Sustav za provjeru identiteta osobe

Juroš, Josip

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

Sveučilišni studij

SUSTAV ZA PROVJERU IDENTITETA OSOBE

Diplomski rad

Josip Juroš

Osijek, 2021.

SADRŽAJ

1. UVOD

U počecima razvoja, sustavi baziranih na umjetnoj inteligenciji mogli su brzo pratiti i riješiti probleme koji su, iako intelektualno zahtjevni čovjeku, mogli opisati formalno kroz matematička pravila te bili računalu relativno jednostavni za riješiti. Pravi izazov umjetnoj inteligenciji je rješavanje problema koji su jednostavni za ljude, ali ih je teško formalno opisati, problemi koje ljudi rješavaju intuitivno ili automatski kao sto su prepoznavanje riječi iz glasa ili prepoznavanje lica na slici. Jedan od najstarijih uspjeha umjetne inteligencije je IBM-ov „Deep Blue“ računalo koje je pobijedilo svjetskog prvaka Garry-a Kasparov-a u šahu [1]. Šah je vrlo jednostavna igra i okruženje za računalo, sastoji se od 64 lokacije i 32 figurice koje se kreću na vrlo lako programski opisiv način. Šah se može lako opisati na vrlo kratak način kroz skup pravila koja se prilože računalu od strane programera prije početka natjecanja. Ironično je kako apstraktni zadaci koji su najteže rješivi za čovjeka su među najlakšima za računalo. Računala već dugo mogu pobijediti čak i najboljeg ljudskog igrača no tek u donedavno su stekla sposobnosti slične čovjekovim kod prepoznavanja zvuka i lica. U ovom radu čitatelju će biti objašnjena jedna od mnogih primjena umjetne inteligencije, prepoznavanje lica u svrhu potvrde identiteta.

Biometrijska sigurnost igra sve veću i značajniju ulogu u modernoj sigurnosti, te administrativnim i poslovnim sustavima. Biometrija uključuje otiske prstiju, skeniranje mrežnice, identifikaciju glasa i prepoznavanje lica. Prepoznavanje lica je privuklo posebnu pažnju, jer pruža diskretno, nenametljivo sredstvo za detektiranje, identifikaciju i provjeru, bez potrebe za znanjem i pristankom subjekta. U današnjem svijetu takvi sustavi se koriste na zračnim lukama te su prihvaćeni od mnogih državne službe u svrhu provođenja zakona. Jedan od novijih i uspješnijih primjera toga je slučaj u Kini, gdje je policija pomoću tehnologija za prepoznavanje lica identificirala i pratila traženog zločinca na koncertu koji je pohađalo 60000 ljudi [2].

Kao posljedica dokaza da sustavi temeljeni na umjetnoj inteligenciji mogu nadjačati ljude kod problema prepoznavanja lica, projicira se da će državni sektor biti najveći korisnik biometrijskih sustava, primjerice prepoznavanja lica, preko nadolazećih deset godina [3]. Tehnologije za prepoznavanje lica nisu isključivo namijenjene za državni sektor, imaju ogromne primjene i u potrošačkom sektoru, kroz mnoge korisničke aplikacije i uređaje, od potvrde identiteta za omogućavanje pristupa, do mnogih primjena u društvenim mrežama. Korisnicima i industriji su potrebne brze, pristupačnije i efektivne aplikacije koje

zadovoljavaju njihove zahtjeve u poslovanju, zapošljavanju, edukaciji, koje se mogu proširiti na sigurnost i smanjenje administrativnih troškova.

Mnogi slučajevi se mogu naći gdje se dokument osobe i njegovo lice uspoređuju, no to se većinom uvijek vrši od strane čovjeka. Prema jednom istraživanju iz Australije trenirani dužnosnici su imali 14% stopu pogreške kod sparivanja lica i slike s dokumenta ljudi na zračnoj luci. Zbog ovakvih i mnogih drugih slučajeva potrebna je implementacija sustava koji će automatizirati proces prepoznavanja [4].

Funkcionalnost sustava koji će biti opisan u ovom radu je verifikacija lica, točnije sustav će kao izlaz dati odgovor na pitanje „Nalazi li se ista osoba na ove dvije slike?“. Svrha ovakve verifikacije, potvrde identitet, je za sigurnosne razloge. Na primjer, potvrda identiteta pri registraciji na neku web stranicu koja to zahtjeva.

2. POSTOJEĆE METODE DETEKCIJE I VERIFIKACIJE LICA

Detekcija lica je temeljni korak u verifikaciji lica. Također se proteže na širok spektar drugih aplikacija koje su navedene u uvodu ovog rada. Povijesno, najveća prepreka kod algoritama detekcije lica je otežano postizanje visoke preciznosti u nekontroliranim uvjetima. Kao posljedica toga, primjene u svijetu su im bile ograničene. Nakon razvoja „*Viola Jones*“ metode za detekciju lica, broj primjena detekcije lica u svijetu je naglo skočio [5]. Za područje detekcije lica, konvolucijske neuronske mreže pokazale su se najboljim alatom za rješavanje problema. Za najveći napredak u razvoju područja detekcije lica zaslužne su masovne količine podataka i grafičke kartice korištene za treniranje modela temeljenih na dubokom učenju [6, 7]. Kontinuiranim napretkom algoritama detekcije lica, najviše su primijenjeni od strane sigurnosnih službi u forenzici, primjerice u automatskom detektiranju sadržaja sa seksualnim iskorištavanjem djece (engl. Child Sexual Exploitation Material – CSEM) te daju podršku u identificiranju žrtava, samim time služe u povezivanju više slučajeva takvog tipa [8].

2.1. Tradicionalne metode detekcije lica

Jedan od prvih načina rješavanja problema detekcije lica je primjena lokalnih binarnih uzoraka (engl. Local Binary Patterns – LBP) kako bi se poboljšale diskriminacijske značajke lica [9]. Prema [10], značajke za prepoznavanje emocija su dobivene primjenom entropije stacionarnih *waveleta*, te je jednoslojna unaprijedna neuronska mreža upotrijebljena kao klasifikator. Štoviše, Jaya algoritam uveden je kako bi se spriječilo da obuka klasifikatora padne u lokalne točke optimuma. U [11] je predložen Associate Predict model koji se bavi sličnostima ljudskog lica pri promjeni poze, osvjetljenja i izraza lica, dok je u [12] predložena metoda višedimenzionalnog skaliranja (engl. Multi-dimensional Scaling – MDS) za učenje matrice preslikavanja koja projicira sliku lica visoke i niske rezolucija na zajednički podprostor. Prethodno navedene metode služe za dodavanje međurazrednog ograničenja kako bi se povećala udaljenost različitih subjekata u podprostoru te bi se time osigurala diskriminativnost.

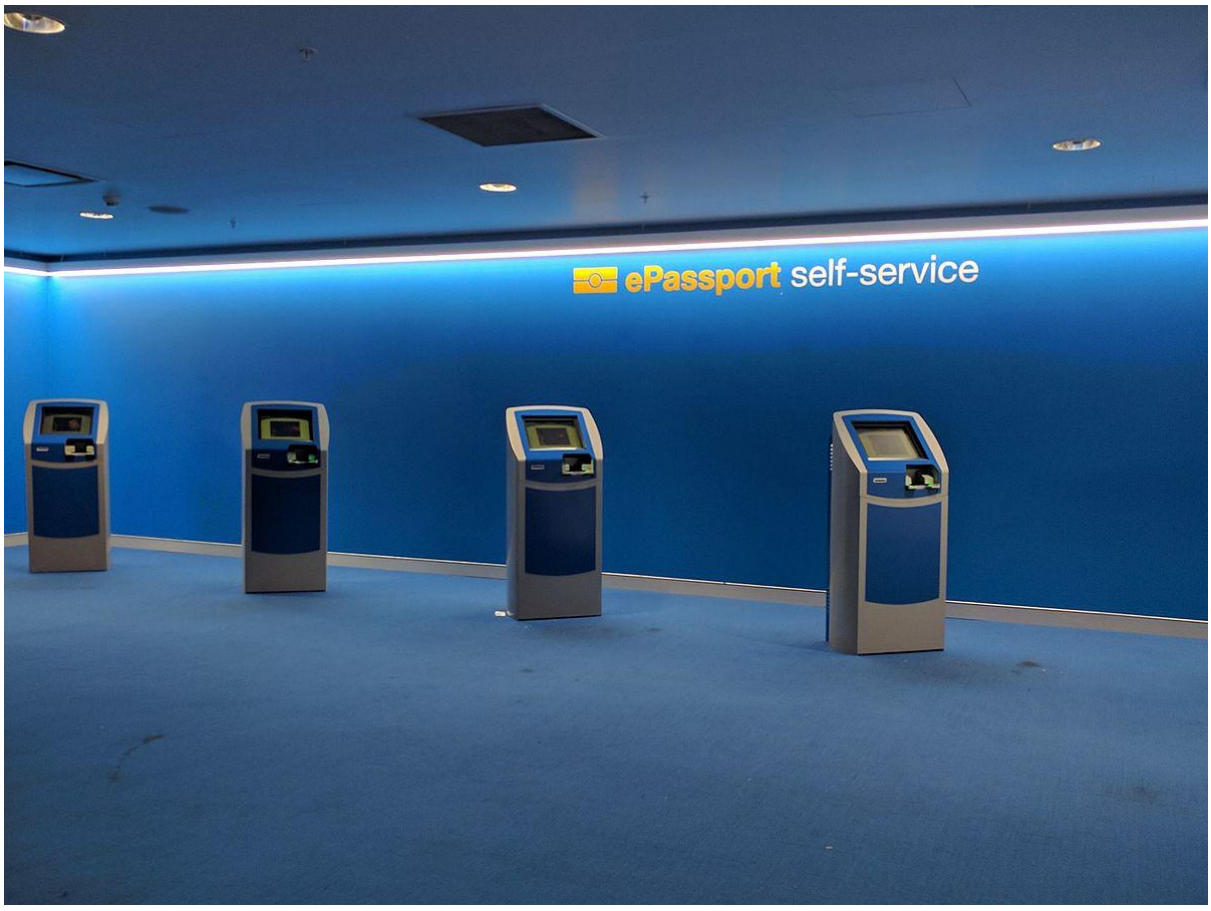
2.2. Metode temeljene na dubokom učenju

Metode temeljene na dubokom učenju, primjerice FaceNet-a [13], VGGFace-a [14], postale su mnogo popularnije u odnosu na metode spomenute u prethodnom potpoglavlju zato što se mogu nositi s velikim skupovima podataka pri učenju bogatom i kompaktnog prikaza lica. Također je uvedena hibridna metoda koja spaja učenje značajki od strane konvolucijske neuronske mreže i značajke dobivene od strane tradicionalnih modela upotrebom vreću riječi

(engl. Bag of Words – BOW)[15]. Zbog ekonomičnosti i manjih hardverskih zahtjeva uveden je model koji se bavi detekcijom lica na slikama niže rezolucije koji se zove ResNet [16].

2.3. Postojeće implementacije

Mnogo sustava koji vrše potvrdu identiteta osobe postoje i ponajviše se koriste na graničnim prijelazima i zračnim lukama. Prvi među njima je „*Smart-Gate*“ u Australiji (slika 2.1.), implementiran 2007. godine [17]. Zbog sve veće količine putnika koji dolaze u Australiju, država je implementirala „*Smart-Gate*“ na većinu svojih zračnih luka kao formu elektronske provjere putovnica za osobe koje su u to vrijeme imale e-putovnice. Sustav je implementiran fizički kao automat, gdje bi korisnici unijeli svoj dokument u automat, dopustili kameri na automatu da im uslika lice, te nakon izvršene potvrde identiteta, osobama je omogućen ulaz u državu. Slični sustavi su postavljeni u Ujedinjenom Kraljevstvu (slika 2.2), Sjedinjenim Američkim Državama (slika 2.3.) i ostalim državama [18]. Osim sustava za nacionalnu sigurnost, jedna od implementacija provjere identiteta je za mrežne bankarske aplikacije kao na primjer *Revolut,* koja posjeduje postupak provjere identiteta [19].



Slika 2.5. „*SmartGate*“ sustav u Australiji [18]

Slika 2.6. „*ePassport gates*“ u Ujedinjenom Kraljevstvu [20]

Slika 2.7. Automatska kontrola putovnica u Sjedinjenim Američkim Državama [21]

3. TEORIJSKA POZADINA PRIMIJENJENIH TEHNOLOGIJA

U ovom je poglavlju objašnjena teorijska pozadina tehnologija primijenjenih u provedbi eksperimenta i izradi korisničkog rješenja sustava.

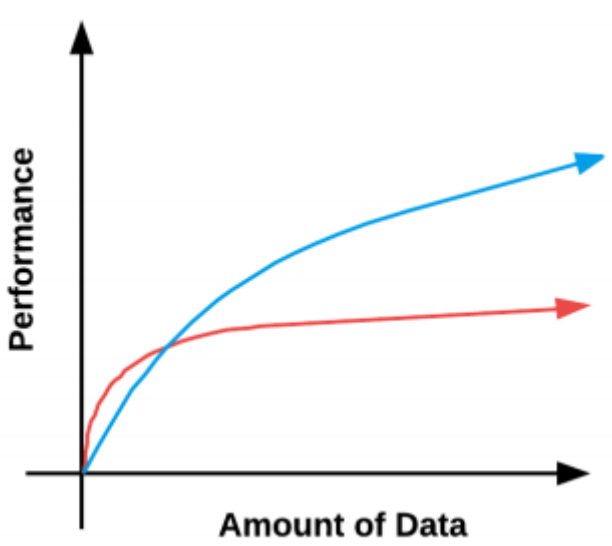
3.1. Duboko učenje

Duboko učenje je podpolje strojnog učenja, koje je podpolje umjetne inteligencije. Glavni cilj umjetne inteligencije je pružiti skup algoritama i tehnika koje se mogu koristiti za rješavanje problema koje ljudi intuitivno i automatski izvode, ali predstavljaju velik izazov računalima. Izvrstan primjer takve klase problema je tumačenje i razumijevanje sadržaja slike, ono što čovjek može učiniti bez imalo truda, strojevima je izuzetno teško kao što je spomenuto u uvodu. Odnos između umjetne inteligencije, strojnog učenja i dubokog učenja je prikazan Venn-ovim dijagrama na slici 3.1.

Slika 3.1. Venn dijagram koji opisuje odnose umjetne inteligencije i njezinih podpolja

Umjetna inteligencija obuhvaća širok i raznolik skup poslova koji se odnose na automatski strojno zaključivanje. Podpolje strojnog učenja specijalizirano je za prepoznavanje

uzoraka i učenje iz podataka. Kada se spominje pojam duboko učenje, odnosi se na neuronske mreže s velikim brojem slojeva. Duboko je učenje zapravo jako širok pojam jer ne postoji općeprihvaćen broj slojeva koji neuronska mreža mora imati kako bi se ta mreža mogla klasificirati kao duboka neuronska mreža. Za razliku od tradicionalnog strojnog učenja koje uzima skup slika i primjenjuje ručno definirane algoritme izdvajanja značajki te vrši treniranje klasifikatora na temelju tih značajki, duboko učenje bazira se na principu slaganja slojeva jednog za drugim koji automatski uče kompleksnije i apstraktnije značajke u odnosu na tradicionalno strojno učenje. Konvolucijska neuronska mreža, koja će detaljnije biti opisana u nastavku rada, je po definiciji tip algoritma dubokog učenja. Čak i konvolucijska neuronska mreža koja ima samo jedan sloj i dalje se može trenirati kao duboka neuronska mreža iako je po dizajnu jako plitka.



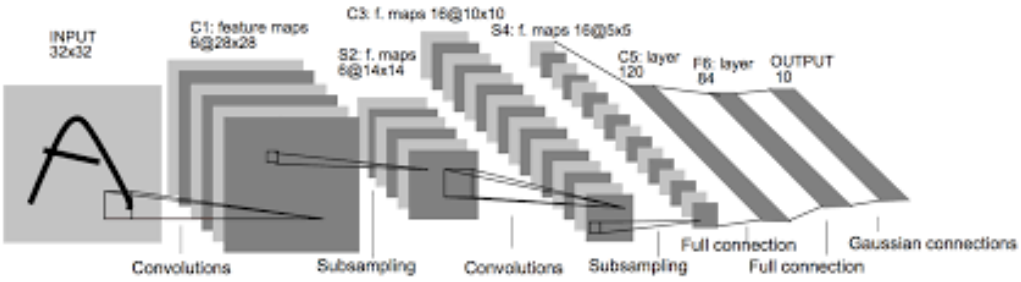
Slika 3.2. Graf ovisnosti performanse mreže o količini podataka. Plava linija je duboko učenje, crvena je tradicionalni klasifikator [22]

S povećanjem količine podataka za učenje, algoritmi neuronske mreže postižu veću klasifikacijsku preciznost. Zbog te povezanosti poboljšanja performanse i količine podataka, može se reći kako su za duboko učenje potrebni veliki skupovi podataka [22].

3.2. Konvolucijske neuronske mreže

Konvolucijska neuronska mreža (engl. Convolutional neural network - *CNN*) evolucija je višeslojne neuronske mreže. *CNN* se grade slaganjem niza slojeva gdje je svaki sloj odgovoran za određeni zadatak. Konvolucijski slojevi uče skup konvolucijskih filtara.

Aktivacijski slojevi nanose se na vrhu konvolucijskih slojeva kako bi se dobila nelinearna transformacija. Spajanje slojeva pomaže u smanjenju prostornih dimenzija ulaznog volumena dok prolazi kroz mrežu. Jednom kada je ulazni volumen dovoljno mali, potpuno povezani slojevi primjenjuju se kao posljednji slojevi za konačno predviđanje izlaza.



Slika 2.3. Struktura konvolucijske neuronske mreže [23]

U praksi *CNN* nude dvije ključne prednosti: lokalnu nepromjenjivost i kompoziciju. Lokalna invarijantnost omogućuje nam da klasificiramo sliku koja sadrži određeni objekt bez obzira na to gdje se objekt pojavljuje na slici. To se može dobiti korištenjem slojeva za udruživanje, koji identificiraju područja ulaznog volumena s visokim odzivom na određeni filtar. Druga je korist kompozicija. Svaki filter sastavlja lokalnu zakrpu značajki niže razine u prikaz više razine. Ovaj sastav omogućuje mreži da dublje u mreži nauči bogatije značajke. Na primjer, mreža može graditi rubove od piksela, oblike od rubova, a zatim složene objekte od oblika - i to sve na automatiziran način koji se prirodno događa tijekom procesa treninga. Koncept izgradnje značajki više razine od nižih razina upravo je razlog zašto su CNN-ovi tako snažni u računalnom vidu.

3.3. DLIB

Provjera identiteta osobe nije moguća bez prethodno provedene detekcije lica. Dlib moderni je C++ alat koji sadrži algoritme strojnog učenja, detekcije lica i mnoge druge [24]. Dlib nudi nekoliko načina detekcije lica koji se baziraju na histogramu orijentiranih gradijenata (engl. Histogram of Oriented Gradients – HOG) i konvolucijskim neuronskim mrežama. Glavne razlike u postupku detekcije lica je sposobnost konvolucijskih neuronskih mreža da bolje detektiraju lice kada lice nije slikano frontalno [25].

4. PRIMIJENJENE TEHNOLOGIJE

U ovom poglavlju će biti opisani programski jezici, okruženja i biblioteke upotrijebljene za provođenje eksperimenta i izradu korisničkog rješenja.

4.1. Python

Python je programski jezik opće namjene na visokoj razini. Python-ova filozofija dizajna naglašava čitljivost koda značajnom uporabom tabeliranja. Njegove jezične konstrukcije kao i objektno orijentirani pristup imaju za cilj pomoći programerima da napišu jasan, logičan kôd za male i velike projekte. Podržava više programskih paradigmi, uključujući strukturirano (posebno proceduralno), objektno orijentirano i funkcionalno programiranje. Također je vrlo pogodan za mnoge projekte zbog ekstenzivne količine biblioteka koje su na raspolaganju korisniku. Guido van Rossum počeo je raditi na Pythonu krajem 1980-ih, kao nasljednik programskog jezika ABC, a prvi ga je objavio 1991. godine kao Python 0.9.0.

4.2. OpenCV

OpenCV (engl. Open Source Computer Vision Library) je biblioteka programskih funkcija koja je uglavnom usmjerena na računalni vid u stvarnom vremenu. Biblioteka funkcionira na više platformi i besplatna je za uporabu pod licencom otvorenog koda Apache 2. Počevši od 2011. godine, OpenCV ima sposobnost GPU ubrzanja za rad u stvarnom vremenu.

4.3. HTML

HTML (engl. HyperText Markup Language) je standardni opisni jezik za dizajniranje web aplikacija. Potpomognut je tehnologijama kao što su CSS (engl. Cascading Style Sheets) te podržava mnoge skriptne jezike kao što su JavaScript i PHP. Upotrijebljen je za izradu korisničkog rješenja, kako bi se korisniku ponudilo sučelje preko kojeg korisnik pristupa funkcionalnosti sustava.

5. EKSPERIMENTALNO ISTRAŽIVANJE

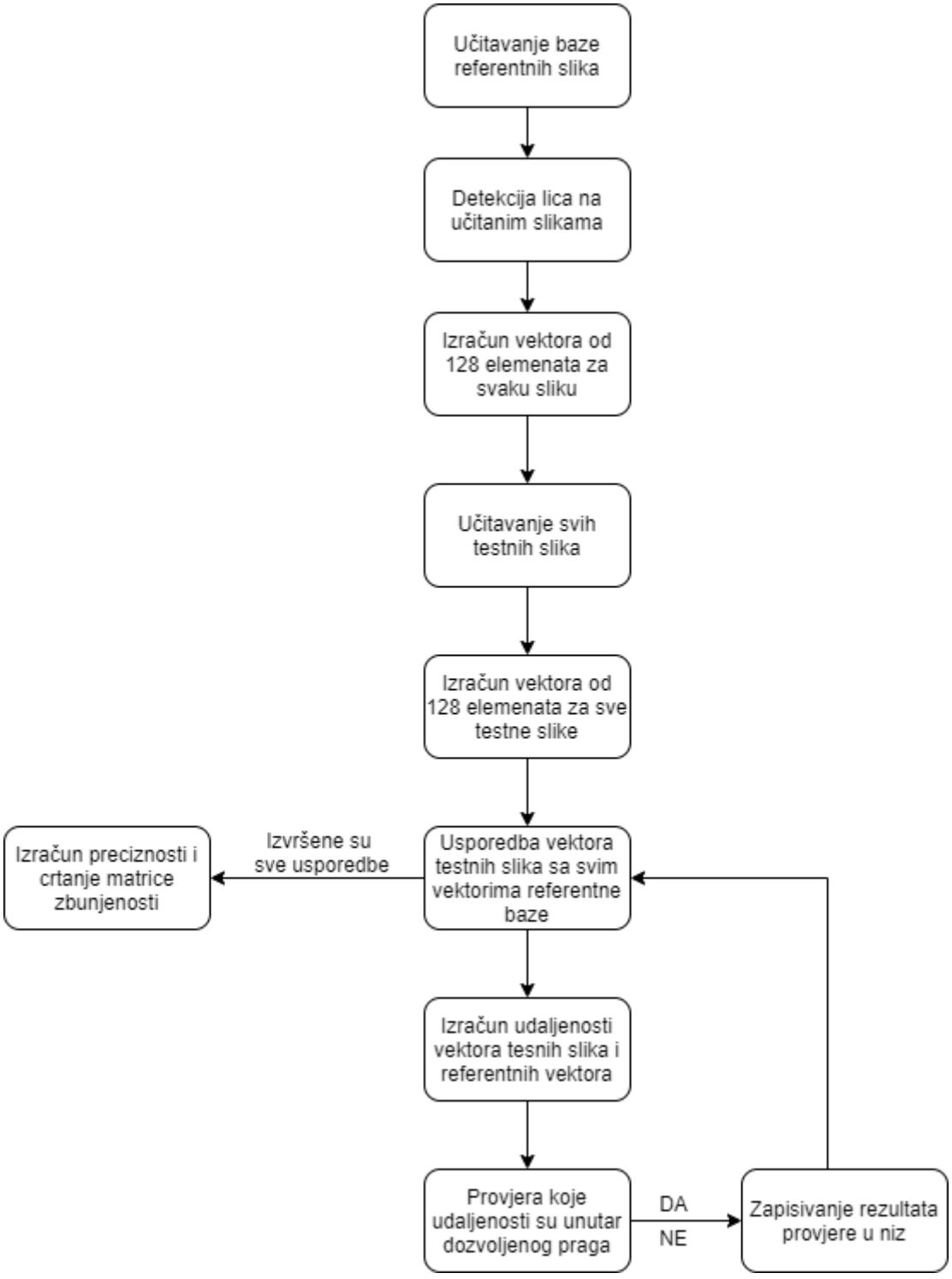
Konvolucijska neuronska mreža koja se koristi za prepoznavanje lica je već naučena na *LFW* skupu podataka te ima točnost od 99.83% pri detekciji lica [26]. Za provođenje pouzdanog eksperimenta potrebno je stvoriti set podataka pogodan za ovakav sustav potvrde identiteta. Skup podataka se sastoji od slika dokumenata koje su referentne slike, te fotografija ljudi u nekoliko poza i drukčijih osvjetljenja koje su testne slike (slika 4.1.). Python biblioteka upotrijebljena u provođenu eksperimenta i izradi korisničkog rješenja je „*face_recognition*“[27].

5.1. Tehnička provedba eksperimenta

U ovom podpoglavlju će biti objašnjen tijek eksperimenta pomoću dijagrama toka prema slici 4.2. Nad referentnim dijelom skupa podataka, slike dokumenata, provodi se detekcija lica, te se od detektiranih lica dobivaju vektori 128 elemenata koji jednoznačno opisuje lica, te se od njih dobiva referentna matrica. Zatim se isti proces ponavlja s testnim slikama i dobiva se matrica testnih vektora.

Svaki vektor testne matrice se uspoređuje sa svakim vektorom referentne matrice te se dobivaju udaljenosti. Udaljenosti predstavljaju razliku između vektora te se uspoređuju s definiranim pragom. Korišten je preporučeni prag prema [28], kako bi utvrdili nalazi li se na obje slike ista osoba. Dobivene su privolu od svih osoba čije su slike uključene u ovom radu.

Slika 4.1. Primjer referentne slike (lijevo), primjer testne slike (desno)



Slika 4.2. Dijagram toka eksperimenta

5.2. Prikaz rezultata

Kao rezultat eksperimenta dobivena je matrica zbunjenosti koja predstavlja performansu neuronske mreže na zadanom skupu podataka. Iz matrice zbunjenosti se mogu izračunati mnogi parametri iz kojih se može odrediti uspješnost eksperimenta kao što su točnost, učestalost pogrešne kvalifikacije, preciznost, odziv i specifičnost. Matrica zbunjenosti

se sastoji od 4 broja koje nazivamo, stvarni pozitiv (engl. True positive - TP), stvarni negativ (engl. True Negative - TN), lažni pozitiv (engl. False Positive - FP) i lažni negativ (engl. False Negative - FN).

$$točnost = \frac{TP + TN}{TP + TN + FP + FN} \tag{4-1}$$

$$učestalost\ pogreške = 1 - točnost \tag{4-2}$$

$$preciznost = \frac{TP}{TP + FP} \tag{4-3}$$

$$odziv = \frac{TP}{TP + FN} \tag{4-4}$$

$$specifičnost = \frac{TN}{TN + FP} \tag{4-5}$$

Na testnog skupu eksperimenta dobivena je sljedeća matrica zbunjenosti (Tablica 4.1.). Matrica zbunjenosti pokazuje granično zadovoljavajući rezultat. S točnošću koja iznosi 89.28 posto. Da bi klasifikator bio uspješan, potrebna je točnost od približno 92 posto do 95 posto.

Matrica zbunjenosti	***Negative***	***Positive***
False	0	0
True	6	50

Tablica 4.1. Matrica zbunjenost

Primjenom formula (4-1) do (4-5) dobiveni su sljedeći parametri:

$$točnost = 0{,}8928$$

$$učestalost\ pogreške = 0{,}1072$$

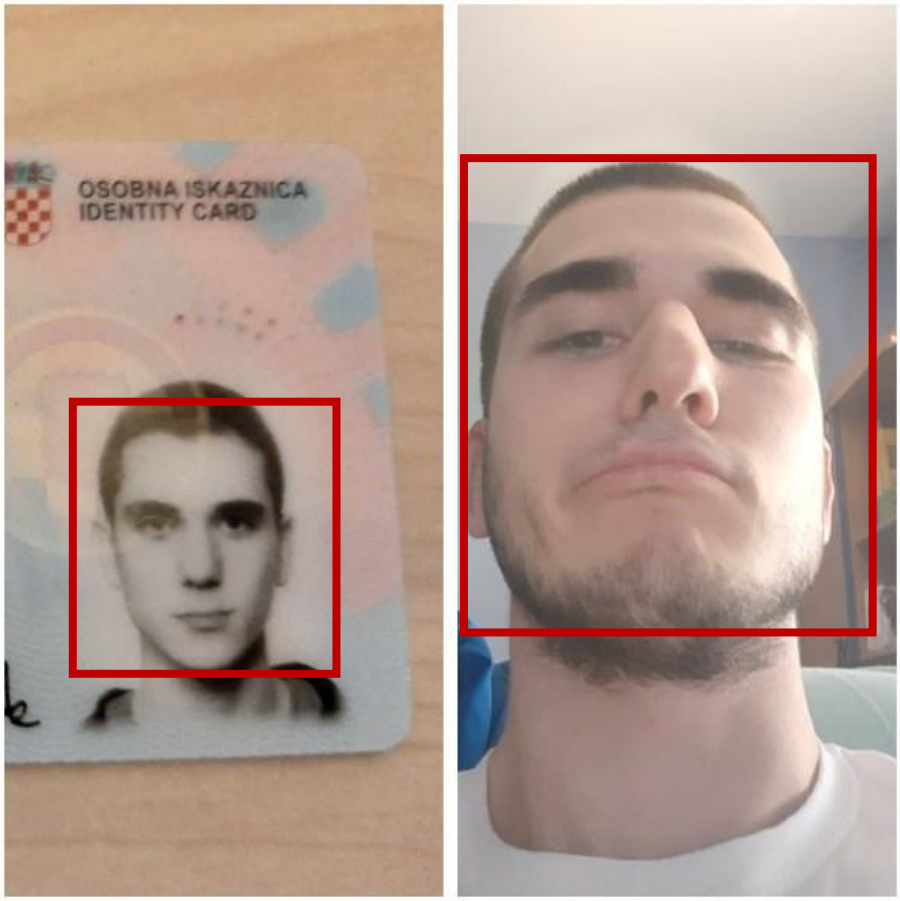
$$preciznost = 1$$

$$odziv = 0{,}8928$$

$$specifičnost = 0$$

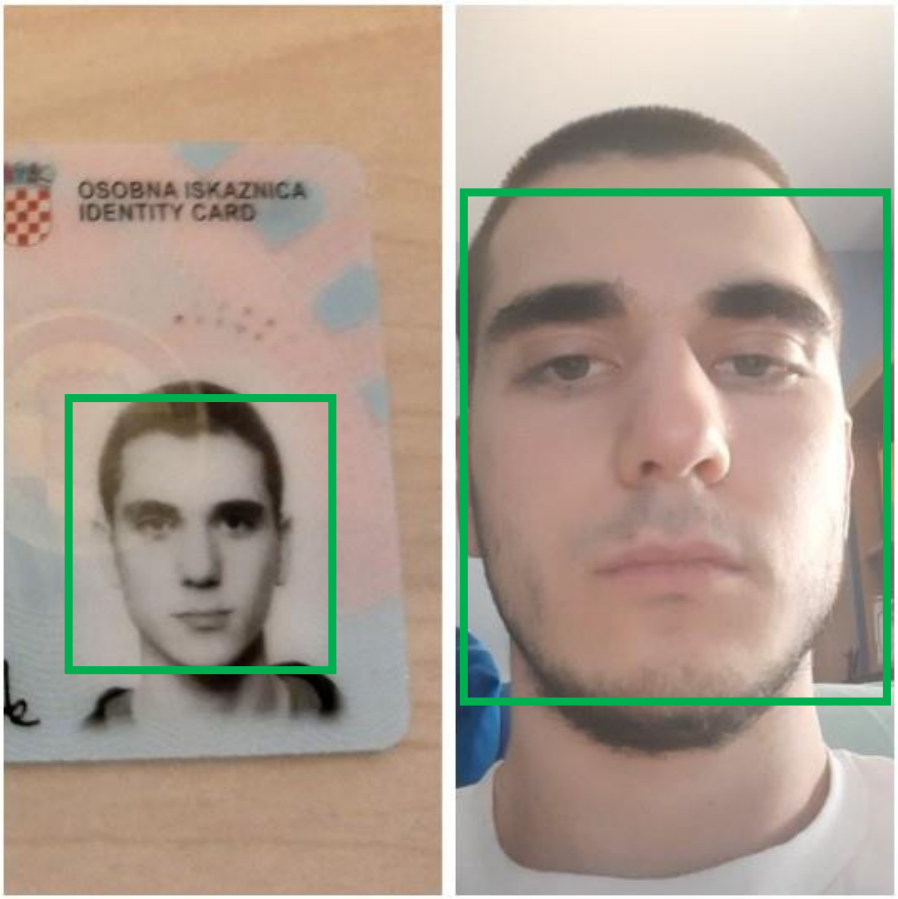
Glavni razlog dobivanja niže točnosti kao rezultat eksperimenta je nekoliko slika testnog skupa koje nemaju ozbiljan izraz lica, što je u slučaju sparivanja dokumenta i slike iznimno bitno. U idealnom slučaju lice na slici mora biti cijelo vidljivo, pozadina jednobojna i s dovoljnim kontrastom prema licu, lice mora biti oštro uslikano kako bi sve značajke bile definirane, lice ne smije biti prekriveno, na primjer naočalama ili kapama, te izraz lica mora biti neutralan i lice mora biti slikano frontalno, kao što je preporučeno prema [29]. Kako skup za učenje sadrži samo po jednu slike od svake osobe, s time da su te slike zapravo detektirane na dokumentima (koje su slikane prema uputama [29]), dok skup za testiranje sadrže slike koje nisu slikane prema uputama prema [29], može se očekivati da sustav za provjeru identiteta nije dovoljno robustan da prepozna istu osobu. Za točniji zaključak potrebno je ponovno provesti eksperiment sa znatno većim testnim skupom podataka.

U nastavku će biti navedeni primjeri uspješnog i neuspješnog sparivanja lica, kao što je prethodno navedeno, neutralno lice je najbitnije za uspješno sparivanje.



Slika 4.3. Primjer neuspješnog sparivanja lica

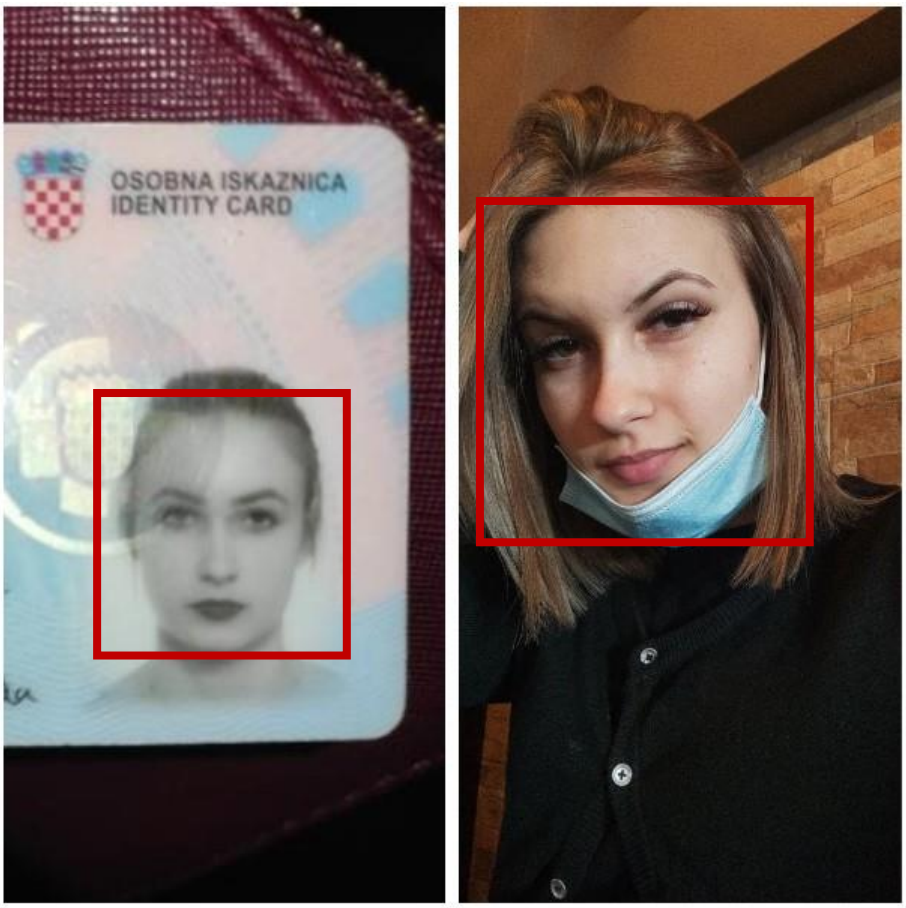
Prema slici 4.3. vidljivo je kako lice nije u skladu s uputama prethodno spomenutim, napravljena je grimasa, lice nije neutralno i nije uslikano frontalno. Kao rezultat sustav daje negativan odgovor.



Slika 4.4. Primjer uspješnog sparivanja lica

Prema slici 4.4. lice na slici je neutralno, nema grimase te je lice uslikano frontalno, zbog tih ispunjenih uvjeta, sustav vraća pozitivan odgovor za ovaj slučaj.

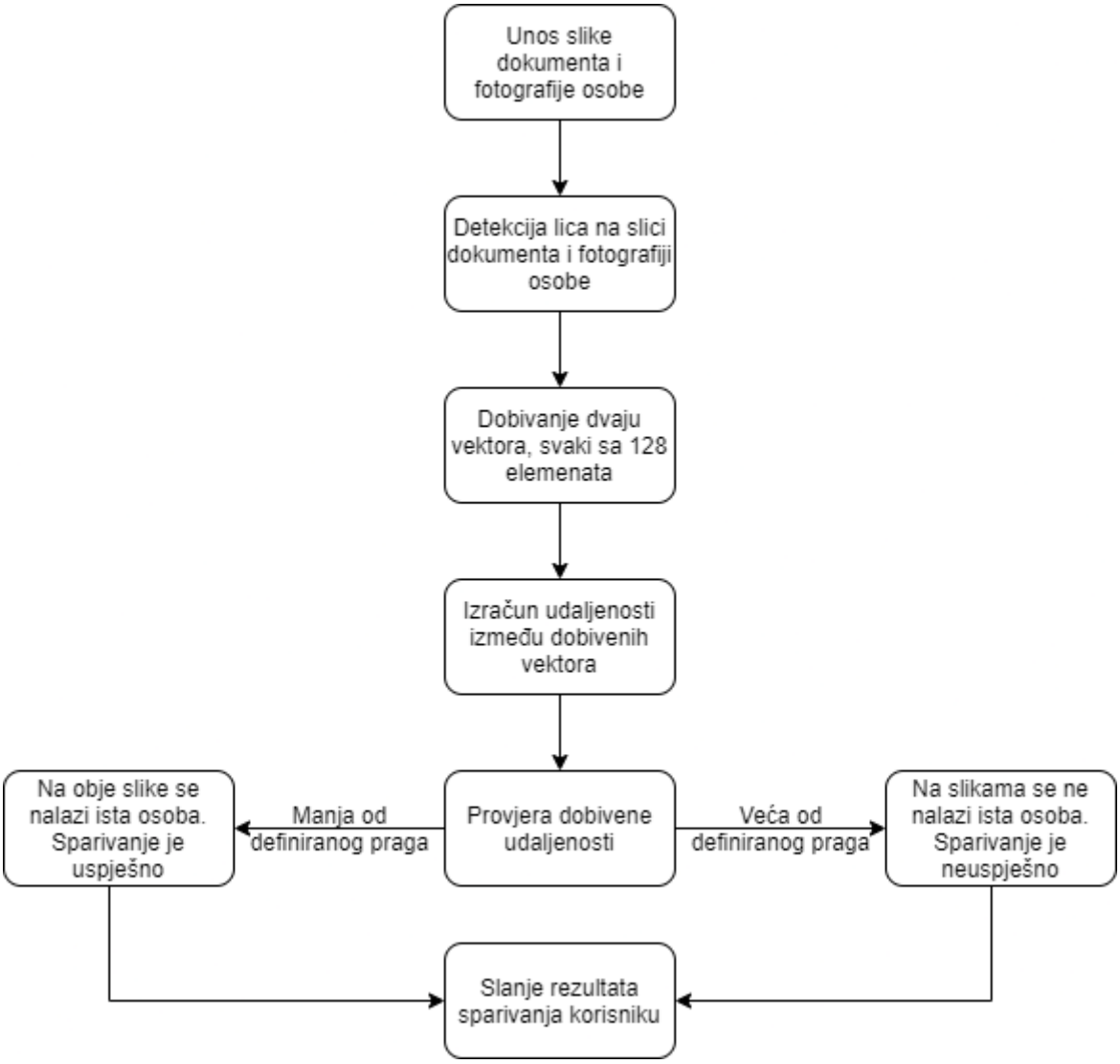
Također je bitno da lice ne bude prekriveno stvarno, tipa naočale, marame ili maske kao na slici 4.5. Uz već spomenuti uvjet da lice mora biti uslikano frontalno.



Slika 4.5. Primjer neuspješnog sparivanja

6. TEHNIČKA IMPLEMENTACIJA KORISNIČKOG RJEŠENJA

U ovom poglavlju će pomoći dijagrama toga biti opisana funkcionalnost korisničkog rješenja sustava za provjeru identiteta osobe.



Slika 5.1. Dijagram toka programa

Korisniku je ponuđeno sučelje pomoću kojeg učitava dvije slike, sliku dokumenta i sliku lica. Nakon potvrde unosa dvije slike, sustav izračunava vektor svakog lica, te ih uspoređuje. Ako je udaljenost između ta dva vektora ispod određenog praga, sustav potvrđuje identitet korisnika. Za realizaciju korisničkog rješenja odabrana je web aplikacija zbog pristupačnosti, jednostavnosti i mogućnosti ugnježđivanja u druge web aplikacije i sustave.

Povezivanje python koda i HTML koda ostvaren je pomoću biblioteke „*Flask*“. Ta biblioteka je klasificirana kao mikrookvir (engl. „*microframework*“) za što ne zahtijeva

posebne alate [30]. Dizajn stranice unesen je u glavni program kao predložak što je jedna od funkcionalnosti upotrijebljene biblioteke.

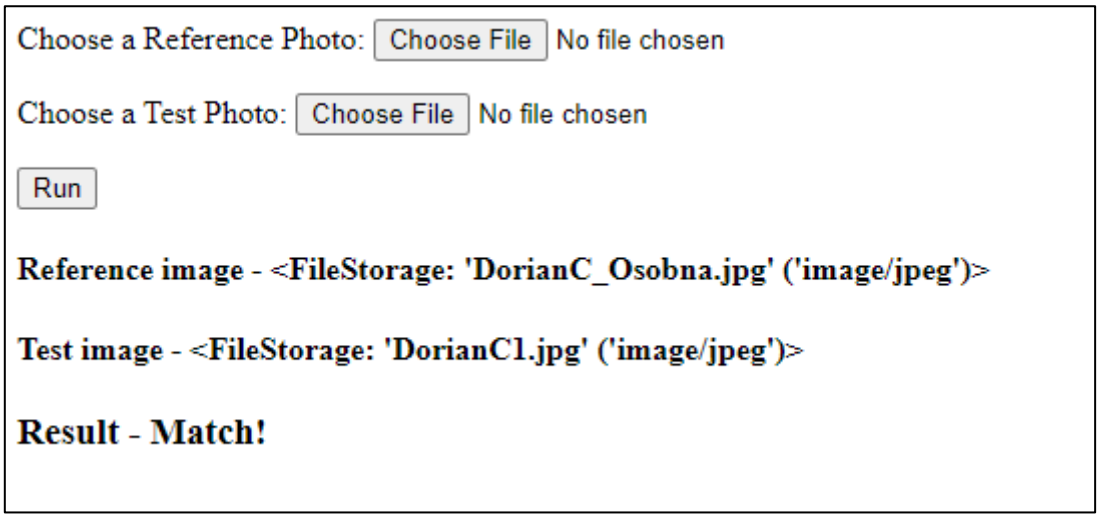
Dizajn web aplikacije sastoji se od forme za unos datoteka pomoću POST metode. Gumb okida proces usporedbe dvije unesene slike, te vraća korisniku rezultat.

6.1. Prikaz korisničkog rješenja

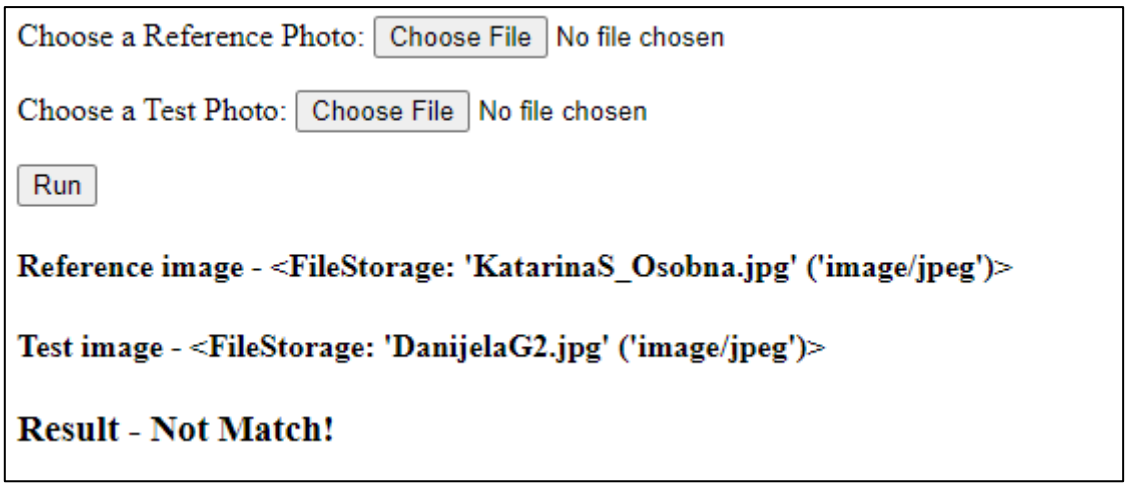
U ovom poglavlju će ukratko biti opisan izgled korisničkog rješenja te dva moguća rezultata koja sustav može vratiti korisniku. Gumb za odabir datoteke otvara tražilicu lokalno spremljenih datoteka.

Slika 6.1. Prikaz korisničkog rješenja pri prvom pokretanju aplikacije.

Kao rezultat, aplikacije vraća niz riječi koji označavaju provedena li je provjera identiteta uspješno ili neuspješno.



Slika 6.2. Prikaz korisničkog rješenja pri uspješnoj provjeri identiteta



Slika 6.3. Prikaz korisničkog rješenja za neuspješnu provjeru identiteta

7. ZAKLJUČAK

S povećanjem procesorske moći i daljnjem poboljšanju konvolucijskih neuronskih mreža, njihova primjena će eksponencijalno rasti, te u konačnici će biti zastupljena u gotovo svakom aspektu ljudskog života. Trenutno najveći interes je sigurnost i autorizacija, taj proces je opisan u ovom radu na bazi prepoznavanja i verifikacije lica. Zbog same prirode eksperimenta i zbog ljudske zaštite osobnih podataka, testni skup je bio poprilično mali, te je to glavni razlog za neuvjerljiv rezultat. Ono što se može potvrditi eksperimentom je kako je robusnost mreže na grimase i starost jako mala. Iznimno je bitno da testni skup podataka bude čist, potrebno je neutralno i frontalno uslikano lice, osim toga lice ne smije biti prekriveno drugim objektima kako bi se osigurala što veća efikasnost sustava pri procesu sparivanja.

8. LITERATURA

[1] F. Hsu, Behind Deep Blue: Building the Computer That Defeated the World Chess Champion, Princeton University Press, Princeton, 2002.

[2] N. Louise, Chinese authorities used Facial recognition AI to catch fugitive among 60,000 concert-goers in China, 2018, dostupno na: https://techstartups.com/2018/04/13/chinese-authorities-used-facial-recognition-ai-catch-fugitive-among-60000-concert-goers-china/ (14.9.2021.)

[3] PRNewswire, Global Market Study on Face and Voice Biometrics: Government Sector Projected to be the Most Attractive End Use Industry Segment During 2017 – 2025, 2017, dostupno na: https://www.prnewswire.com/news-releases/global-market-study-on-face-and-voice-biometrics-government-sector-projected-to-be-the-most-attractive-end-use-industry-segment-during-2017---2025-300500668.html (14.9.2021.)

[4] D. White, R. Kemp, R. Jenkins, M. Matheson, A. Burton, Passport officers errors in face matching, PloS one, 2014, doi:10.1371/journal.pone.0103510

[5] P. Viola, M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, 2001, doi:10.1109/CVPR.2001.990517

[6] J. Deng, J. Guo, S. Zafeiriou, N. Xue, ArcFace: Additive Angular Margin Loss for Deep Face Recognition, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4685-4694, 2019, doi:10.1109/CVPR.2019.00482

[7] B. Amos, B. Ludwiczuk, M. Satyanarayanan, OpenFace: A general-purpose face recognition library with mobile applications, Pittsburgh, PS, 2016.

[8] D. Chaves, E. Fidalgo, E. Alegre, F. Janez-Martino, R. Biswas, Improving Age Estimation in Minors and Young Adults with Occluded Faces to Fight Against Child Sexual Exploitation, 15th International Conference on Computer Vision Theory and Applications, 2020.

[9] T. Ojala, M. Pietikinen, and D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognition, Vol. 29, No. l, pp. 51-59, 1996.

[10] S. H. Wang, P. Phillips, Z. C. Dong, Y. D. Zhang, Intelligent facial emotion recognition based on stationary wavelet entropy and jaya algorithm, Neurocomputing,Vol. 272, pp. 668-676, 2018.

[11] Q. Yin, X. Tang, J. Sun. An associate-predict model for face recognition, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 497-504, 2011.

[12] F. Yang, W. Yang, R. Gao, Q. Liao, Discriminative multidimensional scaling for low-resolution face recognition, IEEE Signal Process Letter, Vol. 25, No. 3, pp. 388-392, 2018.

[13] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815-823, 2015.

[14] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations, pp 1–14, 2015.

[15] M. Georgescu, R. T. Ionescu, M. Popescu, Local Learning With Deep and Handcrafted Features for Facial Expression Recognition, IEEE Access, Vol. 7, pp. 64827-64836, 2019.

[16] Z. Lu, X. Jiang, A. Kot, Deep Coupled ResNet for Low-Resolution Face Recognition, IEEE Signal Processing Letters, Vol. 25, No. 4, pp. 526-530, 2018.

[17] A world first: Australia's plan for advanced biometric airport check, dostupno na: https://www.airport-technology.com/features/featurea-world-first-australias-plan-for-advanced-biometric-airport-checks-5808560/ (14.9.2021.)

[18] Y. Shi, A. K. Jain, DocFace+: ID Document to Selfie Matching, Michigan, USA, 2018.

[19] Revolut, How do I verify my identity?, dostupno na: https://www.revolut.com/en-SH/help/profile-plan/verifying-identity/how-do-i-verify-my-identity (14.9.2021.)

[20] Wikipedia, ePassport gates, dostupno na: https://en.wikipedia.org/wiki/EPassport_gates (14.9.2021.)

[21] Automated Passport Control live At Nassau, dostupno na: https://www.passengerselfservice.com/2015/02/automated-passport-control-live-at-nasssau/ (14.9.2021.)

[22] A. Rosebrock, Deep Learning for Computer Vision, PyImageSearch, PyimageSearch.com, 2017.

[23] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278-2324, 1998.

[24] Službena stranica dlib alata, dostupno na: http://dlib.net/ (14.9.2021.)

[25] A. Rosebrock, Face detection with dlib (HOG and CNN), PyImageSearch.com, 2021, dostupno na: https://www.pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/ (14.9.2021.)

[26] Biblioteka *face-recognition*, pypi.org, 2020, dostupno na: https://pypi.org/project/face-recognition/ (14.9.2021.)

[27] Dokumentacija biblioteke *face-recognition*, dostupno na: https://face-recognition.readthedocs.io/en/latest/face_recognition.html (14.9.2021.)

[28] face_recognition dokumentacija, funkcija *compare_faces*, https://face-recognition.readthedocs.io/en/latest/face_recognition.html (14.9.2021.)

[29] Upute za Pripremu Fotografija za E-dokumente, Ministarstvo unutarnjih poslova Republike Hrvatske, dostupno na: https://mup.gov.hr/UserDocsImages/BannerZona/Upute%20za%20fotografije%202013%20(2).pdf (14.9.2021.)

[30] Flask dokumentacija, Foreword, https://flask.palletsprojects.com/en/2.0.x/foreword/#what-does-micro-mean (14.9.2021.)

SAŽETAK

Korištene tehnologije su: HTML, Python, OpenCV, Flask. Sustav vrši jedan na jedan usporedbu lica. Pomoću konvolucijske neuronske mreže dobivene su značajke lica koji se koriste pri provođenju eksperimenta. Eksperiment je pokazao kako je iznimno bitno imati čist skup referentnih podataka, pravilno stvoren u skladu s uputama, pri realizaciji sustava za provjeru identiteta, te je prema definiciji konvolucijskih neuronskih mreža i dubokom učenju potreban velik skup podataka za učenje. Korisnička aplikacija nudi mogućnost ugnježđivanja rješenja u ostale sustave u svrhu osiguranja osobnih podataka korisnika.

Ključne riječi: CNN, OpenCV, Python, Flask

ABSTRACT

Identity Verification System

Used technologies: HTML, Python, OpenCV, Flask. The system performs a one-on-one face comparison. Using a convolutional neural network face embeddings are obtained and used in the experiment. The experiment has shown that it is extremely important to have a clean reference set of data when implementing an identity verification system, and in accordance with the definition of convolutional neural networks and deep learning, a large set of data is needed for training. The user application offers the possibility of embedding the solution in other systems for the purposes of securing the personal data of users.

ŽIVOTOPIS

Josip Juroš rođen je 12. rujna 1997. godine u Zagrebu. Pohađao je X. Gimnaziju Ivan Supek, prirodoslovno matematički smjer, nakon završetka koje je upisao preddiplomski studij Fakultet Elektrotehnike, Računarstva i Informacijskih Tehnologija u Osijeku. Uspješno je završio preddiplomski studij 2019. godine, nakon čega je upisao diplomski sveučilišni studij, smjer Robotika i Umjetna Inteligencija.

_____

Potpis autora

PRILOG

PRILOG 1 – Kod eksperimenta

Linija:	Kod

```
1.  import face_recognition
2.  import cv2
3.  import numpy as np
4.  import glob
5.  import os
6.  import pandas as pd
7.  import seaborn as sn
8.  import matplotlib.pyplot as plt
9.  from sklearn.metrics import confusion_matrix,accuracy_score
10.
11. IMAGES_PATH = 'images'
12. MAX_DISTANCE = 0.6
13.
14. flag = False
15.
16. def getFaceEmbeddingsFromImage(image, convertToRGB=False):
17.     if convertToRGB:
18.         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
19.     face_locations = face_recognition.face_locations(image)
20.     face_encodings = face_recognition.face_encodings(image, face_locations)
21.     return face_locations, face_encodings
22.
23. def setupReference():
24.     database = {}
25.     for filename in glob.glob(os.path.join(IMAGES_PATH, '*.jpg')):
26.         image_rgb = face_recognition.load_image_file(filename)
27.         identity = os.path.splitext(os.path.basename(filename))[0]
28.         locations, encodings = getFaceEmbeddingsFromImage(image_rgb)
29.         database[identity] = encodings[0]
30.     return database
31.
32. def runFaceRecognition(reference):
33.     predicted_list = list()
34.     expected_list = list()
35.     for n in range(0,56):
36.         expected_list.append(1)
37.     test_images = list()
38.     for root, dirs, files in os.walk("./testImages"):
39.         for name in files:
40.             fpath = os.path.join(root, name)
41.             test_images.append(fpath)
42.     known_face_encoding = list(reference.values())
43.     for test_image in test_images:
44.         image = cv2.imread(test_image)
45.         face_locations, face_encodings = getFaceEmbeddingsFromImage(image,
46. convertToRGB=True)
47.         for location, face_encoding in zip(face_locations, face_encodings):
48.             distances = face_recognition.face_distance(known_face_encoding,
49. face_encoding)
50.             if np.any(distances <= MAX_DISTANCE):
51.                 best_match_idx = np.argmin(distances)
52.                 name = known_face_encoding[best_match_idx]
53.                 print("BEST MATCH INDEX: " + str(best_match_idx))
54.                 #print(name)
55.                 print(str(test_image))
```

```
                print("MATCH!")
                print("------------------------------")
                predicted_list.append(1)
            else:
                print(str(test_image))
                print("NOT MATCH!")
                print("------------------------------")
                predicted_list.append(0)
    CF = confusion_matrix(expected_list, predicted_list)
    print(CF)
    ACC = accuracy_score(expected_list, predicted_list)
    print("ACCURACY: " + str(ACC*100))
    df_cm = pd.DataFrame(CF,range(2), range(2))
    sn.set(font_scale=1.4)
    sn.heatmap(df_cm, annot=True, annot_kws={"size": 16})
    plt.show()

reference = setupReference()
runFaceRecognition(reference)
```

PRILOG 2 – Kod korisničkog rješenja

Linija:	Kod

```
1.  from app import app
2.  from flask import render_template, request
3.  import face_recognition
4.  import cv2
5.  import numpy as np
6.
7.  def getFaceEmbeddingsFromImage(image, convertToRGB=False):
8.      if convertToRGB:
9.          image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
10.     face_locations = face_recognition.face_locations(image)
11.     face_encodings = face_recognition.face_encodings(image, face_locations)
12.     return face_locations, face_encodings
13.
14. def setupReference(image_rgb):
15.     #image_rgb = face_recognition.load_image_file(filename)
16.     locations, encodings = getFaceEmbeddingsFromImage(image_rgb)
17.     database = encodings[0]
18.     return database
19.
20. @app.route('/', methods=['GET', 'POST'])
21. def index():
22.     result = ""
23.     refFile = ""
24.     testFile = ""
25.     if request.method == "POST":
26.         print("Started loading images...")
27.         refFile = request.files['referencePhoto']
28.         refImage = refFile.read()
29.         testFile = request.files['testPhoto']
30.         testImage = testFile.read()
31.         print("Starting decode to cv2...")
32.         reference = cv2.imdecode(np.fromstring(refImage, np.uint8),
33. cv2.IMREAD_UNCHANGED)
34.         testImage = cv2.imdecode(np.fromstring(testImage, np.uint8),
35. cv2.IMREAD_UNCHANGED)
36.         reference1 = setupReference(reference)
37.         face_locations, face_encodings = getFaceEmbeddingsFromImage(testImage,
38. convertToRGB=True)
39.         #for location, face_encoding in zip(face_locations, face_encodings):
40.         distances = face_recognition.face_distance(reference1, face_encodings)
41.         print(distances)
42.         print(type(distances))
43.         if np.any(distances <= 0.6):
44.             result = 'Match!'
45.         else:
46.             result = 'Not Match!'
47.     return render_template('index.html', title='Home', result=result,
48. refFile=refFile, testFile=testFile)
```

```
1.  <html>
2.      <head>
3.          <title>{{ title }} - Face Verify</title>
4.      </head>
5.      <body>
6.          <form method="POST" enctype="multipart/form-data">
7.              <p>
```

```
                <label for="avatar">Choose a Reference Photo:</label>
                <input type="file" id="referencePhotoPath" name="referencePhoto"
accept="image/png, image/jpeg"></br>
            </p>
            <p>
                <label for="avatar">Choose a Test Photo:</label>
                <input type="file" id="testPhotoPath" name="testPhoto"
accept="image/png, image/jpeg"></br>

            </p>
                <input type="submit" value="Run" name="submit">
        </form>
        <div>
            <h4>Reference image - {{ refFile }}</h4>
            <h4>Test image - {{ testFile }}</h4>
            <h3>Result - {{ result }}</h3>
        </div>
    </body>
</html>
```