512, 512) te dva sloja na kraju sa 4096 neurona. Na cijelom modelu koristi popularnu *rectifier* aktivacijsku funkciju.



Slika 1 Prikaz VggFace2 mreže

3. OPIS ZADATKA

³Prepoznavanje lica može se podijeliti na dva zadatka: verifikacija i identifikacija. Verifikacija je 1-1 usporedba lica s poznatim identitetom, a odgovara na pitanje jeli to ta osoba? Identifikacija je 1-više mapiranje lica s bazom poznatih lica, a odgovara na pitanje koja je to osoba?

U ovome se seminaru radi verifikacija. U jednoj se mapi nalaze numerirane slike sa dokumenata, a u drugoj mapi njihovi parovi, tj. slike za koje program treba utvrditi nalazi li se na njima ista osoba kao i na dokumentu. Cilj je pomoću trenirane mreže izvući vektor značajki sa obje slike, te zatim odrediti radi li se o istoj osobi.

Program iterativno prolazi kroz N zadanih parova slika. Svaku sliku učitava sa diska pomoću matplotlib biblioteke, te obrađuje predobradu. Prvi dio predobradbenog postupka je izdvajanje lica sa slike. Ovo se obrađuje pomoću druge istrenirane mreže, MTCNN, koja nam omogućuje da lako izdvojimo pravokutnik u kojem se nalazi neko lice. Nakon izdvajanja lica, drugi je dio skaliranje slike na veličinu koju prihvaća VGGFace. U ovom je projektu određena ulazna veličina slike 250x250 i 3 kanala.

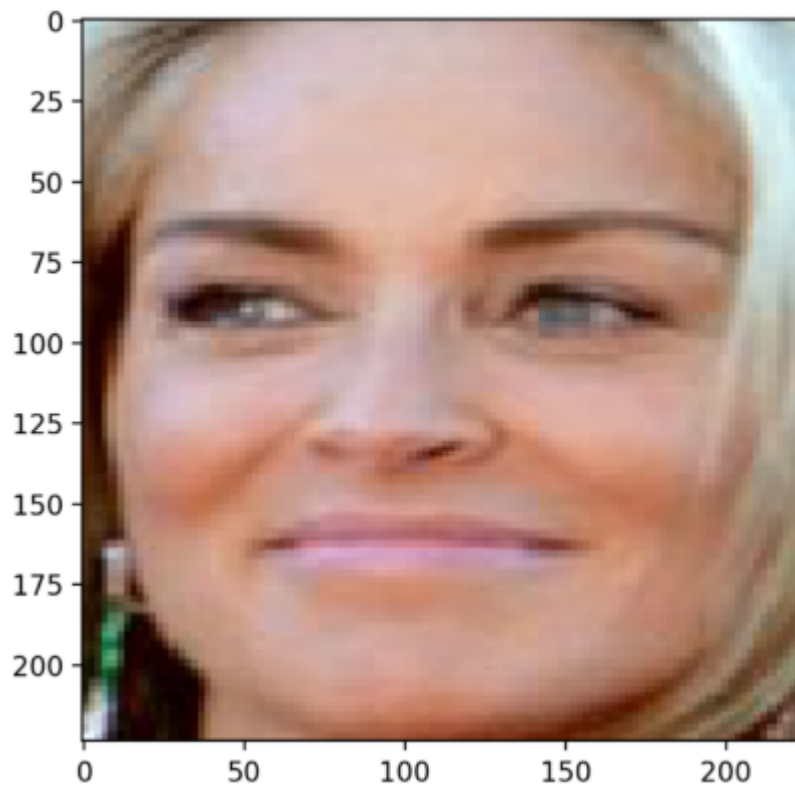
Na sljedećim slikama možemo vidjeti primjer predobrade jedne slike – izdvojeno lice i skalirano na dimenzije 250x250.

² (Github gist, 2020)

³ (Brownlee, 2019)



Slika 2 Sharon Stone, wikipedia



Slika 3 Nakon predobrade, (Brownlee, 2019)

Nakon predobrade, takva slika dolazi na ulaz u VGGFace mreže. Kao svoj izlaz, mreža daje vektor značajki lica. Ovaj vektor je ključan za verifikaciju lica. Na mrežu se iterativno dovode u parovima slike sa dokumenata i trenutne slike. Iz obje slike se izvuče vektor značajki.

Kada su dobiveni vektori značajki para slika, uzimaju se dvije metrike za izračunavanje udaljenosti tih vektora. Jedna je kosinusova udaljenost, a druga euklidska. Kao „konačan odgovor“ može se uzeti neka granica udaljenosti po bilo kojoj metrici. U ovome radu uzeta je jednostavna logika, ako je kosinusova udaljenost manja od 0.5, radi se o istoj osobi. Povećanjem ove granice dobili bismo rezultate koji su „popustljiviji“, a smanjenjem one koji su „stroži“. No, za podatke u seminaru se ovo pokazalo dobrom granicom te je sve točno predviđeno – osim jednog slučaja kada je slika mutna i osoba se promijenila u vremenu između slikanja dvije slike.

⁴Kosinusova udaljenost mjeri je sličnosti dva vektora. Dva vektora jednake orijentacije imaju sličnost 1, a suprotne -1. Dva okomita vektora imaju sličnost 0. Dana je sljedećom formulom

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

⁵Euklidska udaljenost je ordinalna udaljenost dvije točke u euklidskom prostoru. Poziciju neke točke možemo predstaviti euklidskim vektorom (početak u ishodištu i kraj u točki). Ako su vektori značajki predstavljeni kao euklidski vektori, onda je euklidska udaljenost tih vektora zapravo euklidska udaljenost krajnjih točki tih vektora. Euklidska udaljenost između dvije točke p i q je dana formulom

$$\begin{aligned} d(\mathbf{p}, \mathbf{q}) &= d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}. \end{aligned}$$

⁴ (Wikipedia, n.d.)

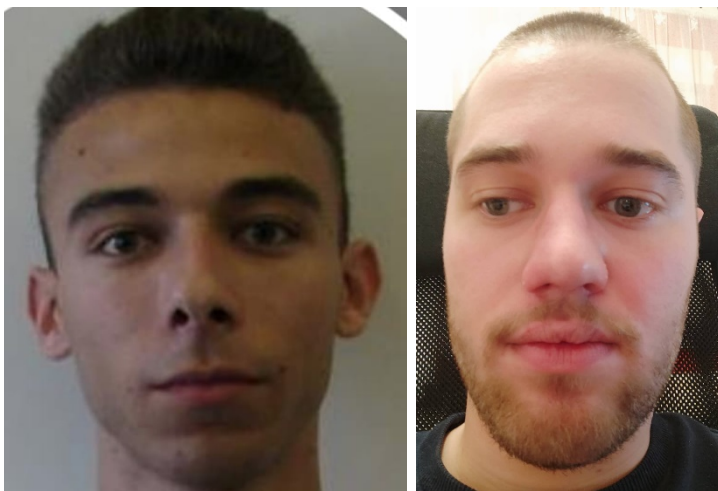
⁵ (Wikipedia, n.d.)

Budući da je teško doći do službenih dokumenata, set slika na kojima se program testirao je malen. U sljedećem primjeru, program je ispravno zaključio da se radi o istoj osobi, unatoč starenju i sunčanim naočalama.

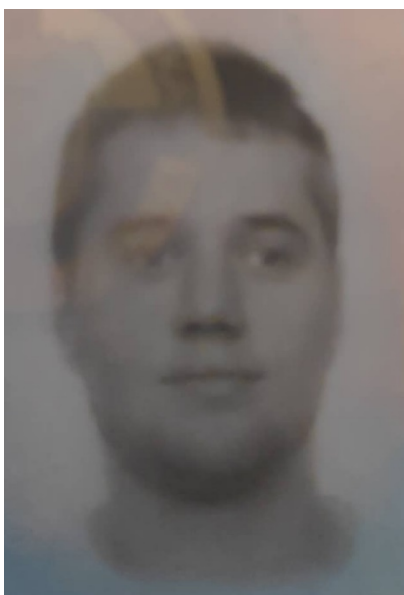


(Wikipedia, n.d.)

Za sljedeći par slika točno pretpostavlja da se ne radi o istoj osobi.



U sljedećem primjeru program netočno pretpostavlja da ne radi o istoj osobi. No, ovo je poprilično težak zadatak, čak i za čovjeka. Zanimljivo, kosinusova udaljenost značajki ovih slika je slična onoj iz prethodnog para.



4. ZAKLJUČAK

Prepoznavanje lica jedno je od najatraktivnijih i najizazovnijih problema u strojnom učenju, ali i u računarstvu općenito. Najčešće se rješava pomoću dubokih konvolucijskih neuronskih mreža. Posljednjih su godina doživjele značajan pomak u primjeni i točnosti. U seminaru je testirana mreža VGGFace2. Pomoću nje su dobiveni vektori značajki lica, a s tim vektorima je napravljen program za verifikaciju lica sa iznenađujućom točnošću.

5. LITERATURA

Balaban, S. (10. 2 2019). Dohvaćeno iz <https://arxiv.org/abs/1902.03524>

Brownlee, J. (5. 6 2019). *Machine Learning Mastery*. Dohvaćeno iz How to Perform Face Recognition With VGGFace2 in Keras: <https://machinelearningmastery.com/how-to-perform-face-recognition-with-vggface2-convolutional-neural-network-in-keras/>

Wikipedia. (n.d.). Dohvaćeno iz https://en.wikipedia.org/wiki/Cosine_similarity

Wikipedia. (n.d.). Dohvaćeno iz https://en.wikipedia.org/wiki/Euclidean_distance

Researchgate. (n.d.). Dohvaćeno iz https://www.researchgate.net/figure/VGG-Face-network-architecture_fig2_319284653

Github *gist*. (2020). Dohvaćeno iz <https://gist.github.com/EncodeTS/6bbe8cb8bebad7a672f0d872561782d9>

VGGFace službene stranice, https://www.robots.ox.ac.uk/~vgg/software/vgg_face/

MTCNN dokumentacija, Github, <https://github.com/ipazc/mtcnn>

6. POPIS PROGRAMA

Izvorni kod programa javno je dostupan na GitHub-u, na poveznici <https://github.com/ijakab/face-verification>. U vrijeme pisanja seminara, posljednja verzija pythona za koju je tensorflow podržan je 3.6, pa je kod potrebno pokrenuti sa pythonom 3, ne novijoj od 3.6.

Također, potrebno je instalirati biblioteke koje se koriste, ukoliko već nisu instalirane. Za instalaciju za konkretnu verziju pythona, može se koristiti naredba *python3.6 -m pip install <ime>*. Korištene biblioteke su matplotlib, numpy, tensorflow, keras, keras_vggface, scipy, mtcnn. Ostale bi trebale doći s distribucijom pythona.

Nove se slike mogu numerirano dodavati u mape *current* i *documents*, te je potrebno izmijeniti konstantu `NUMBER_OF_IMAGES`. Posljednje, potrebno je izmijeniti konstantu `BASE_PATH` na apsolutan put do projekta na računalu na kojem se pokreće, kako bi sve radilo ispravno.