SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij Računarstvo

Izborni blok Robotika i umjetna inteligencija (DRB)

KONTROLA PRISTUPA POMOĆU ZNAČAJKI LICA

Projektni zadatak

Kolegij: Raspoznavanje uzoraka i strojno učenje

Hrvoje Bogadi

Osijek, 2021.

Sadržaj

1. UVOD

Ubrzanim razvojem tehnologije posljednjih godina omogućeno je iskorištavanje i integracija računala u svakodnevicu više nego ikada do sada. U posebnom središtu pozornosti našle su se primjene računala u svrhu prepoznavanja biometrijskih značajki i identifikacije pojedinaca.

Naime, tehnologija praćenja lica dosegla je razine visoke pouzdanosti te je danas svakodnevno primijenjena u mnogim područjima. Sigurnosne agencije koriste tehnologije prepoznavanja lica kako bi mogli pratiti osobe od interesa na širokom području pa tako Interpol (*International Criminal Police Organization*) ima vlastitu bazu traženih ljudi koja se dijeli među svim državama članicama, društvene mreže poput Facebooka koriste prepoznavanje lica kako bi prepoznali i označili poznate ljude na slikama, granične službe potvrđuju identitet osobe pomoću prepoznavanja lica i digitaliziranih putovnica, mnoge tvrtke koriste prepoznavanje lica kako bi implementirali kontrolu pristupa u pojedine prostore, pronalazak nestalih ljudi, djece ili dezorijentiranih odraslih može se znatno olakšati primjenom tehnologija prepoznavanja lica. Kao što vidimo primjena je puno te je pouzdanost ove tehnologije vrlo važna i na nju se danas oslanjamo u mnogim aspektima života.

2. TEMA RADA

Fokus je ovog rada kontrola pristupa pomoću značajki lica pojedinaca. Kao pokazni primjer stvorena je desktop aplikacija za skrivanje ili otkrivanje korisnički zadanih direktorija. Princip rada je vrlo jednostavan. Ukoliko program pomoću kamere prepozna da za računalom sjedi osoba koja je odlučila skriti pojedini direktorij, navedeni direktorij (ili više njih) se otkrivaju. Odlaskom osobe zadani direktorij se ponovno skriva.

3. POSTOJEĆA ISTRAŽIVANJA

Kako bi ostvarili postavljeni cilj prepoznavanja značajki lica, u današnje vrijeme koristimo konvolucijske neuralne mreže (engl. *Convolutional Neural Networks,* dalje u radu CNN). Na takav način omogućen je ne izravan i precizan odabir parametara treniranja mreže nego se ostavlja na izbor samoj mreži da odabere parametre pomoću kojih će najbolje raditi.

Jedan od prvih revolucionarnih pristupa na ovom području bio je AlexNet [1]. Uvodeći duboke neuralne mreže u ovaj problem, autori su postigli značajno bolje rezultate nego tadašnja *state-of-the-art* mreža. Autori su pokazali kako duboka CNN sa 5 konvolucijskih slojeva i 3 potpuno spojena sloja može efikasno i vrlo uspješno klasificirati slike. Do tada je postojalo rješenje problema prepoznavanja brojeva ili slova koristeći CNN, no AlexNet je prvi pokušao riješiti problem generalizirane klasifikacije slika. U tadašnje vrijeme postojao je problem nedostatka dovoljno velike baze podataka za ovakav problem koji se riješio uvođenjem ImageNet baze podataka koja je sadržavala 15 milijuna slika visoke rezolucije raspodijeljenih u 22.000 klasa. Nakon pojave AlexNeta ovakav pristup je uzeo maha pa su se do danas, poboljšanjem kvalitete dostupnih baza podataka, a i velikim porastom računalne snage samog sklopovlja, razvile mnoge mreže daleko boljih performansi. Rad "*FaceNet: A unified embedding for face recognition and clustering*" [1] u kojem autori koristeći duboke CNN postižu tada rekordnu preciznost od 99,63% na bazi podataka „Labeled Faces in the Wild“ [2] odabran je kao glavni izvor za ovaj rad. Ubrzo nakon izdavanja FaceNet-a pojavila se nova struktura dubokih CNN predstavljena u radu autora Kaiming He, Xiangyu Zhang, Shaoqing Ren i Jian Sun [3]. Koristeći novu predstavljenu strukturu rezidualnih mreža (engl. *Residual network,* dalje u radu ResNet), autori uspijevaju produbiti mreže i poboljšati performanse uz lakšu optimizaciju parametara.

4. KONVOLUCIJSKE NEURALNE MREŽE

Konvolucijske neuralne mreže najčešće se primjenjuju pri analizi fotografija i videa, klasifikaciji fotografija, procesuiranju jezika, sustavima preporuke i slično.

Osnovni blokovi (slojevi) od kojih su građene konvolucijske neuralne mreže su ulazni sloj, konvolucijski sloj, pooling sloj, potpuno povezani sloj i normalizacije.

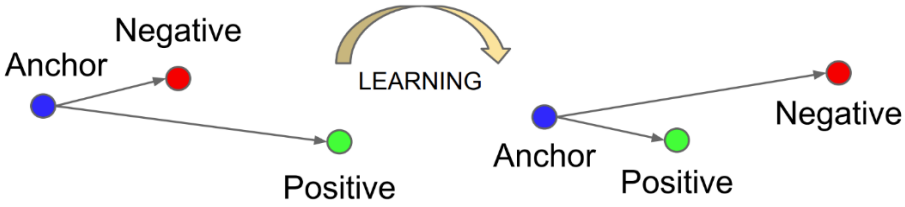
Ulazni sloj prima podatke u određenom, očekivanom obliku. Konvolucijski sloj vrši konvoluciju kojom izdvajamo bitne značajke i smanjujemo dimenzije. Prvi konvolucijski slojevi ekstrahiraju jednostavne značajke poput boje, rubova, orjentacije gradijenta i slično. Ponavljanjem i uključivanjem više ovakvih slojeva možemo izlučiti i složenije značajke.
Pooling sloj neuralne mreže smanjuje veličinu mreže usrednjavanjem vrijednosti ili odabirom najveće vrijednosti unutar odabranog kernela (jezgre). Ova dva načina rada nazivaju se Average Pooling i Max Pooling. Uporabom Max Pooling sloja možemo, uz smanjenje dimenzionalnosti, postići i uklanjanje šuma sa slika.

Potpuno povezani sloj u konvolucijskoj neuralnoj mreži uzima do tada obrađene podatke u obliku matrica, poravnava podatke u vektor te ih transformira u vektor željenih dimenzija i vrši klasifikaciju po principu feed-forward neuralne mreže te kroz svaku iteraciju treninga prilagođava vrijednosti koristeći backpropagation. Kroz određen konačni broj epoha ovakva mreža uči razlikovati podatke koje pripadaju određenim klasama uz pomoć zadane funkcije troška.

5. IMPLEMENTACIJA

Sustav koji pravimo mora moći raditi u stvarnom vremenu, odnosno trenutno prepoznavati korisnika koji sjedi ispred računala. Koristeći mrežu na tradicionalan način gdje uvodimo sliku korisnika i učimo mrežu da ga prepoznaje uzelo bi previše vremena i resursa. Kako bi riješili ovaj problem, uputno je slike predstaviti kao funkciju $f(x) : x \rightarrow \mathbb{R}^d$ gdje je x ulazna slika koju projiciramo na d-dimenzionalni euklidski prostor, gdje svaki od navedenih izlaza funkcija leži na istoj d-dimenzionalnoj hipersferi (odnosno $\|f(x)\|_2 = 1$). Na koji način mreža točno dobije značajke nam nije poznato, ali nije ni nužno, dokle god mreža za dvije iste ili slične slike kao izlaz ponudi iste ili slične vektore. U našem slučaju, to će biti 128D vektori koji se nazivaju *face embedding* odnosno *face encoding*.

Parametre mreže prilagođavamo koristeći *triplet loss* funkciju troška. Na osnovnoj razini ova metoda predstavljena je u radu [5] u kontekstu k najbližih susjedstava. Način rada metode je provjeravanje euklidske udaljenosti između slika te osiguravanje da su pripadnici iste klase (odnosno fotografije iste osobe) međusobno bliže od pripadnika različitih klasa. Za treniranje koristimo tri fotografije: fotografiju osobe za koju pravimo embedding (*anchor*), još jednu fotografiju iste osobe (*positive*) i fotografiju neke druge osobe (*negative*).



Slika 1 Grafički prikaz triplet loss metode [2]

Matematički, funkciju triplet loss možemo prikazati kao,

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2, \quad \forall \left(f(x_i^a), f(x_i^p), f(x_i^n)\right) \epsilon \tau \tag{1}$$

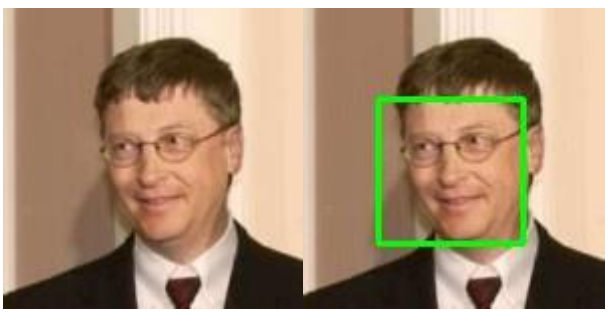
gdje je α postavljena margina između pozitivnih i negativnih parova, x_i^a *anchor* slika, x_i^p *positive* slika, x_i^n *negative* slika, a τ skup svih mogućih trojki iz trening seta s kardinalnosti N [2]. Trošak koji minimiziramo je onda,

$$L = \sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right] \quad (2)$$

Više informacija o odabiru parametara i slika koje se koriste, kao i strukturi neuralne mreže može se pronaći u [2].

Implementacija navedenoga može se postići u programskom jeziku Python koristeći biblioteke poput Keras i Tensorflow. Kako treniranje ovakve mreže traje stotinama sati uz uvjet da postoji poprilično velika računalna snaga na kojoj se trenira, u ovom radu takva mreža nije stvarana ni od čega nego je upotrijebljen gotov model prilagođen za uporabu s Keras bibliotekom [6]. Ovaj model treniran je na bazi podataka "MS-Celeb-1M".

Kako bi mogli optimalno prepoznati osobe na slikama potrebno je obraditi ulazne slike kako bi bile formata koji mreža očekuje. U našem slučaju, potrebno je imati RGB slika dimenzija 160x160x3. Kako bi tražili značajke lica, a ne okolnog prostora, potrebno je iz svake slike preuzete s kamere izolirati lice. Kako bi normalizirali slike lica i bez obzira na položaj glave korisnika uvijek imali standardizirane parametre, potrebno je pobrinuti se da je lice korisnika uvijek rotirano na isti način i da se oči uvijek nalaze na otprilike istom mjestu. Prepoznavanje lica na slikama radi se pomoću funkcija iz dlib biblioteke koje su implementacija rada „*One Millisecond Face Alignment with an Ensemle of Regression Trees"* [7]. Algoritam radi na principu klasifikacije histograma orjentiranih gradijenata pomoću stroja s potpornim vektorima te kao izlaz daje četiri koordinate pravokutnika koji opisuje pronađeno lice te koordinate 68 točaka značajki koje se traže (Slika 2 Prepoznato područje lica).



Slika 2 Prepoznato područje lica

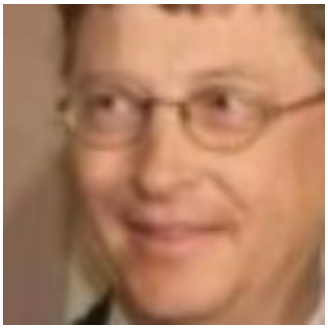
Iz ovih podataka možemo izolirati lice od ostatka slike te poravnati rotiranu sliku kako bi ulaz u neuralnu mrežu uvijek bio jednako orjentiran. Kako bi ispravili zarotirano lice, pronalazimo područje očiju te provjeravamo odnos središta dvaju pravokutnika koji opisuju oči (Slika 3). Kut θ za koji će se slika zarotirati vrlo jednostavno možemo dobiti trigonometrijom (3).

$$\theta = \tan^{-1}\left(\frac{x}{y}\right) \tag{3}$$

gdje je *x* broj piksela koji predstavljaju visinu, a *y* broj piksela koji predstavljaju širinu između dva oka osobe na slici.

Slika 3 Pronađen kut i ispravljena slika

Nakon što smo fotografiju ispravili, prepoznato lice izrezujemo iz ostatka slike i uvodimo u mrežu.



Slika 4 Izrezano područje lica

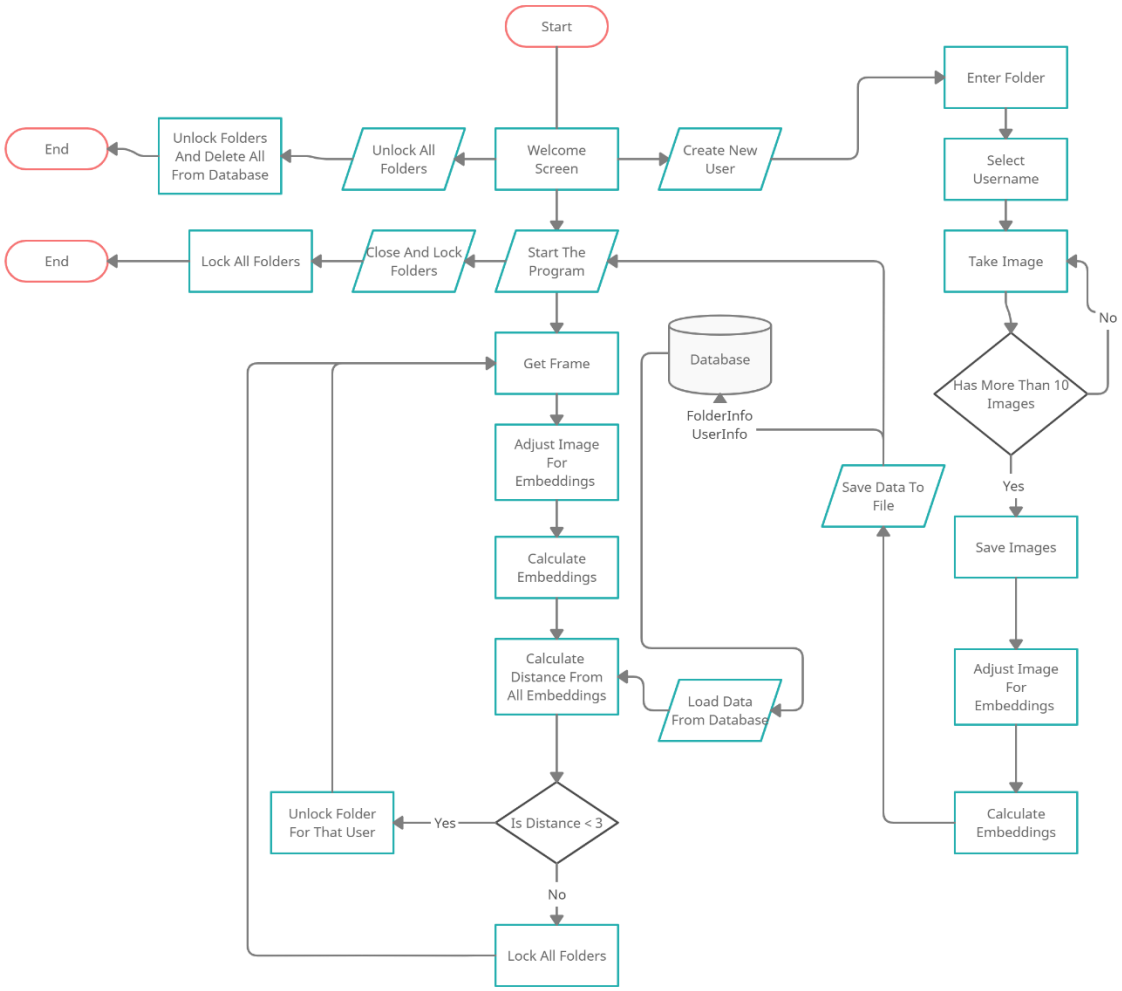
Pri stvaranju korisničkog profila korisnik uzima minimalno 10 slika svoga lica u položaju u kojemu se nalazi pri redovitoj uporabi računala te odabire direktorij koji želi skriti. Ovi podaci, zajedno s korisničkim imenom spremaju se u obliku riječnika u dvije datoteke, jedna sadrži popis zaključanih direktorija i imena korisnika koji su vlasnici direktorija, a druga sadrži embeddinge lica korisnika i pripadajuća korisnička imena.

Kada se program pokrene u radnom modu, kontinuirano uzima slike lica s kamere i za svako prepoznato lice računa embedding vektor. Nakon što je izračunao navedeni vektor, program provjerava je li mu osoba poznata i koja je to osoba tako što dani vektor uspoređuje sa svakim vektorom spremljenim u zapisanim u datoteci. Ukoliko je euklidska udaljenost dvaju vektora manja od zadane margine, program pamti ime korisnika, traži koje direktorije taj korisnik ima zaključane te ih sve ponovno prikazuje. Ukoliko se korisnik makne od računala, program više ne može prepoznati lice te automatski ključa sve direktorije iz datoteke s popisom direktorija.

Ukoliko se završi s testiranjem i korištenjem programa, klikom na dugme za otključavanje svih direktorija mogu se ponovno prikazati svi direktoriji, a datoteke s podacima se brišu.

Grafičko sučelje programa napravljeno je koristeći PySimpleGUI biblioteku.

6. TIJEK RADA PROGRAMA



7. MOGUĆNOSTI POBOLJŠANJA ALGORITMA

U trenutnom stanju pokazan je principijalan način funkcioniranja jednog ovakvog programa te njegove osnovne značajke. Mjesta za poboljšanje svakako ima pa tako možemo daljnje implementirati sigurnosne značajke ukoliko želimo napraviti iskoristivu desktop verziju programa. Na primjer, možemo implementirati da korisnik može otključati samo one direktorije koje on posjeduje umjesto brisanja kompletne baze podataka jednim klikom. Također je moguće poboljšati parametre mreže dodatnim treniranjem i boljom obradom podataka. Iako ne nužno, podaci se mogu možda bolje klasificirati koristeći neki od klasifikacijskih algoritama poput na primjer SVM, umjesto da se samo mjeri njihova udaljenost i na temelju isključivo udaljenosti određuje koji je korisnik pred računalom.

8. VREDNOVANJE RADA PROGRAMA

Analiza rada programa provodila se na način da su dvije osobe izradile korisnički profil u gotovo istim kontroliranim uvjetima te su se bilježili podaci o prepoznavanju kroz normalan rad programa. Vrednovanje je postavljeno tako da se vrednuju samo fotografije koje nisu odbačene zbog ne pronađenog lica, odnosno vrednuju se samo fotografije s kojima dobijemo embeddinge koje uspoređujemo s onima stvorenim pri kreiranju korisnika.

Ovakvim postupkom došli smo do sljedečih spoznaja:
Pri navedenim primjerima korištena je HD kamera (720p) laptopa tvrtke HP. Označimo prvu osobu sa O1 te drugu sa O2. Ukoliko pokušavamo prepoznati O1 te O1 sjedi ispred računala dobili smo podatke da je uspješno prepoznato 83 od 91 uzoraka, odnosno da je korisnik dobro prepoznat u 91,21% slučajeva. Ukoliko pokušavamo prepoznati O2 te pred računalom sjedi O2, uspješno je prepoznato 188 od 265 uzoraka, odnosno 70,94%. Ukoliko pred računalom sjedi O1, a pokušavamo prepoznati O2 dobijemo približno jednake rezultate, odnosno kriva osoba se prepozna u oko 5% slučajeva (odnosno u 95% slučajeva je prepoznata osoba koja zapravo sjedi za računalom), a kriva osoba se u slučaju kada se traži O1 prepoznaje u približno 9% slučajeva (odnosno u 91% slučajeva je program prepoznao dobru osobu).

Ono što možemo primjetiti je da povećanjem kvalitete kamere uvelike povećavamo kvalitetu rada programa. Na kameri kvalitete 480p točnost prepoznavanja O1 bila je 84,33%. Također, možemo vidjeti da na referentnoj kameri (720p) postoje razlike pri različitim osvjetljenjima. Iako je ukupni broj slika koje su uopće prihvaćene kao slike na kojima se radi prepoznavanje u programu bio manji, performanse same neuralne mreže nisu se pretjerano značajno promijenile pa je tako postotak točno prepoznatih lica O2 bio 72% (s obzirom na 70,94% u dobrim uvjetima osvjetljenja). Problem koji se istaknuo pri lošim uvjetima osvjetljenja bio je mali broj slika koje su prihvaćene kao ulazi za mrežu te veliki vremenski razmak između pronađenih lica zbog kojeg su odabrani direktoriji dobar dio vremena koji trebaju biti otključani, bili zaključani.

9. LITERATURA

[1] A. Krizhevsky, I. Sutskever i G. E. Hinton, »ImageNet classification with deep convolutional neural networks,« *Proceedings of the 25th International Conference on Neural Information Processing Systems,* svez. 1 (NIPS'12), 2012.

[2] F. Schroff, D. Kalenichenko i J. Philbin, »FaceNet: A unified embedding for face recognition and clustering,« *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 815-823, 2015.

[3] »Labeled Faces in the Wild,« [Mrežno]. Available: http://vis-www.cs.umass.edu/lfw/. [Pokušaj pristupa 20 Veljače 2021].

[4] K. He, X. Zhang, S. Ren i J. Sun, »Deep Residual Learning for Image Recognition,« *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 770-778, 2016.

[5] K. Q. Weinberger, J. Blitzer i L. K. Saul, »Distance metric learning for large margin nearest neighbor classification,« *NIPS. MIT Press,* 2006.

[6] nyoki-mtl. [Mrežno]. Available: https://github.com/nyoki-mtl/keras-facenet. [Pokušaj pristupa 20 Veljače 2021].

[7] V. Kazemi i J. Sullivan, »One millisecond face alignment with an ensemble of regression trees,« *2014 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1867-1874, 2014.