SVEUČILIŠTE U ZAGREBU
GRAFIČKI FAKULTET

Kristina Puhanić

Programiranje aplikacije za ulazak u društvene mreže putem AI identifikacije lica

Zagreb, 2022

Ovaj rad izrađen je na Grafičkom fakultetu na Katedri za tiskarske procese pod vodstvom doc. dr. sc. Diane Bratić i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2021/2022.

Popis i objašnjenja kratica korištenih u radu

1) AI – eng. *Artificial Intelligence;* u prijevodu na hrvatski *umjetna inteligencija,* predstavlja simulacija procesa ljudske inteligencije pomoću strojeva, posebno računalnih sustava, njezine specifične primjene uključuju ekspertne sustave, obradu prirodnog jezika, prepoznavanje govora i strojni vid.
2) OpenCV – eng. *Computer Vision library;* u prijevodu na hrvatski *biblioteka za računalni vid,* izgrađena je kako bi osigurala zajedničku infrastrukturu za aplikacije računalnog vida i ubrzala korištenje strojne percepcije u komercijalnim proizvodima
3) GUI – eng. *Graphical user Interface*; u prijevodu na hrvatski *grafičko korisničko sučelje*, sučelje operativnog sustava temeljeno na grafici koje koristi ikone, izbornike i miš (za klikanje na ikonu ili povlačenje izbornika) za upravljanje interakcijom sa sustavom

SADRŽAJ

1 UVOD

U dobu sa značajno velikom količinom društvenih mreža u kojemu korisnik nesmetano izmjenjuje virtualni oblik komunikacije, bilo putem slike, video sadržaja ili teksta, postupak do same ekspresije sadržaja komuniciranja je otežan. Prijavljivanje putem korisničkog imena ili e-maila i lozinke u svaku od društvenih mreža pojedinačno je iscrpno, te oduzima vrijeme korisnika. Naime, na prekretnici između dvadesetog i dvadeset prvog stoljeća, na samim početcima virtualnog društvenog života kada su samo postojali rijetki oblici društvenih mreža poput SixDegreesa ili Ryzea, problem prijave nije bio od velikog značaja jer nije bilo toliko lozinka koje je bilo potrebno pamtiti. No, budući da su u manje od jedne generacije društveni mediji evoluirali od izravne elektroničke razmjene informacija, do virtualnog okupljališta, maloprodajne platforme, pa tako sve do vitalnog marketinškog alata 21. stoljeća, način prijave je postao otežan. Napretkom novih tehnologija, kreiranjem novih ideja, stupit će na snagu nove društvene mreže što će zahtijevati kolekcije dodatnih informacija potrebnih za prijavu. Brojne različite lozinke, e-mailovi, korisnička imena i sl., predstavljaju prečicu za korisnika jer usporavaju njegovo viđenje ili dijeljenje željenog sadržaja.

U ovom radu će biti prikazano kako se uz pomoć umjetne inteligencije ili AI (eng. *Artificial Intelligence*) može omogućiti prijava u više društvenih mreža bez unosa tekstualnih podataka potrebnih za prijavu putem samo dva klika. Poticaj za izradu rada leži u pitanju *Kako olakšati i ubrzati ulazak korisnika u neku od njegovih društvenih mreža pod uvjetom da se sve društvene mreže nalaze na jednom mjestu, odnosno da su pohranjene u istoj aplikaciji?* Polazna ideja koja će omogućiti ostvarenje cilja podrazumijeva prepoznavanje lica u stvarnom vremenu (eng. *Facial Recognition*) služeći se pritom algoritmom *Haar Cascade*, te uporabom *OpenCV* biblioteke (eng. *Optimized Computer Vision library*). U integriranom razvojnom okruženju *PyCharm* uz uporabu programskog jezika *Python*, kreirat će se simulacija aplikacije čija će se izrada sastojati od tri faze: ulaska u aplikaciju, prepoznavanja korisničkog lica u stvarnom vremenu i preusmjeravanja u aplikaciju gdje su pohranjeni linkovi društvenih mreža te će tako biti omogućen ulazak u bilo koju od društvenih mreža bez dodatne prijave, samo je potreban jedan klik koji korisnika direktno spaja na njegov profil željene odabrane društvene mreže. Sva zamorna prijavljivanja mogu biti zamijenjena identifikacijom korisnikova lica kroz kameru koja se automatski uključuje nakon klika u prvoj fazi, odnosno ulaskom u aplikaciju.

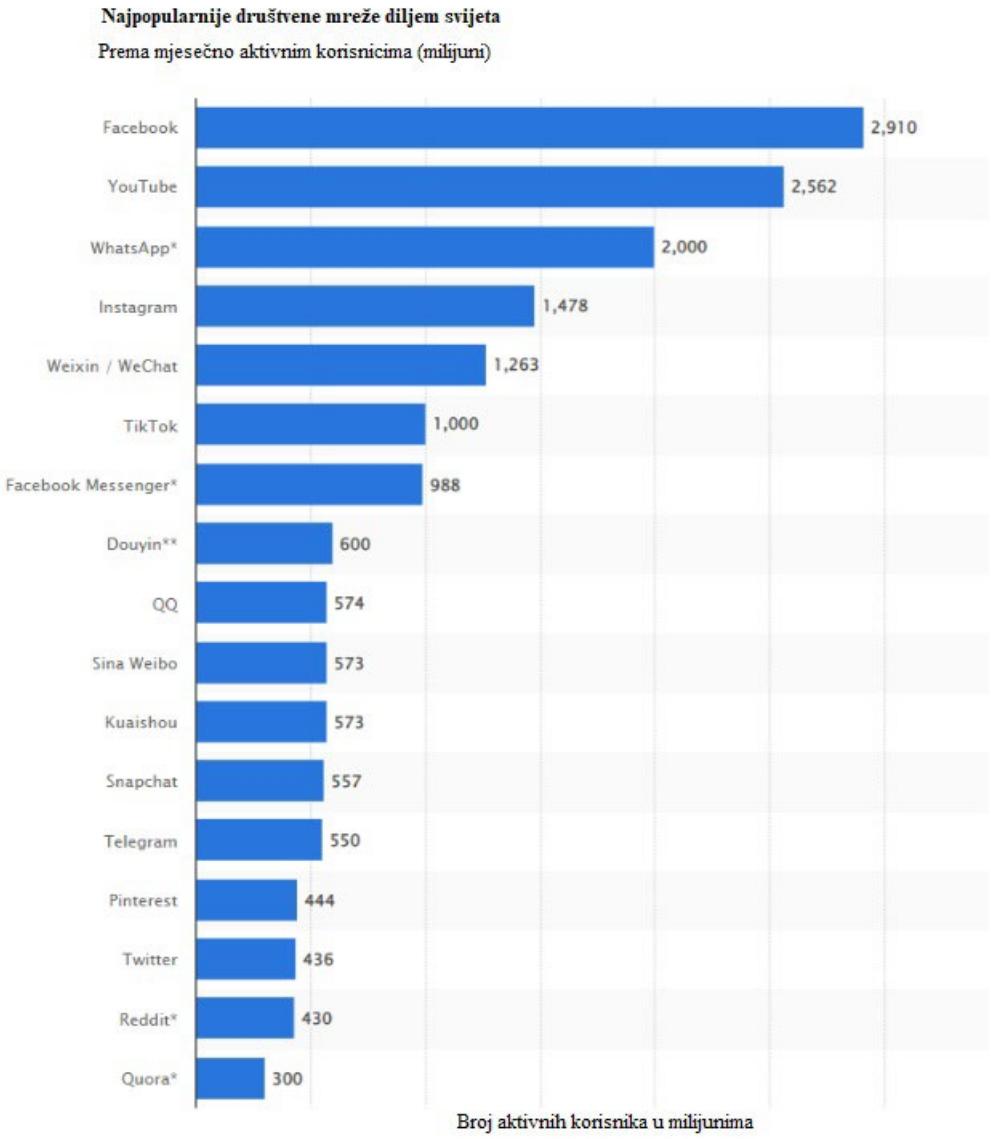
2 TEORIJSKI DIO

2.1 Društvene mreže

Društvene mreže su skupina internetskih aplikacija koje se izgrađuju na ideološkim i tehnološkim temeljima, te omogućuju stvaranje i razmjenu sadržaja koji generiraju korisnici. Temelj društvenih mreža je društvena interakcija koja sudionicima daje mogućnost dijeljenja mišljenja, misli i znanja na globalnom forumu gdje su vrijeme i mjesto beznačajni [1]. Milijarde ljudi diljem svijeta koriste društvene mreže za dijeljenje informacija i povezivanje. Na osobnoj razini, društvene mreže omogućuju komunikaciju s prijateljima i obitelji, učenje novih stvari, razvoj vlastitih interesa i zabavu. Na profesionalnoj razini se one mogu koristiti kako bi korisnik proširio znanje u određenom području i izgradio svoju profesionalnu mrežu povezujući se s drugim profesionalcima u industriji. Na poduzetnoj razini, društvene mreže omogućuju razgovor sa svojom publikom, dobivanje povratnih informacija kupaca i podizanje brenda [2].

Prema *Word Stream* izvješćima, najpopularnije društvene mreže korištene za poslovne svrhe, no ujedno i vodeće platforme društvenih mreža koje koriste trgovci širom svijeta, u periodu od siječnja 2021., pa sve do siječnja 2022. godine bile su: Facebook (93%), Instagram (78%), LinkedIn (61%), You Tube (55%), Twitter (48%), Tik Tok (9%) i Snapchat (4%) [3].

Također prema istraživanjima *Word Streama*, pet najpopularnijih društvenih mreža diljem svijeta u periodu od siječnja 2021., pa sve do siječnja 2022. godine bile su: Facebook, YouTube, WhatsApp, Instagram i Facebook Messenger. No, osim pet navedenih društvenih mreža snažno je prevladavao i utjecaj WeChata, te Tik Toka. Slika 1. prikazuje tablicu najpopularnijih društvenih mreža u svijetu u siječnju 2022., rangiranih prema broju mjesečno aktivnih korisnika [3].



Slika 1. Najpopularnije društvene mreže u svijetu - siječnj 2022.

Izvor: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Suvremeni turisti se sve više oslanjaju na platforme društvenih medija kako bi informirali o svom izboru potrošnje. Sve je više putnika koji se oslanjaju na društvene mreže jer putem njih dobivaju ideje za nadolazeća putovanja, dokaz tome je istraživanje objavljeno u časopisu *Journal of Outdoor Recreation and Tourism* koje svjedoči kako je TikTok preko noći proslavio dvije neuobičajene destinacije u Hainanu u Kini [4].

Za razliku od Tik Toka koji je usmjeren na video sadržaj, Instagram je sve popularnija platforma za društveno umrežavanje zasnovano na fotografiji. Časopis *Computers in Human Behavior Reports* je objavio istraživanje koje je nastojalo razumjeti utjecaj interakcije s Instagramom na ponašanje i dobrobit korisnika, te su istraživači proizveli iznenađujuće oprečne dokaze. Utvrdili su da se u velikom dijelu postojećih istraživanja korištenje Instagrama smatra standardiziranim, homogenim iskustvom, slično kao gledanje TV emisije ili filma. Realnost je da Instagram podržava širok raspon mogućih iskustava za korisnike, koji imaju neku agenciju u oblikovanju tog iskustva [5].

2.2 Umjetna inteligencija

U današnje vrijeme društvene mreže su postale jedan vrlo bitan čimbenik ljudskih života, a umjetna inteligencija koja se implementira u njih ima potencijal poboljšati povezanost i komunikaciju s korisnicima. Umjetna inteligencija (eng. *Artificial Intelligence - AI*) predstavlja sposobnost digitalnog računala ili računalno kontroliranog robota da obavljaju zadatke koji se obično povezuju s inteligentnim bićima [6]. Na tržištu su vidljive masovne digitalne transformacije i usvajanje tehnologija umjetne inteligencije i strojnog učenja (eng. *Machine Learning - ML*) kako bi se ubrzao rast poslovanja i povećalo zadovoljstvo kupaca.

Umjetna inteligencija ima potencijal transformirati način na koji se brendovi plasiraju na društvenim mrežama kao što su Facebook, Instagram i Twitter. Može automatizirati mnoge zamorne zadatke povezane s upravljanjem društvenih mreža, a može čak i nadgledati društvene mreže u velikom obimu. Umjetna inteligencija omogućuje trgovcima na društvenim mrežama da se približe svojoj publici i razumiju njihove preferencije. No, ona im uvelike pomaže i da svoje oglase plasiraju na bolji način, kao i kreiraju sadržaj za iste [7].

Brojna istraživanja prikazuju kako doprinos umjetne inteligencije u društvenim mrežama kreira pozitivne i korisne rezultate, jedno takvo istraživanje objavio je časopis *Software Impacts* u kojemu prezentira sustav za identifikaciju i analizu prirodnih katastrofa. *AI-Social Disaster* predstavlja sustav za identifikaciju i analizu prirodnih katastrofa poput potresa, poplava, požara pomoću sadržaja društvenih medija. Snima poruke društvenih medija u stvarnom vremenu, a zatim koristi algoritme temeljene na obradi prirodnog jezika (eng. *Natural Language Processing - NLP*) kao što su otkrivanje entiteta, klasifikacija kategorija i analiza osjećaja za prepoznavanje i lociranje

različitih prirodnih katastrofa. Štoviše, koristeći algoritme temeljene na umjetnoj inteligenciji kao što su otkrivanje anomalija, regresija i grupiranje, *AI-Social Disaster* generira uvide temeljene na umjetnoj inteligenciji za planere katastrofa i strategе. Softveru se može pristupiti putem Windows, iOS i Android aplikacija, te sa širokog spektra uređaja uključujući mobitele, tablete, laptope i stolna računala [8].

Budući da je sve veća dostupnost digitalnih slika, zajedno sa sofisticiranim tehnikama umjetne inteligencije za klasifikaciju slika, istraživači biološke raznolikosti došli su na ideju da zabilježe nove skupove podataka promatranja vrsta. U istraživanju koje je objavio časopis *Cel Press* ispitivalo se može li AI klasifikator biljnih vrsta izdvojiti prethodno neiskorištene podatke o biološkoj raznolikosti s fotografija društvenih mreža. Istraživači su pronašli preko šezdeset tisuća geolociranih slika označenih ključnom riječi „cvijet" na urbanim i ruralnim lokacijama u Ujedinjenom Kraljevstvu i klasificirali su ih pomoću umjetne inteligencije, pregledavajući te identifikacije i procjenjujući reprezentativnost slika. Slike su bile pretežno usmjerene na biološku raznolikost, prikazujući pojedinačne vrste. Klasifikator umjetne inteligencije pokazao se najboljim kada su fotografije bile usredotočene na pojedinačne autohtone vrste u divljim situacijama, ali i na višim taksonomskim razinama kao što su rod i obitelj, pa i u situacijama kada su slike znatno odstupile od toga [9].

2.2.1 Računalni vid

Računalni vid (eng. *Computer Vision*) je područje računalne znanosti koje se usredotočuje na repliciranje dijelova složenosti sustava ljudskog vida i omogućavanje računalima da identificiraju i obrađuju objekte u slikama i video zapisima na isti način na koji to čine ljudi. Donedavno je računalni vid radio samo u ograničenom kapacitetu. Zahvaljujući napretku u umjetnoj inteligenciji, te inovacijama u dubokom učenju i neuronskim mrežama, ovo je polje posljednjih godina napravilo velike skokove i uspjelo je nadmašiti ljude u nekim zadacima vezanim za otkrivanje i označavanje objekata.

Jedan od pokretačkih čimbenika razvoja računalnog vida je količina podataka koja se danas generira, a zatim se koristi za treniranje i njegovo poboljšanje. Uz ogromnu količinu vizualnih podataka, u današnje doba je dostupna i računalna snaga potrebna za analizu podataka. Kako je polje računalnog vida raslo s novim hardverom i algoritmima, tako je rasla i stopa točnosti za

identifikaciju objekata. Za manje od jednog desetljeća, današnji su sustavi dosegli 99% točnosti od 50%, što ih čini točnijim od ljudi u brzoj reakciji na vizualne unose. Rani eksperimenti s računalnim vidom započeli su 1950-ih godina, a do 1970. se nalazio u komercijalnoj upotrebi [10].

Posljednjih godina, uz kontinuirani proboj tehnologije računalnog vida, točnost detekcije objekata i prepoznavanja ciljeva naglo je poboljšana. Prepoznavanje lica jedan je od važnih istraživačkih pravaca u području računalnog vida. Tradicionalne metode prepoznavanja lica moraju ručno izdvojiti značajke slike lica. Na izdvojene značajke uvelike utječu subjektivni čimbenici, dugotrajni su i naporni. Dubinsko učenje trenutno je najvažnija tehnologija u području računalnog vida. U usporedbi s tradicionalnim metodama prepoznavanja lica, može izdvojiti bitnije značajke slike lica bez ručnog sudjelovanja. Časopis *Journal of Outdoor Recreation and Tourism* je objavio istraživanje u kojemu je prikazana izgradnja sustava za prepoznavanje lica bazirana na modelu neuronskog računanja i principu neuronske mreže. Eksperimentalni rezultati su pokazali da predložena metoda ima visoku stopu detekcije i kratko vrijeme obrade [11]. Međutim, istraživanje koje je objavio časopis Neuro computing također pridaje veliki značaj području računalnog vida. Kako je svijet doživio zdravstvenu krizu s početkom izbijanja virusa COVID-19, maska je identificirana kao najučinkovitiji način sprječavanja širenja virusa [12]. Naime, to je dovelo do potrebe za prepoznavanjem maske za lice na način koji ne samo da detektira prisutnost maske, već i daje točnost koja osoba nosi masku za lice. Također, masku za lice treba prepoznati i u svim kutovima. Cilj istraživanja bio je stvoriti novo i poboljšano prepoznavanje maski za lice u stvarnom vremenu korištenjem obrade slika i pristupa računalnog vida. Korišten je skup podataka *Kaggle* koji se sastojao od slika s maskom i bez nje. Za potrebe istraživanja korištena je unaprijed obučena konvolucijska neuronska mreža *Mobile Net V2*. Procijenjena je izvedba zadanog modela. Predstavljeni model iz istraživanja mogao je detektirati masku za lice s 98% preciznosti [13].

2.2.2 Prepoznavanje lica

Prepoznavanje lica (eng. *Facial Recognition*) način je identifikacije ili potvrđivanja identiteta pojedinca pomoću njegovog lica. Sustavi za prepoznavanje lica mogu se koristiti za prepoznavanje ljudi na fotografijama, videozapisima ili u stvarnom vremenu. Prepoznavanje lica je kategorija biometrijske sigurnosti. Ostali oblici biometrijskog softvera uključuju prepoznavanje glasa, prepoznavanje otiska prsta i prepoznavanje mrežnice ili šarenice oka. Tehnologija se uglavnom

koristi za sigurnost i provođenje zakona, iako prevladava sve veći interes i za druga područja uporabe [14].

2.2.2.1 Primjena prepoznavanja lica

Tehnologija se koristi u razne svrhe što uključuje:

1) *Otključavanje telefona*

   Razni telefoni, uključujući najnoviji iPhone, koriste prepoznavanje lica za otključavanje uređaja. Tehnologija nudi moćan način zaštite osobnih podataka i osigurava da osjetljivi podaci ostanu nedostupni u slučaju krađe telefona. Apple tvrdi da je šansa da slučajno lice otključa korisnikov telefon otprilike jedan prema milijun.

2) *Provedba zakona*

   Policija redovito koristi prepoznavanje lica. Prema izvješću NBC-a, tehnologija se povećava među agencijama za provođenje zakona u SAD-u, a isto vrijedi i za druge zemlje. Policija prikuplja fotografije uhićenika i uspoređuje ih s lokalnim, državnim i saveznim bazama podataka za prepoznavanje lica. Nakon što se napravi fotografija uhićenika, njihova će slika biti dodana u baze podataka kako bi se skenirala svaki put kada policija izvrši još jednu kriminalističku pretragu. Također, mobilno prepoznavanje lica omogućuje službenicima da koriste pametne telefone, tablete ili druge prijenosne uređaje kako bi snimili fotografiju vozača ili pješaka na terenu i odmah usporedili tu fotografiju s jednom ili više baza podataka za prepoznavanje lica kako bi pokušali identificirati.

3) *Zračne luke i granična kontrola*

   Prepoznavanje lica postalo je uobičajen prizor na mnogim zračnim lukama diljem svijeta. Sve veći broj putnika ima biometrijske putovnice, koje im omogućuju da preskoče uobičajeno duge redove i umjesto toga prođu kroz automatiziranu kontrolu e-putovnice kako bi brže stigli do ulaza. Prepoznavanje lica ne samo da skraćuje vrijeme čekanja, već i omogućuje zračnim lukama da poboljšaju sigurnost. Američko Ministarstvo domovinske sigurnosti predviđa da će se prepoznavanje lica koristiti na 97% putnika do 2023. godine. Kao i na zračnim lukama i graničnim prijelazima, tehnologija se koristi za poboljšanje sigurnosti na velikim događajima kao što su različiti sportski događaji kao što su Olimpijske igre [14].

2.2.2.2 Primjeri prepoznavanja lica

Apple koristi prepoznavanje lica kako bi korisnicima pomogao da brzo otključaju svoje telefone, prijave se u aplikacije i kupuju. British Airways omogućuje prepoznavanje lica putnicima koji se ukrcavaju na letove iz SAD-a. Lica putnika mogu se skenirati kamerom kako bi se potvrdio njihov identitet za ulazak u zrakoplov bez pokazivanja putovnice ili propusnice za ukrcaj. Zrakoplovna tvrtka koristi tehnologiju na domaćim letovima u Ujedinjenom Kraljevstvu s Heathrowa i radi na biometrijskom ukrcavanju na međunarodne letove iz zračne luke. Cigna, zdravstveno osiguranje sa sjedištem u SAD-u, dopušta korisnicima u Kini da podnose zahtjeve za zdravstveno osiguranje koji su potpisani fotografijom, a ne pisanim potpisom, u pokušaju da smanje slučajeve prijevare. Coca-Cola je koristila prepoznavanje lica na nekoliko načina diljem svijeta. Primjeri uključuju nagrađivanje kupaca za recikliranje na nekim od njezinih automata u Kini, isporuku personaliziranih oglasa na svojim automatima u Australiji i za marketing događaja u Izraelu. Facebook je počeo koristiti prepoznavanje lica u SAD-u 2010. godine kada je automatski označio ljude na fotografijama koristeći svoj alat za prijedloge oznaka. Alat skenira lice korisnika i nudi prijedloge o tome tko je ta osoba. Od 2019., Facebook je uključio tu značajku kao dio nastojanja da se više usredotoči na privatnost. Facebook time pruža informacije o tome kako se korisnik može uključiti ili isključiti za prepoznavanje lica. Google ugrađuje tehnologiju u Google fotografije i koristi je za sortiranje slika i automatsko označavanje na temelju poznatih osoba. MAC make-up koristi tehnologiju prepoznavanja lica u nekim od svojih uobičajenih trgovina, omogućujući kupcima da virtualno testiraju šminku pomoću ogledala proširene stvarnosti u trgovini. McDonald's je koristio prepoznavanje lica u svojim japanskim restoranima kako bi procijenio kvalitetu usluge korisnicima koja se tamo pruža, uključujući analizu smiješe li se zaposlenici dok pomažu korisnicima. Snapchat je jedan od pionira softvera za prepoznavanje lica. Omogućuje brendovima i organizacijama da kreiraju filtere koji se oblikuju prema licu korisnika [14].

2.2.2.3 Prednosti prepoznavanja lica

1) *Povećana sigurnost*

   Na vladinoj razini, prepoznavanje lica može pomoći u identifikaciji terorista ili drugih kriminalaca. Na osobnoj razini, prepoznavanje lica može se koristiti kao sigurnosni alat za zaključavanje osobnih uređaja i za osobne nadzorne kamere.

2) *Smanjen kriminal*

   Prepoznavanje lica olakšava pronalaženje provalnika, lopova i prijestupnika. Samo saznanje o prisutnosti sustava za prepoznavanje lica može poslužiti kao odvraćanje, posebno za sitne zločine. Osim fizičke sigurnosti, prednosti ima i kibernetička sigurnost. Tvrtke mogu koristiti tehnologiju prepoznavanja lica kao zamjenu za lozinke za pristup računalima. U teoriji, tehnologija se ne može hakirati jer se ne može ništa ukrasti ili promijeniti, kao što je slučaj s lozinkom.

3) *Uklanjanje pristranosti iz zaustavljanja i pretraživanja*

   Zabrinutost javnosti zbog neopravdanih zaustavljanja i pretraga izvor je kontroverzi za policiju, tehnologija prepoznavanja lica mogla bi poboljšati proces. Izdvajanjem osumnjičenika među gomilom putem automatiziranog, a ne ljudskog procesa, tehnologija prepoznavanja lica mogla bi pomoći u smanjenju potencijalne pristranosti i smanjenju zaustavljanja i pretraživanja građana koji poštuju zakon.

4) *Veća udobnost*

   Kako tehnologija postaje sve raširenija, kupci će moći plaćati u trgovinama licem, umjesto da izvlače svoje kreditne kartice ili gotovinu. To bi moglo uštedjeti vrijeme u redovima za naplatu. Budući da za prepoznavanje lica nije potreban kontakt kao što je to slučaj s otiskom prsta ili drugim sigurnosnim mjerama, korisnim u svijetu nakon COVID-19 krize, prepoznavanje lica nudi brzu, automatsku i besprijekornu provjeru.

5) *Brža obrada*

   Proces prepoznavanja lica traje samo sekundu, što ima prednosti za tvrtke koje koriste prepoznavanje lica. U eri cyber napada i naprednih alata za hakiranje, tvrtkama su potrebne sigurne i brze tehnologije. Prepoznavanje lica omogućuje brzu i učinkovitu provjeru identiteta osobe.

6) *Integracija s drugim tehnologijama*

   Većina rješenja za prepoznavanje lica kompatibilna je s većinom sigurnosnih softvera, lako se integrira. To ograničava iznos dodatnih ulaganja potrebnih za njegovu provedbu [14].

2.2.2.4 Nedostaci prepoznavanja lica

Iako nekim ljudima ne smeta snimanje u javnosti i ne protive se korištenju prepoznavanja lica gdje postoji jasna korist ili razlog, tehnologija može potaknuti intenzivne reakcije drugih. Neki od nedostataka ili zabrinutosti uključuju:

1) *Nadzor*

   Postoje osobe koje su sklone zabrinutosti da korištenje prepoznavanja lica uz sveprisutne video kamere, umjetnu inteligenciju i analitiku podataka stvara potencijal za masovni nadzor, što bi moglo ograničiti slobodu pojedinca. Iako tehnologija prepoznavanja lica omogućuje vlastima da uđu u trag kriminalcima, također bi im mogla omogućiti da pronađu obične i nevine ljude u bilo kojem trenutku.

2) *Opseg za pogrešku*

   Podaci o prepoznavanju lica ne sadrže pogreške, što bi moglo dovesti do toga da ljudi budu upleteni u zločine koje nisu počinili. Na primjer, mala promjena kuta kamere ili promjena izgleda, kao što je nova frizura, može dovesti do pogreške. Godine 2018. *Newsweek* je izvijestio da je Amazonova tehnologija za prepoznavanje lica lažno identificirala 28 članova američkog Kongresa kao osobe uhićene zbog zločina.

3) *Kršenje privatnosti*

   Pitanje etike i privatnosti je najspornije. Europska komisija je 2020. godine izvijestila da razmatra zabranu tehnologije prepoznavanja lica u javnim prostorima do pet godina, kako bi se omogućilo vrijeme za izradu regulatornog okvira za sprječavanje privatnosti i etičkih zloporaba.

4) *Ogromna pohrana podataka*

   Softver za prepoznavanje lica oslanja se na tehnologiju strojnog učenja, koja zahtijeva ogromne skupove podataka da bi se trenirali za isporuku točnih rezultata. Tako veliki skupovi podataka zahtijevaju robusnu pohranu podataka. Postoji mogućnost da male i srednje tvrtke nemaju dovoljno resursa za pohranu potrebnih podataka [14].

2.2.3 OpenCV biblioteka

OpenCV je virtualna biblioteka otvorenog koda za računalni vid, strojno učenje i obradu slika, no danas igra i glavnu ulogu u radu u stvarnom vremenu što je vrlo važno za današnje sustave. Njime se može obraditi slike i videozapise kako bi se identificirali objekti, lica ili čak rukopis čovjeka. Kada se integrira s raznim bibliotekama, kao što je *NumPy*, programski jezik *Python* je sposoban obraditi strukturu polja *OpenCV* biblioteke za analizu.

Da bi se identificirao uzorak slike i njegove različite značajke, koristi se vektorski prostor te se izvode matematičke operacije na tim značajkama. Prva verzija *OpenCV* biblioteke bila je 1.0. *OpenCV* biblioteka i objavljena je pod *BSD* (eng. *Berkeley Source Distribution*) licencom, te je stoga besplatna za akademsku i komercijalnu upotrebu. Posjeduje sučelja za programske jezike kao što su C, C++, P*ython* i Java, a podržava Windows, Linux, Mac OS, iOS i Android. Kada je OpenCV biblioteka dizajnirana, glavni fokus su imale aplikacije u stvarnom vremenu za računsku učinkovitost. Sve bitne stavke napisane su u optimiziranom C/C++ programskom jeziku kako bi se iskoristile prednosti višejezgrene obrade [15].

U eri informacijskog sustava autentifikacija je veliki problem. Automatizirani ugrađeni sustavi u današnjem svijetu napravili su veliki napredak. Važnost automatiziranog ugrađenog sustava pokazala se vrlo učinkovitom u aplikacijama kao što su nadzor i privatna sigurnost. Moderne pametne brave za vrata vrlo su osjetljive na greške i oštećenja, što će smanjiti sigurnost. Gotovo svaka inteligentna brava ima ulaznu šifru ili prepoznavanje lica izvan vrata što je čini ranjivom. U istraživanju koje je objavio časopis *Computer Communications*, istraživačima je bilo u cilju osigurati da sustav zaključavanja ključem koji je retro i moderan istovremeno nudi određenu dozu sigurnosti, te pouzdanosti, no također su nastojali pružiti korisniku softver otvorenog izvora izveden putem *OpenCV* biblioteke i predložiti algoritam za učinkovito praćenje stava (eng. *European Association for Transactional Analysis - EATA*). Eksperimentalni rezultati su pokazali da je predloženi sustav učinkovitiji, troši manje energije, te je isplativiji [16].

S druge strane, u istraživanju koje je objavio časopis *Science Technology and Engineering* postignuta je metoda detekcije i praćenja lica temeljena također na *OpenCV* biblioteci. Lice je važna informacija u videu, a dinamičko prepoznavanje i praćenje lica važan je dio video analize i prepoznavanja slike. Metoda *AdaBoost* koristi se za otkrivanje dinamičnog lica, a korišten je algoritam *Camshaft* koji se temelji na boji, te se služi za praćenje dinamičkog lica. Predložena

metoda je programirana korištenjem *OpenCV* biblioteke i C++ programskog jezika na razvojnoj platformi *VS 2010*, a *MFC framework* se koristio za dizajn sučelja sustava. Simulacijski eksperimentalni rezultati su pokazali da je metoda detekcije i praćenja lica u radu dala zadovoljavajuće izlaze, odnosno imala je manju računsku složenost i bolju robusnost [17].

2.3 Zaštita korisničkih podataka

Čuvanje lozinki, financijskih i drugih osobnih podataka sigurnim i zaštićenim od vanjskih uljeza dugo je bio prioritet poslovanja, ali je za potrošače i pojedince sve važnije da primjenjuju savjete o zaštiti podataka i koriste razumne prakse kako bi osjetljive osobne podatke zaštitili te osigurali. Postoji obilje informacija za potrošače, obitelji i pojedince o zaštiti lozinki, adekvatnoj zaštiti stolnih računala, prijenosnih računala i mobilnih uređaja od hakera, zlonamjernog softvera i drugih prijetnji te o najboljim praksama za sigurno korištenje interneta.

No, postoji toliko mnogo informacija, od korištenja virtualne privatne mreže (eng. *Virtual Private Network - VPN*) do korištenja jedinstvenih i jakih lozinki ili antivirusnog softvera, da vrlo lako može doći do zabune, osobito ako korisnik nije upućen u tehnologiju. Neki od načina zaštite korisničkih podataka su: šifriranje podataka, sigurnosno kopiranje podatke, osiguravanje bežične mreže kod kuće ili tvrtke, onemogućavanje čitljivosti tvrdih diskova s korisnikova prijašnjeg računala, korištenje firewalla, šifriranje podataka na USB pogonima i SIM karticama, onemogućavanje dijeljenja datoteka i medija u slučaju da ono nije potrebno, brisanje starih datoteka iz sigurnosnih kopija u oblaku, itd. [18].

Uz brzu evoluciju web tehnologija, Webu 3 (eng. *World Wide Web – Web 3*) je u cilju proširiti se na postojeće i nove platforme društvenih mreža kao što su Facebook, Twitter i TikTok, te integrirati nove računalne paradigme, uključujući IoT (eng. *Internet of Things - IoT*). Kombinacije ovih platformi u Webu 3 potrošačima obećavaju veću integraciju, interakciju i besprijekorno kretanje između fizičkih prostora. Međutim, osiguranje privatnosti podataka u takvim sustavima potencijalni je izazov u takvom prostoru. *Ad Hoc Networks* časopis je objavio istraživanje od autora Salima i Turnbulla koji su prikazali novi okvir društvenih mreža 3 za očuvanje privatnosti koji ilustrira interakciju SM (eng. *Smart Manufacturing - SM*) i IoT usluga i procjenjuje kako bi ta interakcija mogla utjecati na ponašanje korisnika. Okvir se sastoji od tri glavne komponente. Prvo, novi skup relacijskih podataka, nazvan SM-IoT, dizajniran je za dinamičko povezivanje

korisnika s IoT uslugama i pomoći u obradi heterogenosti podataka. Drugo, modul za prethodnu obradu podataka koristi se za filtriranje heterogenih podataka i osiguravanje određene razine očuvanja privatnosti podataka. Treće, analitika podataka korištenjem različitih statističkih metoda i metoda strojnog učenja primjenjuje se kako bi se ispitala složenost podataka, te identificirala ponašanja korisnika. Rezultati su otkrili da predloženi okvir može učinkovito identificirati ponašanja korisnika iz izvora podataka društvenih mreža 3 [19].

Umjetna inteligencija se naširoko koristi u društvenim mrežama kao *AD Recommender Systems*, virusni marketing i detekcija osjećaja korisnika. Međutim, takve stvari kao što su lažne vijesti i curenje privatnosti mogu narušiti privatnost i sigurnost. AI tehnologija se također često koristi u društvenim medijima kao glavna tehnologija za sprječavanje curenja osobne privatnosti. U jednom od istraživanja koje je objavio časopis *Computer Communications* ispitivalo se hoće li korisnici društvenih mreža otkriti informacije ili će dodatno zaštititi privatnost kada se suoče s problemima privatnosti informacija. Rezultati su pokazali da razumijevanje svijesti o informacijskoj sigurnosti i otuđenosti korisnika putem AI tehnologije ima pozitivan utjecaj na zabrinutost potrošača o privatnosti. Putem upravljanja privatnošću, potrošači doista mogu smanjiti svoje sumnje u vezi s curenjem privatnosti. Brige o privatnosti korisnika i kalibracija korisnika imaju pozitivne i značajne učinke na upravljanje privatnošću i samootkrivanje. Studija je također pokazala da će povjerenje korisnika u platformu utjecati na stupanj otkrivanja korisničkih podataka. Industrija bi se trebala više usredotočiti na održavanje privatnosti korisnika i uspostaviti potpuni zaštitni mehanizam [20].

3 EKSPERIMENTALNI DIO

Simulacija aplikacije koja prepoznaje lice, te nakon uspješnog prepoznavanja preusmjerava korisnika u područje pohrane njegovih društvenih mreža, sastojala se od tri dijela izrade:

1) ulazak u aplikaciju
2) prepoznavanje korisničkog lica
3) ulazak u društvenu mrežu.

Za izradu simulacije računalnog programiranja korišteno je integrirano razvojno okruženje češke tvrtke JetBrains, *PyCharm*. Za programiranje aplikacije koristio se programski jezik *Python*.

Ciljevi ovog rada su upoznavanje s konceptima, protokolima, idejama i načinom rada algoritma *Haar Cascade*, koji je ujedno i korišten u simulaciji. No, onaj najbitniji cilj zbog kojeg je nastala ideja izrade aplikacije polazi od potrebe za ubrzanijim i jednostavnijim ulaskom u neku od društvenih mreža, bez potrebe upisa lozinke, korisničkog imena ili e-maila. Naime, a dodatan plus je bila želja za mogućnošću posjedovanja svih društvenih mreža korisnika u jednoj aplikaciji, stoga je tako proizašlo istraživačko pitanje *Kako olakšati i ubrzati ulazak korisnika u neku od njegovih društvenih mreža pod uvjetom da se sve društvene mreže nalaze na jednom mjestu, odnosno da su pohranjene u istoj aplikaciji?*

Odgovor na ovo pitanje leži raspisan u poglavljima ispod, ona opisuju plan i način izvedbe svake od navedenih faza postavljenih u ovom eksperimentalnom dijelu rada. Ostvareno nastojanje autora rada omogućuje prikaz prijave u neku od korisnikovih društvenih mreža putem samo dva klika. Sva dosadašnja standardna prijavljivanja u društvene mreže pojedinačno zamijenjena su identifikacijom korisnikova lica u stvarnom vremenu, što podrazumijeva automatsku aktivaciju kamere nakon klika u prvoj fazi, odnosno ulaskom u aplikaciju. Nakon što je identifikacija izvedena uspješno, korisnik je preusmjeren u sučelje aplikacije gdje se nalazi tablica u kojoj su pohranjene njegove društvene mreže. Kada se korisnik nalazi u aplikaciji, on ostvaruje svoj drugi klik koji predstavlja njegov izbor za željenom društvenom mrežom.

3.1 Ulazak u aplikaciju

Gumb za prijavu (eng. *Login Button*) jedan je od najčešće korištenih u *GUI* (eng. *Graphical User Interface - GUI*) aplikacijama. Grafičko korisničko sučelje, koje je kasnih 1970-ih razvio istraživački laboratorij Xerox Palo Alto i komercijalno implementirano u Appleov Macintosh i Microsoft Windows operativni sustav, osmišljen je kao odgovor na problem neučinkovite upotrebljivosti u ranim, tekstualnim sučeljima naredbenog retka za prosječnog korisnika [21]. Grafička korisnička sučelja postala su standard dizajna usmjerenog na korisnika u programiranju softverskih aplikacija, pružajući korisnicima mogućnost intuitivnog upravljanja računalima i drugim elektroničkim uređajima kroz izravnu manipulaciju grafičkim ikonama kao što su gumbi, trake za pomicanje, prozori, kartice, izbornici, kursori i pokazivački uređaj miša.

Mnoga moderna grafička korisnička sučelja imaju zaslon osjetljiv na dodir i mogućnosti interakcije glasovnih naredbi. Načela dizajna grafičkog korisničkog sučelja sukladna su softverskom obrascu model–pogled–kontrolor, koji odvaja interne prikaze informacija od načina na koji se informacije prezentiraju korisniku, što rezultira platformom na kojoj se korisnicima pokazuje koje su funkcije moguće umjesto da zahtijevaju unos kodova naredbi. Korisnici stupaju u interakciju s informacijama manipulirajući vizualnim widgetima koji su dizajnirani da reagiraju u skladu s vrstom podataka koje posjeduju i podržavaju radnje potrebne za dovršetak korisnikovog zadatka. Izgled operativnog sustava ili aplikacijskog softvera može se redizajnirati po želji zbog prirode grafičkih korisničkih sučelja koja su neovisna o funkcijama aplikacije. Aplikacije obično implementiraju svoje jedinstvene elemente prikaza grafičkog korisničkog sučelja uz elemente grafičkog korisničkog sučelja koji su već prisutni u postojećem operativnom sustavu.

Tipično grafičko korisničko sučelje također uključuje standardne formate za predstavljanje grafike i teksta, što omogućuje dijeljenje podataka između aplikacija koje rade pod uobičajenim softverom za dizajn grafičkog korisničkog sučelja. Testiranje grafičkog korisničkog sučelja odnosi se na sustavni proces generiranja test slučajeva kako bi se ocijenila funkcionalnost sustava i njegovih elemenata dizajna. Alati za testiranje grafičkog korisničkog sučelja, koji su ručni ili automatizirani i obično implementirani od strane operatera trećih strana, dostupni su pod različitim licencama i podržani od strane raznih platformi. Popularni primjeri uključuju: Tricentis Tosca, Squish GUI Tester, Unified Functional Testing (UFT), Maveryx, Appium i Eggplant Functional.

Prednost grafičkog korisničkog sučelja je značajno poboljšanje u upotrebljivosti za prosječnu osobu. Značajke grafičkog korisničkog sučelja koriste poznate metafore, kao što je povuci i ispusti za prijenos datoteka, i koriste poznate ikone, kao što je koš za smeće za izbrisane datoteke, stvarajući okruženje u kojem su operacije na računalu intuitivne, te se lako svladavaju bez ikakvih prethodnih praksi ili poznavanje računalnih strojeva ili jezika.

Aplikacije grafičkog korisničkog sučelja su samoopisne, povratne informacije su obično neposredne, a vizualni znakovi potiču i usmjeravaju otkrivanje. Naime, gumb za prijavu pomaže korisnicima da se prijave koristeći najčešće korisničko ime i lozinku, nakon što su autorizacijski podaci potvrđeni, korisniku se može dati privilegirani pristup. No, u ovom eksperimentalnom radu izrada gumba za prijavu ne posjeduje navedene parametre, budući da se nakon klika na gumb za prijavu korisnik nastoji preusmjeriti direktno na drugu fazu u aplikaciji, odnosno fazu prepoznavanja lica. Drugim riječima, nakon što se klikne na gumb, poziva se funkcija *show_webcam*. Slika 2. prikazuje programski kod koji omogućava aktivaciju kamere nakon klika gumba za prijavu.

```
    self.label = tk.Label(self)
    self.label.pack()

    log_in_button = tk.Button(self, text='Log in', bg='red', command=self.show_webcam)
    log_in_button.pack()

def show_webcam(self):
    image = cv2.cvtColor(vid.read()[1], cv2.COLOR_BGR2RGB)
```

Slika 2. Programski kod aktivacije kamere nakon klika gumba za prijavu

Izvor: Prikaz autora

Za izradu gumba za prijavu koristio se Tkinter, standardna biblioteka u programskom jeziku *Python* koja se koristi za stvaranje grafičkog korisničkog sučelja za radne aplikacije. Važna značajka u korist *Tkintera* je to što je višeplatformski, tako da isti kod može lako raditi na Windowsima, MacOS-u i Linuxu. Također, biblioteka *Tkinter* znatno olakšava cijeli proces izrade gumba za prijavu, jer je *Tkinter* skup „omota“ koji implementiraju Tk widgete kao *Python* klase. *Python* s *Tkinterom* pruža brži i učinkovitiji način za izgradnju korisnih aplikacija koje bi oduzele mnogo vremena s izravnim programiranjem na C/C ++ uz pomoć izvornih knjižnica OS sustava. *Tkinter* se temelji na Tk kompletu alata koji je izvorno dizajniran za naredbeni jezik alata (Tcl).

Kako je Tk vrlo popularan, portiran je na razne druge skriptne jezike, uključujući Perl (Perl/Tk), Ruby (Ruby/Tk) i *Python* (Tkinter). Gumb za prijavu definiran je u crvenoj boji.

Osnovni koraci postavljanja *GUI* aplikacije služeći se *Tkinterom* u *Pythonu* su sljedeći:

1) pozivanje *Tkinter* modula
2) stvaranje prozorskog objekta najviše razine koji sadrži cijelu *GUI* aplikaciju
3) postavljanje svih *GUI* komponenta i njihovih funkcionalnosti
4) povezivanje komponenata *GUI-ja* s osnovnim kodom aplikacije.

Slika 3. prikazuje programski kod modula *Tkinter* za izradu gumba za prijavu.

```
class TkinterApp(tk.Tk):

    def __init__(self, *args, **kwargs):
        tk.Tk.__init__(self, *args, **kwargs)

        self.geometry('800x600')

        container = tk.Frame(self)
        container.pack(side='top', fill='both', expand=True)

        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)

        self.frames = {}

        for f in (LoginFrame, DisplayFrame):
            frame = f(container, self)
            self.frames[f] = frame
            frame.grid(row=0, column=0, sticky='nsew')

        self.show_frame(LoginFrame)

    def show_frame(self, container):
        frame = self.frames[container]
        frame.tkraise()
```

Slika 3. Programski kod moodula Tkinter za izradu gumba za prijavu

Izvor: Prikaz autora

3.2 Prepoznavanje korisničkog lica

Od svih biometrijskih mjerenja, prepoznavanje lica smatra se najprirodnijim. Intuitivno gledajući, vrlo je smisleno budući da obično ljudi prepoznaju sebe i druge gledajući lica, a ne otiske palca i šarenice. Procjenjuje se da je više od polovice svjetske populacije redovito dotaknuto tehnologijom prepoznavanja lica.

Jedna od poznatijih tehnologija prepoznavanja lica današnjice je *Face ID*, aplikacija za prepoznavanje lica koja se koristi za otključavanje iPhonea. Ovo je tehnologija prepoznavanja lica koju je dizajnirao Apple i pušten u prodaju 2017. godine, te je stvorena u sigurnosne svrhe i uvedena je u većinu novih iPhone modela, kao i na sve modele iPad Pro. Korisnici mogu otključati kompatibilne uređaje pomoću prednje kamere. *Face ID* koristi 3D modeliranje lica korisnika, tako da ga se ne može prevariti nošenjem maske ili pokazivanjem fotografije vlasnika telefona. Mape otiska lica su prilično detaljne; više od 30.000 varijabli je ugrađeno i uspoređeno. *Face ID* se može koristiti kao token za autentifikaciju u cijelom Apple ekosustavu, omogućujući autorizaciju kupnje u App Storeu, iTunes Storeu, iBooks Storeu i Apple Payu.

Međutim, *Face ID* je samo jedna od aplikacija u nizu za prepoznavanje lica, uz nju prednjače i *Face Net*, te *Face App*. *Face Net* je tehnologija umjetne neuronske mreže koju su objavili istraživači Googla 2015. godine. Ima izuzetno visoku stopu točnosti od 99,63%. Koristi se u Google fotografijama za automatsko označavanje fotografija na kojima se prepoznaje lice osobe. *FaceNet* koristi skup podataka „Označena lica u divljini“ koji je javno mjerilo za provjeru lica.

Za razliku od *Face Neta*, *Face App* je aplikacija za prepoznavanje lica koja je implementirana 2019. godine u čisto zabavne svrhe. Korisnici mogu snimiti selfie i promijeniti crte lica, promatrajući kako će izgledati starije, mlađe ili čak kao suprotni spol. Aplikacija ima mogućnosti poput promjene boje kose, nanošenja ruža za usne, dodavanja brade, te podrazumijeva promjenu drugih specifičnih detalja povezanih s izgledom [22].

Prepoznavanje lica softverski je algoritam koji se koristi za provjeru ili identifikaciju identiteta pojedinca obradom video okvira ili digitalne slike na kojoj je vidljivo lice pojedinca. Postoji nekoliko različitih metoda na kojima funkcioniraju tehnologije prepoznavanja lica, ali one općenito uspoređuju crte lica na slici s licima sadržanim u bazi podataka. Bilo koji algoritam za prepoznavanje lica koristi biometriju za mapiranje crta lica snimljenih na video-fotografiji ili fotografiji. Te se informacije zatim uspoređuju s bazom podataka lica.

U ovome radu prepoznavanje lica se ne izvršava putem videa ili fotografije, već u stvarnom vremenu. Drugim riječima, u željenom trenutku ili potrebom za prepoznavanjem lica kamera se automatski uključuje. Budući da je *OpenCV* biblioteka u današnje vrijeme toliko razvijena, omogućuje široke izvedbe implementacija računalnih programiranja, te je samim time znatno pridonijela ovome radu, omogućivši pojednostavljenu implementaciju prepoznavatelja lica (eng. *Facial Recognizer)*.

OpenCV biblioteka ima u sebi ugrađene značajke koje pojednostavljuju cijeli proces, no biblioteke dubokog učenja kao što su *TensorFlow* ili *Torch* omogućuju takav proces još naprednije. Dakle, u ovoj fazi izrade aplikacije postoje dva različita *Python* modula, jedan koji je posvećen treningu, te drugi koji je posvećen ne samo prepoznavanju, već i identifikaciji lica u stvarnom vremenu.

3.2.1 Koraci u prepoznavanju lica

1) *Prvi korak - Prepoznavanje lica*

   Prvo, kamera će otkriti i prepoznati ljudsko lice, a ono može biti u gomili ili samo. Najlakše se otkriva kada osoba gleda ravno u kameru. Međutim, suvremeni tehnološki napredak dopušta softveru za prepoznavanje lica da i dalje radi ako je lice osobe pod blagim kutom.

2) *Drugi korak - Analiza lica*

   Nakon detekcije i prepoznavanja, biti će snimljena fotografija lica i potom će biti analizirana. Većina tehnologija za prepoznavanje lica koristi 2D slike umjesto 3D, zato što se 2D fotografije lakše povezuju s javnim fotografijama ili slikama u bazi podataka, koje su obično također 2D. Tijekom analize, lice će se razdvojiti u prepoznatljive orijentire, te točke se nazivaju čvornim točkama. Ljudsko lice ima osam čvornih točaka. Tehnologija prepoznavanja lica analizirat će svaku od točaka, na primjer, udaljenost između obrva, očiju, dubinu očnih duplji, udaljenost od čela do brade, oblik jagodica i konturu usana, ušiju i brade.

3) *Treći korak - Pretvaranje slike u podatke*

   Proces snimanja lica pretvara analogne informacije, tj. lice u skup digitalnih informacija, odnosno podataka na temelju crta lica osobe. Drugim riječima, analiza lica je pretvorena u matematičku formulu. Brojčani kod naziva se otisak lica. Na isti način na koji su otisci

palca jedinstveni, svaka osoba ima svoj otisak lica. Nakon analize, svaka čvorna točka postaje broj u aplikacijskoj bazi podataka.

4) *Četvrti korak - Pronalaženje podudaranja*

   U zadnjem koraku se lice osobe uspoređuje s bazom podataka drugih poznatih lica. Na primjer, FBI ima pristup do 650 milijuna fotografija, izvučenih iz raznih državnih baza podataka. Na Facebooku svaka fotografija označena imenom osobe postaje dio Facebookove baze podataka, koja se također može koristiti za prepoznavanje lica. Ako se lice osobe podudara sa slikom u bazi podataka za prepoznavanje lica, donosi se odluka [22].

3.2.2 Algoritam Haar Cascade

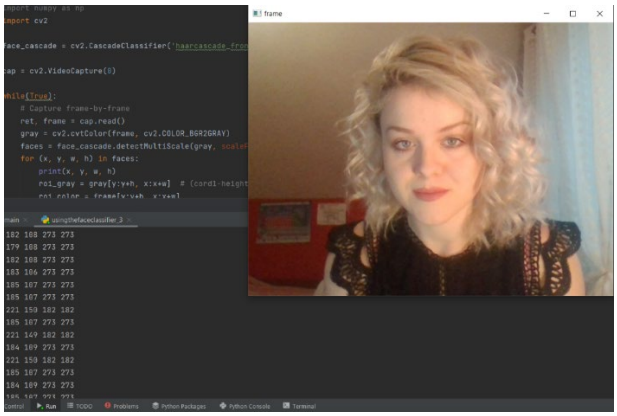
Za izradu prepoznavanje lica u aplikaciji koristio se algoritam *Haar Cascade*. Algoritam je namijenjen za otkrivanje objekata, a koristi se za prepoznavanje lica na slici ili videu u stvarnom vremenu. Algoritam koristi značajke detekcije rubova ili linija koje su predložili Viola i Jones u svom istraživačkom radu *Rapid Object Detection using a Boosted Cascade of Simple Features* objavljenom 2001. godine. Moguće je osposobiti *Haar cascade* detektor za otkrivanje raznih objekata kao što su automobili, bicikli, zgrade, voće itd [23].

Postoji veliki broj različitih vrsta algoritma *Haar Cascade*, u ovome radu se koristila vrsta algoritma koja je namijenjena prepoznavanju prednje strane lice, a naziva se *Haar Cascade frontal face default*.

3.2.2.1 Klasifikator lica

Haar Cascade je pristup koji se temelji na strojnom učenju gdje se puno pozitivnih i negativnih slika koristi za treniranje klasifikatora. Pozitivne slike su one slike koje sadrže objekt za koji se želi da klasifikator identificira, dok su negativne slike one koje ne sadrže objekt koji se želi otkriti. Kada je definirana mapa u kojoj se algoritam nalazi, nužno je bilo preoblikovati slike iz RGB-a u grayscale. Glavni razlog zašto se grayscale koristi za izdvajanje deskriptora umjesto izravnog rada na slikama u boji je taj što sivi tonovi pojednostavljuju algoritam i smanjuju računske zahtjeve. *Detect multiscale* je funkcija koja se koristi za otkrivanje lica.

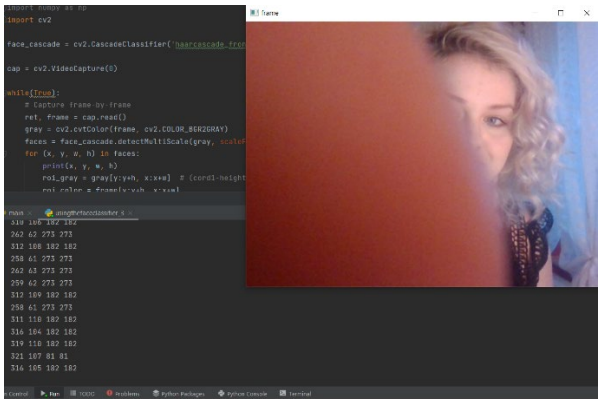
Scale factor određuje kako se veličina slike mijenja na svakoj skali slike, tako da model ima fiksnu veličinu za trening. Ako je faktor mali, otkrivanje bi moglo biti sporo jer je preciznije. Ako je faktor velik, detekcija bi mogla propustiti neka lica. Za ovaj zadatak scale factor je postavljen na 1.5. Zatim je postavljen parameter min neighbors na 5. Takav parametar određuje koliko „susjeda" svaki pravokutnik kandidata treba imati kako bi ga zadržao. Drugim riječima, ovaj parametar utječe na kvalitetu otkrivenih lica. Veća vrijednost rezultira manjim brojem otkrivanja, ali s većom kvalitetom. Vrlo je poželjno najprije isprintati navedene vrijednosti kako bi provjerili funkcionalnost. Naime, kada se programski kod pokrene automatski se uključuje kamera koja detektirano lice iščitava u numeričkom zapisu. Redovi brojeva će se pojavljivati sve dok kamera uspješno detektira lice, u tom se slučaju brojčani zapis neprestano fluidno ispisuje. Navedeni primjer prikazan je na slici 4.



Slika 4. Ispis brojčanog zapisa detekcije

Izvor: Prikaz autora

U slučaju kada kamera nije u mogućnosti detektirati lice, npr. kada je ona prekrivena kao u primjeru na slici 5., tada se ispis brojeva automatski zaustavlja.



Slika 5. Zaustavljen brojčani ispis detekcije

Izvor: Prikaz autora

Ono što se događa u navedenim primjerima na slikama 4. i 5. se zove regija interesa (eng. *Region of Interest - ROI*). Regija interesa se može prevesti još i kao relevantni raspon mjerenja. Pojam se koristi za označavanje relevantnog dijela mjerne krivulje. Ovo područje se tada može po mogućnosti promatrati statistički. Na temelju regije interesa mogu se izračunati, na primjer, maksimalna vrijednost, prosjek i širina vrha, kao i površina ispod krivulje. Krivulje mjerenja snimljene na temelju regije interesa mogu biti vrijednosti poput stope brojanja (broj po vremenskoj jedinici) registrirane spektralno, u vremenu ili duž putanje.

Što se tiče regije interesa kao dvodimenzionalnog ili trodimenzionalnog raspona, uobičajeno je u računalno kontroliranoj obradi slike i također u procesima snimanja. U ekstremnim slučajevima, u određenom vremenskom razdoblju postoji čak i četverodimenzionalni pogled. Često se područje interesa koristi u medicinskim primjenama, posebno u nuklearnoj medicini. Postupak se također koristi u kompjuterskoj tomografiji. U ovom slučaju je regija interesa višedimenzionalna [24]. Na slici 6. se nalazi prikaz programskog koda opisanih pojmova u ovom odlomku, no ujedno je prikazan i način te tijek prema kojemu ispravno funkcionira klasifikator lica.

```python
import numpy as np
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture(0)

while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)
    for (x, y, w, h) in faces:
        print(x, y, w, h)
        roi_gray = gray[y:y+h, x:x+w]  # (cord1-height, cord2-height) or (ycord_start, ycord_end)
        roi_color = frame[y:y+h, x:x+w]
        img_item = "my-image.jpg"
        cv2.imwrite(img_item, roi_gray)

    # Display the resulting frame
    cv2.imshow('frame', frame)
    if cv2.waitKey(20) & 0xFF == ord('q'):
        break
# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

Slika 6. Programski kod klasifikatora lica

Izvor: Prikaz autora

3.2.2.2 Izrada okvira

Okvir u ovom slučaju predstavlja pravokutnik koji oblikuje lice nakon što ga kamera detektira. Postavljanjem parametara koji su nužni za izradu pravokutnika kao što su boja te debljina linije, *OpenCV* biblioteka uz dane parametre je osposobljena nacrtati pravokutnik. Slika 7. prikazuje programski kod koji omogućuje izradu okvira. Okvir pravokutnika izveden je u plavoj boji s debljinom linije 2. Pogledom na programski kod, *end_cord_x* je isto što i širina (eng. *Width*), dok je *end_cord_y* zamjena za visinu (eng. *Height*).

```python
        color = (255, 0, 0)  # BGR 0-255
        stroke = 2
        end_cord_x = x + w
        end_cord_y = y + h
        cv2.rectangle(frame, (x, y), (end_cord_x, end_cord_y), color, stroke)
```

Slika 7. Programski kod za izradu okvira

Izvor: Prikaz autora

3.2.3 Trening

Kada je stvorena regija interesa, potreban je algoritam kako bi se ona prepoznala. *OpenCV* biblioteka ima mogućnost treniranja prepoznavatelja (eng. *Recognizer*), može detektirati lice, no potrebno je kreirati način putem kojega će to lice biti identificirano. U ovom odlomku je objašnjena izvedba prema kojoj je moguća identifikacija osobe. Prije samog treninga uz već postojeći dokument gdje se nalazi glavni kod zadatka bilo je potrebno kreirati novi *Python* dokument koji je bio namijenjen za prikupljanje podataka, stvaranje listi, no ujedno i njihovo spremanje, te za treniranje prepoznavatelja.

3.2.3.1 Prikupljanje podataka

Izrađena je glavna datoteka *Images* koja sadrži četiri datoteke koje su dobile naziv prema imenu osobe čije su slike u toj datoteci. Nazivi datoteka su sljedeći: *Christian*, *Gerard*, *Katherine* i *Kristina*. Datoteka *Christian* posjeduje 10 slika, datoteka *Gerard* sadrži 15 slika, datoteka *Katherine* posjeduje 12 slika, dok datoteka *Kristina* sadrži nešto više slika, 27, budući da ona predstavlja datoteku autora rada za čije se lice prepoznavatelj nastojao istrenirati. Na slici 8. se nalaze slike iz datoteke *Christian*, na slici 9. one iz datoteke *Gerard*, slika 10. predstavlja slike iz datoteke *Katherine*, dok slika 11. obuhvaća one slike koje se nalaze u datoteci *Kristina*. Količina slika i izbor osoba izabranih za trening je čisti produkt želje autora rada.

Slika 8. Slike iz datoteke Christian

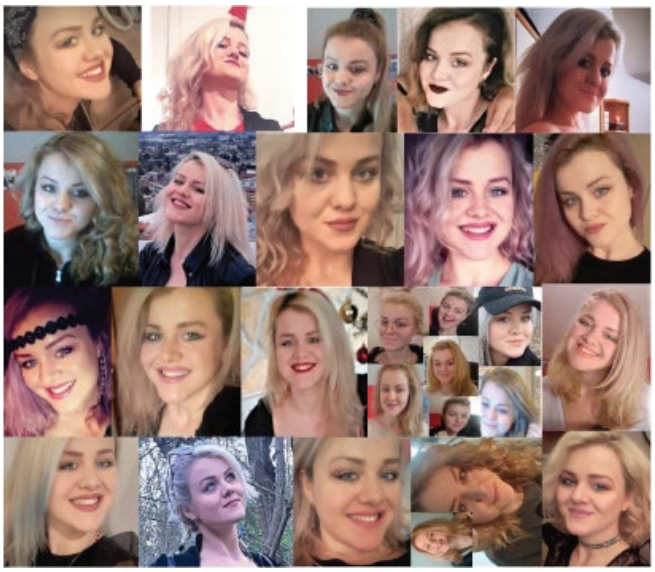
Izvor: Prikaz autora

Slika 9. Slike iz datoteke Gerard

Izvor: Prikaz autora

Slika 10. Slike iz datoteke Katherine

Izvor: Prikaz autora



Slika 11. Slike iz datoteke Kristina

Izvor: Prikaz autora

Prikupljanje slika, odnosno otvaranje slika putem programskog koda i manipuliranje njima završava se najprije pozivanjem modula *OS*. Takav modul u *Pythonu* pruža funkcije za interakciju s operativnim sustavom. *OS* dolazi pod standardnim *Pythonovim* uslužnim modulima. Ovaj modul pruža prijenosni način korištenja funkcionalnosti ovisne o operacijskom sustavu. Moduli *OS* i *os.path* uključuju razne funkcije za interakciju sa sustavom datoteka. Slika 12. prikazuje programski kod koji objašnjava kako modul *OS* prikuplja podatke.

```
import os

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
image_dir = os.path.join(BASE_DIR, "Images")

for root, dirs, files in os.walk(image_dir):
    for file in files:
        if file.endswith("png") or file.endswith("jpg"):
            path = os.path.join(root, file)
            print(path)
```

Slika 12. Programski kod prikupljanja podataka s modulom OS

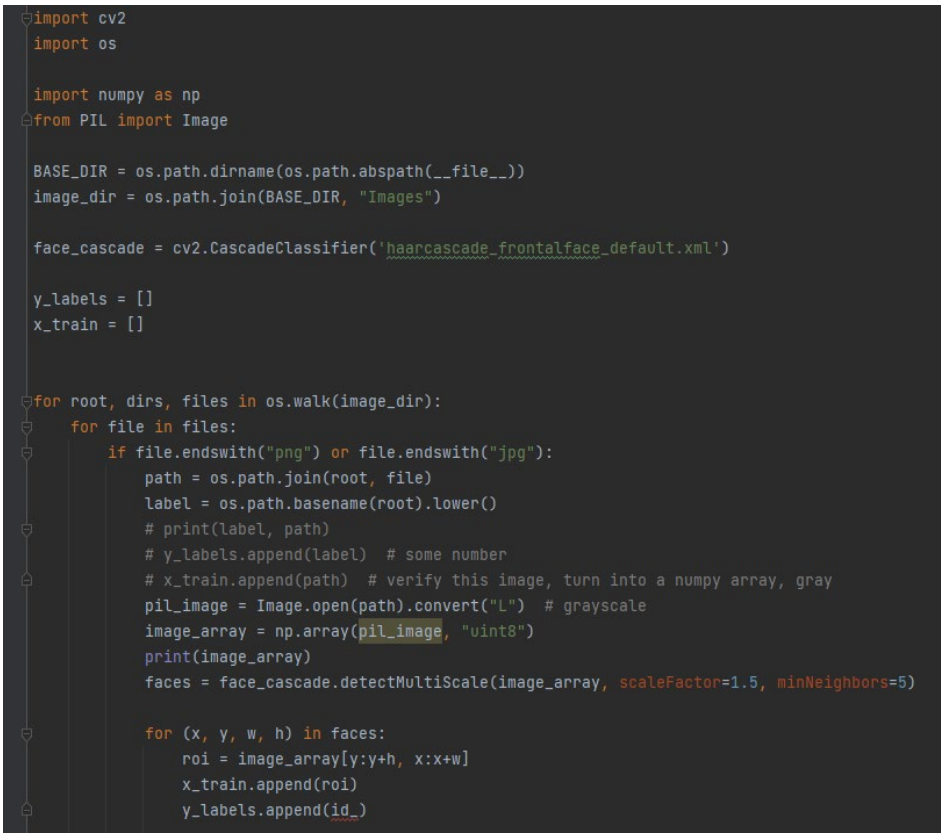
Izvor: Prikaz autora

3.2.3.2 Identifikacijske liste

Prije same izrade Identifikacijskih listi osoba sa slika, postavljena su dva parametra za izradu: *y_labels* i *x_train*. Također je postavljen i kod za trening u kojemu su slike pretvorene u *NumPy* niz (eng. *NumPy array*), jer je u cilju bilo koristiti brojčanu vrijednost za listu. *NumPy* niz je mreža vrijednosti, sve istog tipa, a indeksirana je nizom negativnih cijelih brojeva. Broj dimenzija je rang niza, a oblik niza je skup cijelih brojeva koji daje veličinu niza duž svake dimenzije.

Prije nego li su slike pretvorene u np.array, nužno ih je bilo učitati iz *PIL-a*, odnosno iz *Pythonove* slikovne bibliotke (eng. *Python Imaging Library - PIL*). Takva biblioteka predstavlja besplatnu dodatnu bazu otvorenog koda za programski jezik *Python* koja dodaje podršku za otvaranje, manipulaciju i spremanje mnogih različitih formata slikovnih datoteka. Dostupna je za Windows, Mac OS X i Linux.

Osim što se RGB vrijednost ranije preoblikovala u skalu sivih tonova (eng. *Grayscale*), ono što se također događa u navedenom zapisu na slici 13. je da su se vrijednosti piksela spremne preoblikovati u brojčani zapis, ono što u suštini i *NumPy* niz i je. Slika 13. Prikazuje programski kod prikupljenih podataka potrebnih za treniranje, no bez izrađenih lista određene osobe koja je povezana s ovom regijom interesa. Točnije, programski kod se odnosi na prikaz *NumPy* niza i detektora koji je kopiran iz glavnog koda, a vrijednosti za *scale factor*, te *min Neighbors* su jednake kao i u skripti gdje se nalazi glavni programski kod.

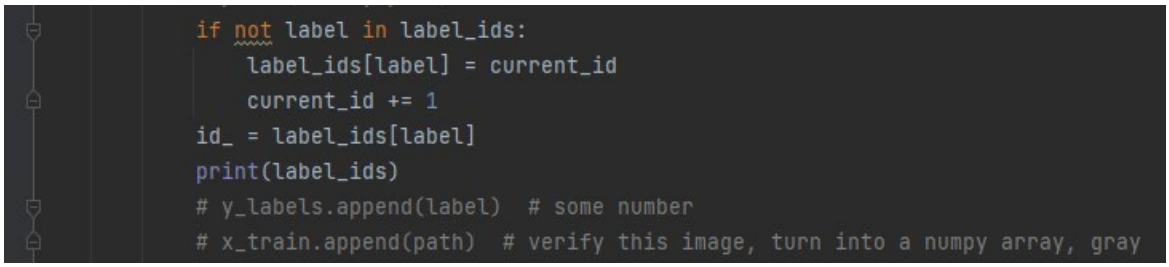


```python
import cv2
import os

import numpy as np
from PIL import Image

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
image_dir = os.path.join(BASE_DIR, "Images")

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

y_labels = []
x_train = []


for root, dirs, files in os.walk(image_dir):
    for file in files:
        if file.endswith("png") or file.endswith("jpg"):
            path = os.path.join(root, file)
            label = os.path.basename(root).lower()
            # print(label, path)
            # y_labels.append(label)  # some number
            # x_train.append(path)  # verify this image, turn into a numpy array, gray
            pil_image = Image.open(path).convert("L")  # grayscale
            image_array = np.array(pil_image, "uint8")
            print(image_array)
            faces = face_cascade.detectMultiScale(image_array, scaleFactor=1.5, minNeighbors=5)

            for (x, y, w, h) in faces:
                roi = image_array[y:y+h, x:x+w]
                x_train.append(roi)
                y_labels.append(id_)
```

Slika 13. Programski kod numpy niza i detektora

Izvor: Prikaz autora

Način na koji su kreirane identifikacijske liste vidljiv je u programskog kodu na slici 14. Naime, postavljeno je da je trenutni *id* (eng. *Current Id*) jednak nuli, a zatim za svaku novu listu koja je stvorena dodaje se jedan, odnosno *current_id += 1*. Drugim riječima, stvoren je prazan rječnik, što u programskom zapisu izgleda kao *label_ids = {}*. Stoga ako se identifikacijska lista nalazi u ovom rječniku za nju vrijedi izjava *pass*, ona se koristi kao rezervirano mjesto za budući programski kod. Kada se izvrši naredba *pass*, ništa se ne događa, ali se izbjegava dobivanje pogreške kada prazan kod nije dopušten. Prazan kod nije dopušten u petljama, definicijama funkcija, definicijama klasa ili *if* izjavama. No, za većinu identifikacijskih listi koje se ne nalaze unutar praznog rječnika napravljen je id_ koji je jednak bilo kojoj trenutnoj identifikacijskoj listi. Vrlo je bitno da se na jednoj pojedinačnoj slici nalazi samo jedna osoba, ne više njih, tj. ne više od jednog lica, jer je u cilju identificirati samo jedno lice. Na slici 14. se nalazi programski kod kreiranja identifikacijskih listi.



Slika 14. Programski kod kreiranja identifikacijskih listi

Izvor: Prikaz autora

Naposljeku potrebno je spremiti identifikacijske liste koje su dobivene za svaku izabranu osobu, jer će se liste koristiti u glavnom programskom kodu, tj. u *main.py-u*. Spremanje identifikacijskih listi izvršava se putem modula *Pickle*. Serijalizacija je tehnika koja se koristi za spremanje stanja objekta iz bilo kojeg procesa. Kasnije se može koristiti ovo stanje deserijalizacijom, za nastavak procesa. *Pickle* je *Python* modul koji olakšava serijalizaciju ili spremanje varijabli i učitavanje po potrebi. *Pickle* pretvara objekt u binarni niz, specifičan je za *Python* i može serijalizirati prilagođene klase. Zbog ove značajke, intenzivno se koristi u treniranju modela strojnog učenja. Nakon učitanog modula *Pickle*, na kraju cijelog programskog koda koji se do tada nalazio u *Python* skripti *trening.py*, zapisao se programski kod koji je prikazan na slici 15., a omogućuje uvid u pregled primjene modula pickle na identifikacijske liste. *WB* označava kraticu za pisanje bajtova (eng. *Writing Bites*). Kada su identifikacijske liste spremljene, moglo se započeti treniranje.

Slika 15. Programski kod modula pickle

Izvor: Prikaz autora

3.2.3.3 LBPH Prepoznavatelj lica

U radu je korišten *LBPH* prepoznavatelj lica (eng. *Face Recognizer*). *LBPH* (eng. *Local Binary Pattern Histogram*) je algoritam koji se koristi za prepoznavanje lica osobe. Poznat je po svojoj izvedbi i tome kako može prepoznati lice osobe i s prednje i s bočne strane [25]. Kada je postavljen prepoznavatelj u skriptu *trening.py*, što u programskom obliku izgleda poput *recognizer* =

cv2.face.LBPHFaceRecognizer_create(), započeo je trening kojim je kreirana skripta *trening.yml*, no pristup njezinom uvidu moguć je u području rezultata i rasprave ovog rada. Na slici 16. je prikazan programski kod za treniranje *LBPH* prepoznavatelja lica.

```
recognizer.train(x_train, np.array(y_labels))
recognizer.save("trening.yml")
```

Slika 16. Programski kod LBPH prepoznavatelja

Izvor: Prikaz autora

3.2.4 Implementacija istrenitranog prepoznavatelja

Nakon treniranja prepoznavatelja, nužna je njegova implementacija u glavnu skriptu programskog koda, odnosno implementacija u *main.py*. Implementacija obuhvaća dva reda programskog koda u glavnoj skripti. Naime, prvi glasi *recognizer = cv2.face.LBPHFaceRecognizer_create()*, a drugi red programskog koda posjeduje istrenirane podatke, što je ujedno i dobivena *yml* skripta, *trening.yml*. Red programskog koda glasi *recognizer.read("trening.yml")*. Nakon izvršene implementacije, može se započeti predikcija.

3.2.5 Predikcija

*Id_* i *conf* predstavljaju parametre predviđanja. Predviđala se regija interesa, točnije regija interesa sive skale tonova. Koristila se istrenirana regija interesa i identifikacijske liste. U glavni programski kod, tj. u skriptu *main.py* je postavljen interval pouzdanosti (eng. *Confidence*). Interval pouzdanosti za srednju vrijednost raspon je vrijednosti za koje je vjerojatno da će sadržavati srednju vrijednost populacije s određenom razinom pouzdanosti. U radu je postavljeno, ako je interval pouzdanosti veći ili jednak 45, a manji ili jednak 85, onda je softver u mogućnosti identificirati osobu. Programski kod intervala pouzdanosti prikazan je na slici 17.

```
        # recognize?
        id_, conf = recognizer.predict(roi_gray)
        if conf>=45 and conf <= 85:
            print(id_)
```

Slika 17. Programski kod intervala pouzdanosti

Izvor: Prikaz autora

3.2.5.1 Identifikacija s ispisom imena

Kako bi se prilikom identifikacije osobe također identificiralo njezino ime i ispisalo, potrebno je bilo kopirati pickle vrijednost iz skripte *trening.py*, te naravno, učitati modul *Pickle*. U prazni rječnik se upisuje ime osobe te se izjednačuje s vrijednošću jedan ili nekom identifikacijskom listom. Na slici 18. je prikazan programski kod za identifikaciju s ispisom imena. Kako bi se pokretanje izvršilo potrebno je bilo ispisati identifikacijske liste *_id*, što je u programskom kodu izgledalo kao *print(labels[id_]*).

```
labels = {"person_name": 1}
with open("labels.pickle", 'rb') as f:
    og_labels = pickle.load(f)
    labels = {v:k for k, v in og_labels.items()}
```

Slika 18. Programski za identifikaciju s ispisom imena

Izvor: Prikaz autora

Ispisivanje imena je zahtijevalo postavljanje nekih parametara čije je definiranje nužno za uspješnu izvedbu. Bilo je potrebno definirati parametre poput fonta, boje i debljine linije. Na slici 19. je prikazan programski kod postavljanja parametara za ispis imena.

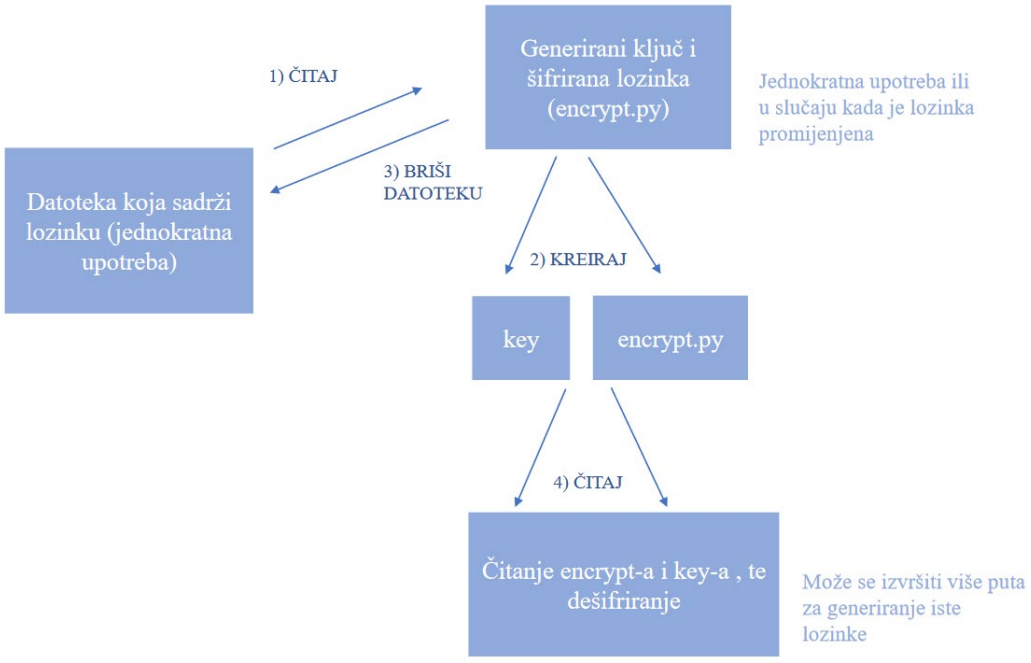
```
        font = cv2.FONT_HERSHEY_SIMPLEX
        name = labels[id_]
        color = (255, 255, 255)
        stroke = 2
        cv2.putText(frame, name, (x,y), font, 1, color, stroke, cv2.LINE_AA)
```

Slika 19. Programski kod parametara za ispis imena

Izvor: Prikaz autora

3.3 Ulazak u društvenu mrežu

U trećoj fazi eksperimentalnog dijela rada postavljen je programski kod za izradu simulacije gumba za prijavu (eng. *Login button*) u društvenu mrežu, te kod za ispis *URL-a* (eng. *Uniform Resource Locator*) i korisničkog imena (eng. *Username)*. Simulacija je kreirana samo za prijavu u jednu društvenu mrežu, no također je moguća implementacija *URL-a* i korisničkog imena većeg broja društvenih mreža. Postavljeni kod za izradu jednog gumba za ulazak u društvenu mrežu, a ne više njih, sasvim je dovoljan jer isto tako omogućuje odličan uvid u ideju rada kao i da je postavljen kod za ulazak u više društvenih mreža. U suštini kreirana je simulacija upravitelja lozinki (eng. *Password Manager*). Slika 20. prikazuje ilustrativni prikaz nastajanja upravitelja lozinki.



Slika 20. Ilustrativni prikaz nastanka upravitelja lozinki

Izvor: Prikaz autora

Svaki upravitelj lozinki mora spremiti lozinku koju korisnik daje i objaviti je kada je korisniku zatreba. U daljnjem nastavku bit će opisana tri koraka koja omogućuju izradu jednog takvog upravitelja lozinki.

1) *Izrada tekstualne datoteke*

   Prvi korak u izradi upravitelja lozinki podrazumijeva stvaranje *.txt* datoteke za pohranu lozinki. Najprije je potrebno provjeriti postoji li datoteka *info.txt* u direktoriju gdje se nalazi *Python* datoteka, no ako ne, samo se izradi. Koristi se argument *w* koji omogućuje ispis.

2) *Pisanje u datoteku*

   Kao što ispisuje u terminalu, može se na sličan način pisati u datoteku, samo se upotrebljava način za pisanje unutar datoteke. Najprije se otvara datoteka koja je kreirana u prvom koraku. Argumentom, npr. *a* se označava ono što se dodaje u tu datoteku. Može se upotrijebiti *w* što znači napisati. Ali svaki put kada se otvori datoteka s argumentom *w* ona briše sve prethodno napisano, što se nikako ne želi postići. Zatim se uzimaju korisnički podaci poput korisničkog imena, lozinki i web stranice. Stvaraju se tri varijable niza za pohranu korisničkog imena, lozinke i web stranice, a nakon toga se piše u datoteku pomoću funkcije pisanja.

3) *Iznošenje lozinki*

   Najprije se otvara datoteka, no ovaj put, umjesto dodavanja, datoteka se otvara kao pročitana za što se koristi argument *r*. Zatim se kreira nova varijabla za sadržaj koja će biti mjesto držača sadržaja u datoteci.

Datoteka s autorizacijskim podacima (eng. *Credential File*) zapravo predstavlja konfiguracijsku šifriranu datoteku s nevidljivom sigurnosnom strukturom u pozadini. Postoje situacije u kojoj se može naići na ove vrste datoteka dok se koristiti neka vrsta platformi u oblaku (eng. *Cloud*). Sve što je potrebno da bi se korisnik prijavio na instagram ili omogućio skripti dopuštenje da nešto učini bez njegovog korisničkog imena i lozinke čini se pomalo teško za zamisliti, no u suštini je vrlo jednostavno. Naime, potrebno je izraditi skriptu koja zahtijeva nekoliko konfiguracija ili autorizacijskih podataka za prijavu. Za korisnika je prilično iritantno unositi autorizacijske podatke ili konfiguracije svaki put kada želi pokrenuti programski kod. Postoji više načina na koji se to može izvesti, a u ovome radu će biti opisan jedan od njih. Najprije se kreira datoteka s autorizacijskim podacima ili datoteka konfiguracija koju skripta kasnije može koristiti za dobivanje pojedinosti koje zahtijeva. Netom nakon toga se dodaje kreator datoteke s autorizacijskim podacima. Ako korisnik želi dodati druge podatke, samo je potrebno dodati novu varijablu ili ju zapiše izravno u datoteku. Šifriranje se u ovome radu vršilo pomoću *Ferneta* u kriptografskom paketu. Kriptografija je praksa osiguravanja korisnih informacija tijekom

prijenosa s jednog računala na drugo ili pohranjivanja podataka na računalo. Kriptografija se bavi šifriranjem otvorenog teksta u šifrirani tekst i dešifriranjem šifriranog teksta u otvoreni tekst. *Python* podržava kriptografski paket koji pomaže kod šifriranja i dešifriranja podataka. *Fernet* modul kriptografskog paketa ima ugrađene funkcije za generiranje ključa, šifriranje otvorenog teksta u šifrirani tekst i dešifriranje šifriranog teksta u otvoreni tekst korištenjem metoda šifriranja, odnosno dešifriranja. *Fernet* modul jamči da se podaci šifrirani pomoću njega ne mogu dalje manipulirati ili čitati bez ključa. Metoda *generate_key ()* generira novi *Fernet* ključ. Ključ se mora čuvati na sigurnom jer je on najvažnija komponenta za dešifriranje šifriranog teksta. Ako se ključ izgubi, korisnik više ne može dešifrirati poruku. Također, ako uljez ili haker dobiju pristup ključu, ne samo da mogu čitati podatke, već ih i krivotvoriti. Metoda *encrypt (data)* šifrira podatke proslijeđene kao parametar metodi. Ishod ove enkripcije poznat je kao *Fernet token* koji je u osnovi šifrirani tekst. Šifrirani token također sadrži trenutnu vremensku oznaku kada je generiran u otvorenom tekstu. Metoda šifriranja izbacuje iznimku ako podaci nisu u bajtovima. Dakle, ključ je pohranjen u *.key* datoteci i ako korisnik želi da neka treća strana razbije enkripciju, potrebno je pretvoriti kreatora datoteke s autorizacijskim podacima (eng. *Credential Creator File*) u *.exe* ili druge formate koji se ne mogu lako čitati. Svi korišteni moduli dolaze ugrađeni u *Python*, tako da nema potrebe za vanjskom instalacijom. Što se tiče čitanja datoteke s autorizacijskim podacima, jednostavno je kao čitanje normalne datoteke pomoću *Python* metodologije čitanja datoteka, ali za dešifriranje podataka korisnik mora imati ključ koji se koristi za šifriranje. Dakle, kreator datoteke s autorizacijskim podacima stvara i datoteku s autorizacijskim podacima kao i datoteku koja predstavlja ključ. Skripta za dohvaćanje koristi ključnu datoteku i dešifrira podatke. Slika 21. prikazuje programski kod definirane klase *password store*.

```
class PasswordStore:

    def __init__(self):
        self.key = None
        self.password_file = PASSWORD_FILE_PATH
        self.password_dict = {}
        if not os.path.exists(self.password_file):
            open(PASSWORD_FILE_PATH, 'a').close()
        self.load_key(KEY_FILE)

    def load_key(self, path):
        with open(path, 'rb') as f:
            self.key = f.read()

    def list_keys(self):
        self.load_password_file(self.password_file)
        keys = self.password_dict.keys()

        def deconstruct(key):
            username, password = self.password_dict[key]
            return key, username, password

        values = [deconstruct(key) for key in keys]
        self.password_dict = {}

        return values
```

Slika 21. Programski kod klase password store

Izvor: Prikaz autora

Slika 22. prikazuje postavljeni programski kod omogućenog ulaska u društvenu mrežu nakon klika na gumb za ulaz u društvenu mrežu.

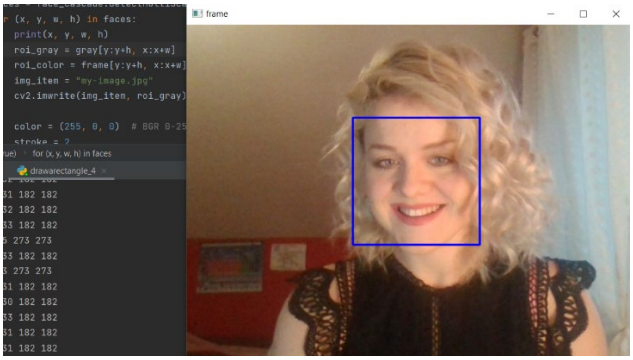
```
def __instagram_login(browser, username, password):
    browser.visit('https://instagram.com')

    browser.find_by_text('Allow essential and optional cookies').click()

    browser.find_by_name('username').fill(username)
    browser.find_by_name('password').fill(password)

    time.sleep(0.5)

    # TODO: finds div instead
    browser.find_by_text('Log In').click()


__providers = {
    'https://instagram.com': __instagram_login
}
```

Slika 22. Programski kod ulaza u društvenu mrežu

Izvor: Prikaz autora

4 REZULTATI I RASPRAVA

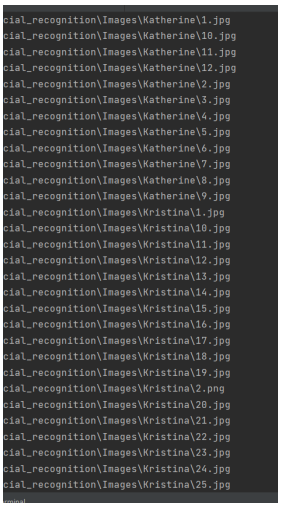
Prije samog treniranja postavljeni su parametri za izradu okvira koji je ujedno bio i pravokutnik, njegova uloga je bila uokviriti lice osobe nakon što kamera uspješno detektira lice u stvarnom vremenu, a njegova vidljivost prikazana je na slici 23. Budući da su parametri postavljeni ispravno, *OpenCV* biblioteka je bila osposobljena uspješno nacrtati pravokutnik čiji je okvir bio plave boje s debljinom linije 2.



Slika 23. Okvir za detekciju lica

Izvor: Prikaz autora

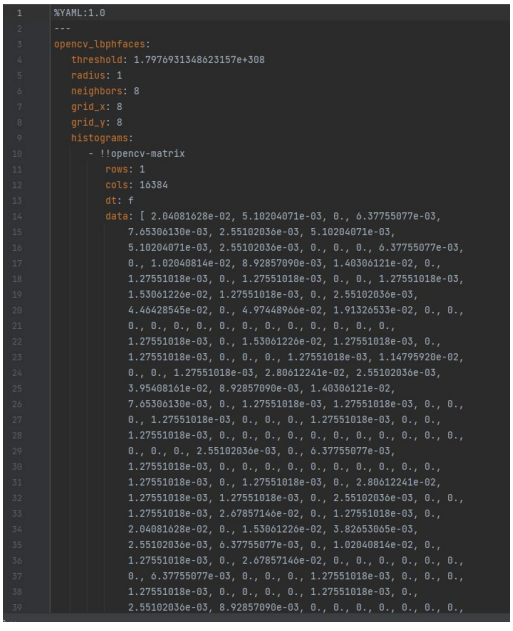
Prilikom prikupljanja podataka i ispisivanjem programskog koda prikazanog u eksperimentalnom dijelu na slici 10. (str. 25), koji je temelji na modulu *OS*, omogućen je ispravan ispis slika osoba korištenih za daljnji trening. Drugim riječima, zahvaljujući modulu *OS* koji je vrlo bitan za funkcije koje imaju ulogu s interakcijom sustava datoteka, pokretanje programskog koda je omogućilo ono što je prikazano na slici 24., djelomičan uvid u ispis slika korištenih za trening.



Slika 24. Ispis slika korištenih za trening

Izvor: Prikaz autora

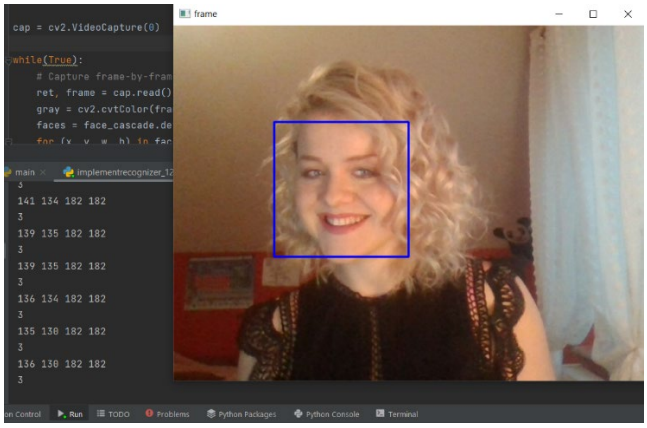
Nakon pokretanja programskog koda za treniranje prepoznavatelja, nastao je *yml* oblik skripte, tj. *trening.yml*. Takav oblik skripte radi sa sustavom predmemorije, a njezin djelomičn uvid vidljiv je na slici 25. Naznačeno je „djelomičan uvid“ zato što je veličina takve skripte znatno velika, te je njezin pregled nažalost nije nemoguć putem jedne slike.



Slika 25. Uvid u skriptu trening.yml

Izvor: Prikaz autora

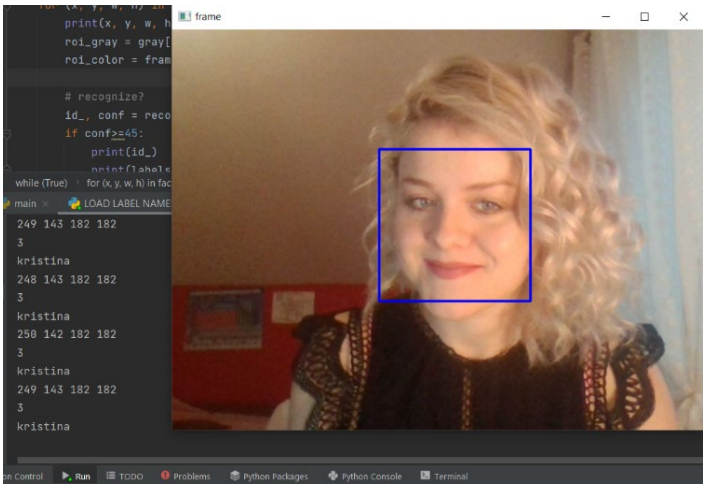
Implementacijom istreniranog prepoznavatelja u glavni programski kod, *main.py*, omogućena je vrlo povoljna predikcija. Naime, iako je algoritam klasificiran kao djelomično nepouzdan glede predikcije u odnosu na neke druge algoritme, u ovome radu pružio je zadovoljavajuće ishode. Algoritam je istreniran tako da prepoznaje lica osoba koje su na slikama u jednoj od četiri datoteke. Od svih navedenih lica osoba, u cilju je bilo da prepoznaje lice osobe koja se nalazila na slikama u datoteci *Kristina*, tj. osobe koja je ujedno i autor ovog rada. Prilikom pokretanja koda, nakon što je implementiran prepoznavatelj, predikcija je davala vrlo povoljne rezultate, odnosno prepoznavala je ispravno osobu koja se nalazila ispred kamere u stvarnom vremenu. Ispisivao se broj 3 koji je ujedno bio i zamjena za ime osobe koja se nalazila na slikama u datoteci *Kristina*. Ponekad je predviđanje znalo biti neispravno, odnosno umjesto da ispisuje broj osobe na slikama u datoteci *Kristina*, ispisivalo je broj osobe koja se nalazi na slikama u datoteci *Katherine*. Slika 26. prikazuje predikciju u stvarnom vremenu.



Slika 26. Predikcija u stvarnom vremenu

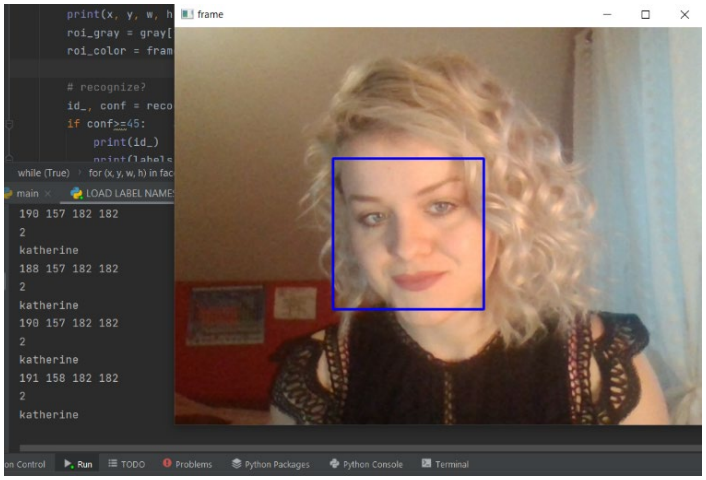
Izvor: Prikaz autora

Drugi primjer dobivene predikcije, onaj koji je prikazan na slici 27., također se temelji na predikciji u stvarnom vremenu, no s ispisom osobe koja je identificirana nakon aktivacije kamere. Budući da se ispred kamere nalazila osoba sa slika iz datoteke *Kristina*, predviđalo se da će se ispisati njezino ime, što i je. Kao i u prethodnom primjeru, bilo je ponekad situacija u kojoj bi se ispisalo ime *Katherine*, umjesto *Kristina*. Slika 27. prikazuje predikciju u stvarnom vremenu s ispisom imena osobe koja se nalazila ispred kamere, dok se na slici 28. nalazi prikaz neispravne predikcije, istrenirani prepoznavatelj ispisuje ime osobe koja se nalazi na slikama u datoteci *Katherine*.



Slika 27. Predikcija u stvarnom vremenu s ispisom imena

Izvor: Prikaz autora



Slika 28. Neispravna predikcija s ispisom krivog imena

Izvor: Prikaz autora

Razlog zbog kojega je ponekad došlo do zamijene osoba u predikciji u navedenim primjerima i zbog kojeg općenito dolazi, leži u nekoliko stavki. Budući da su zamijenjene osobe u navedenim primjerima ženskog spola, osoba sa slika iz datoteke *Katherine* kao i osoba sa slika iz datoteke *Kristina*, vrlo je moguće da je istrenirani prepoznavatelj uočio sličnosti između te dvije osobe na temelju spola, te ih je zamijenio. Naime, obje osobe imaju plavu boju kose na većini fotografija, te sličan oblik lica, kao što su im i oči slično izražene. Iako su fotografije preoblikovane u skalu sivih tonova (eng. *Grayscale*), navedeni primjeri zamijenjenih osoba i dalje su mogući.

Drugi razlog zbog kojeg su moguće potencijalne zamijene osoba tijekom predikcije, odnosno neispravne identifikacijske predikcije, je količina slika koja se koristi za treniranje prepoznavatelja. S većom količinom istreniranih slika sama predikcija bi trebala omogućiti ispravnije rezultate.

Također je od velikog značaja za ispravnu predikciju je omjer lica na slici u odnosu na okolinu. Kada je lice u većem omjeru od okoline na slici, istrenirani prepoznavatelj je u boljoj mogućnosti identificirati osobu koja se nalazi ispred kamere u stvarnom vremenu. No, kada je lice u znatno manjem omjeru u odnosu na okolinu na slici, tada postoji velika mogućnost da prepoznavatelj neće biti istreniran kvalitetno, odnosno prilikom predikcije neće biti u stanju identificirati ispravno osobu koja se nalazi ispred kamere u stvarnom vremenu. Upravo na temelju ovog slučaja, vrlo je poželjno pripremiti slike prije treninga na kojemu su lica osoba u većem omjeru spram okoline, ili jednostavno učiniti promjenu veličine (eng. *Resize*) slika putem programskog koda. Promjenom veličine slika putem programskog koda osoba koja priprema fotografije uštedi svoje vrijeme, a dobiva sličan ishod, promjena veličine slika putem programskog koda izgleda ovako:

„size = (width, height)

final_image = pil_image.resize(size, Image.ANTIALIAS)

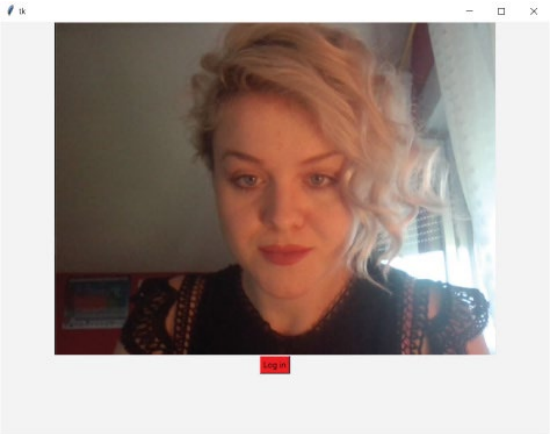
Image_array = np.array(final_image, „uint8“)“.

Spojivši sve tri faze opisane u eksperimentalnom dijelu, nastala je simulacija aplikacije. Pokretanjem glavnog programskog koda, onoga koji se nalazi u skripti *main.py*, ispisivalo je sljedeće prikazane, te ranije opisane stavke prilikom postavljanja programskog koda u eksperimentalnom dijelu. Slika 29. prikazuje rezultat prve faze eksperimentalnog dijela, tj. prvi gumb za ulazak u aplikaciju.

Slika 29. Rezultat prve faze

Izvor: Prikaz autora

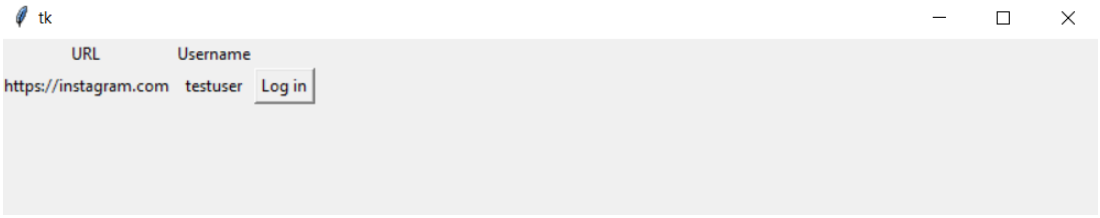
Slika 30. prikazuje rezultat postavljenog programskog koda u drugoj fazi eksperimentalnog dijela rada, odnosno identifikaciju lica putem kamere u stvarnom vremenu kako bi se omogućio ulazak u aplikaciju gdje se nalazi gumb za ulazak u društvenu mrežu.



Slika 30. Rezultat druge faze

Izvor: Prikaz autora

Slika 31. prikazuje rezultat postavljenog programskog koda za treću fazu eksperimentalnog dijela rada, što je ujedno i simulacijski prikaz URL-a, korisničkog imena, te gumba za ulazak u društvenu mrežu.



Slika 31. Rezultat treće faze

Izvor: Prikaz autora

5 ZAKLJUČAK

Danas je poznato kako je autentifikacija utemeljena na lozinkama nesigurna. Jedan od potencijalnih problema može biti što korisnik pokušava odabrati lozinke koje je lako zapamtiti. Drugi potencijalni problem može biti posjedovanje jake lozinke koja se koristi za sve aplikacije i web stranice na kojima ima račun. Navedeni pristupi mogu povećati šanse za uspješne napade, stoga za rješavanje navedenih problema postoje upravitelji lozinki. Korištenjem upravitelja lozinki korisnik treba zapamtiti samo jednu lozinku koja služi za pristup svim ostalim lozinkama, no u ovom radu je prikazano kako se sve lozinke mogu zamijeniti metodom prepoznavanja lica u stvarnom vremenu.

Znatno je puno benefita koji dolaze s tehnologijom prepoznavanja lica, kao na primjer povećavanje javne sigurnosti, brza i neinvazivna provjera identiteta, no prepoznavanje lica također proširuje mogućnosti računalnog vida. Istraživanje prepoznavanja lica moglo bi s vremenom omogućiti računalima da percipiraju i identificiraju stvari i ljude na isti način na koji to rade ljudi. Kao posljedica toga, s biometrijskom provjerom na internetu pojavit će se nove prilike za angažman. Naime, velike prednosti leže i u bankarskim prepoznavanjima lica. Unatoč nekoliko sigurnosnih mjera koje provode financijske institucije u cijelom svijetu, bankovne prijevare i dalje su trajni problem, online bankarstvo za prepoznavanje lica može pomoći u njegovom rješavanju. Provođenje zakona jedna je od najčešće predloženih primjena tehnologije prepoznavanja lica. Osim toga, to je izvor najveće svađe i žestokih rasprava. Još jedna prednost tehnologije prepoznavanja lica je njena brzina obrade i činjenica da ne treba nikakvu interakciju s korisnicima. Biometrijsko prepoznavanja lica može spriječiti neovlašteno osoblje da uđe na korisničko radno mjesto. Prepoznavanje lica može se koristiti za prepoznavanje zaposlenika pri ulasku, kao potvrda ili odbijanje pristupa. Ovo je osobito korisno ako tvrtka ima skup inventar, osjetljive informacije ili zahtijeva određene sanitarne mjere. Nadalje, korištenje biometrijskog sata s prepoznavanjem lica također čuva podatke o zaposlenicima sigurnijim nego što bi to ikada mogao ručni sustav praćenja vremena. Naime, korisnik ne može promijeniti svoje lice ili jedinstveni otisak prsta, stoga je drugim zaposlenicima ili osobama izvan organizacije nemoguće pristupiti podacima zaposlenika. Time se radni raspored zaposlenika, stopa plaće i sve druge informacije o privatnom radu čuvaju povjerljivima i dostupnima samo tom zaposleniku.

Detekcija objekata u stvarnom vremenu ključna je za mnoge aplikacije vida stvarnog svijeta. Nedavni radovi pokazuju da se određeni objekti, kao što su lica i tekst mogu pouzdano i prilično brzo detektirati.

U radu je ustanovljeno je kako pouzdanost algoritma *Haar Cascade* može vrlo varirati, no sasvim je korektan izbor za uvid u način funkcioniranja tehnologija prepoznavanja lica. Varijacije u pouzdanosti dokazane su na primjeru kada je istrenirani prepoznavatelj identificirao lice osobe koje se nalazilo ispred kamere u stvarnom vremenu kao osobu na slikama iz datoteke *Katherine*, a radilo se o osobi iz datoteke *Kristina*. Vrlo je poželjno koristiti što više slika za trening kako bi naposljetku istrenirani prepoznavatelj znao točno procijeniti o kojoj je osobi riječ, odnosno koja se nalazi ispred kamere u stvarnom vremenu.

Prognoza identifikacije u svakom slučaju mora biti pozitivna za korisnika jer u protivnom neće imati pristup društvenim mrežama koje se nalaze u aplikaciji. Generalno gledano, uspjeh algoritma za zadatke detekcije objekata, kao što je detekcija lica ili detekcija teksta, temelji se na činjenici da većina dijelova slike ne sadrži objekte od interesa. Ključni uvid algoritma je korištenje jednostavnih, brzo izračunljivih klasifikatora kako bi se odbacili dijelovi slike koji ne sadrže objekte, a da se sačuvaju oni dijelovi slike koji sadrže objekte. Zatim složenije i dugotrajnije klasifikatore treba primijeniti samo na ograničene dijelove slike.

Okvir *OpenCV* biblioteke pruža unaprijed izgrađene klasifikatore za detekciju lica i očiju koji su prilično dobre kvalitete. *OpenCV* biblioteka je izgrađena kako bi osigurala zajedničku infrastrukturu za aplikacije računalnog vida i ubrzala korištenje strojne percepcije u komercijalnim proizvodima. Kako je u radu i pokazano, postavljanjem parametara koji su nužni za izradu pravokutnika kao što su boja te debljina linije, *OpenCV* biblioteka je bila u mogućnosti uz dane parametre unaprijed nacrtati pravokutnik. Štoviše, samo je potrebno kreirati način putem kojega će lice biti identificirano, a zatim je *OpenCV* biblioteka spremna detektirati lice, budući da ima mogućnost treniranja prepoznavatelja. No, također budući da je proizvod s *BSD* licencom, *OpenCV* biblioteka olakšava tvrtkama korištenje i modificiranje koda.

ZAHVALE

Zahvaljujem svojoj mentorici doc. dr. sc. Diani Bratić za savjetovanje i vođenje tijekom izrade ovog projekta.

Velike zahvale pridajem bratiću Emanuelu Skrenkoviću na pomoći, no također i svim bližnjima na podršci.

LITERATURA

[1] Tartari, E. (2015). Benefits and Risks of Children and Adolescents Using Social Media. *European Scientific Journal*, 11(13), str. 1857-7881.

[2] University of South Florida - University Communications and Marketing. (2018). Introduction to Social Media. Dostupno na: https://www.usf.edu/ucm/marketing/intro-social-media.aspx (pristupljeno 10.6.2022.)

[3] McCormick, K. (2022). The 6 Biggest, Baddest, Most Popular Social Media Platforms in 2022 (+How to Wield Their Power). Dostupno na: https://www.wordstream.com/blog/ws/2022/01/11/most-popular-social-media-platforms (pristupljeno 10.6.2022.)

[4] Wengel, Y. et al. (2022). The Tik-Tok effect on destination development: Famous overnight, now what?. *Journal of Outdoor Recreation and Tourism,* 37(2022), str. 1-6.

[5] Ryan, E., Linehan, C. (2022). A qualitative exploration into personal psychological agency in Instagram use. *Computers in Human Behavior Reports,* 6(2022), str. 1-10.

[6] Copeland, B. J. (2022). Brittanica: Artificial intelligence. Dostupno na: https://www.britannica.com/technology/artificial-intelligence (pristupljeno 13.6.2022.)

[7] Kenyon, T. (2021). How are social media platforms using AI? Dostupno na: https://aimagazine.com/ai-strategy/how-are-social-media-platforms-using-ai (pristupljeno 12.6.2022.)

[8] Sufi, F. K. (2022). AI-Social Disaster: An AI-based software for identifying and analyzing natural disasters from social media. *Software Impacts*, 13(2022), str. 1-6.

[9] August T. A., Pescott, O. L., Joly, A., Bonnet, P. (2020). AI Naturalists Might Hold the Key to Unlocking Biodiversity Data in Social Media Imagery. *Cel Press,* 9(2020), str. 1-24.

[10] Mihajlović, I. (2019). Everything You Ever Wanted To Know About Computer Vision. Dostupno na: https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e (pristupljeno 13.6.2022.)

[11] Yu, C., Pei, H. (2021). Face recognition framework based on effective computing and adversarial neural network and its implementation in machine vision for social robots. *Computers & Electrical Engineering*, 92(1), str. 107128.

[12] Center for Disease Control and Prevention. (2022). Types of Masks and Respirators. Dostupno na: https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/types-of-masks.html (pristupljeno 13.6.2022.)

[13] Li, B., Ye, L., Liang, J. (2022). Region-of-interest and channel attention-based joint optimization of image compression and computer vision. *Neurocomputing*, 500(1), str. 13-25.

[14] Kaspersky. (2022). What is Facial Recognition – Definition and Explanation. Dostupno na: https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition (pristupljeno 13.6.2022.)

[15] Geeks for Geeks. (2021). OpenCV – Overview. Dostupno na: https://www.geeksforgeeks.org/opencv-overview/ (pristupljeno 13.6.2022.)

[16] Zhu, Z., Cheng, Y. (2020). Application of attitude tracking algorithm for face recognition based on OpenCV in the intelligent door lock. *Computer Communications*, 154(15), str. 390-397.

[17] Wang, R., Hou, P. P., Zeng, Z. L. (2014). The Application of Face Detection and Tracking Method Based on OpenCV. *Science Technology and Engineering,* 24(2014), str. 1-13.

[18] De Grot, J. (2022). 101 Data Protection Tips: How to Keep Your Passwords safe in 2022. Dostupno na: https://digitalguardian.com/blog/101-data-protection-tips-how-keep-your-passwords-financial-personal-information-safe (pristupljeno 12.6.2022.)

[19] Salim, S., Turnbull, B. P., Moustafa, N. (2022). Data analytics of social media 3.0: Privacy protection perspectives for integrating social media and Internet of Things (SM-IoT) systems. *Ad Hoc Networks,* 128(6), str. 102786.

[20] Chung, K., Chunhui, C. (2021). Social media privacy management strategies: A SEM analysis of user privacy behaviors, *Computer Communications*, 174(1), str. 122-130.

[21] Heavy AI. (2022). Graphical User Interface Definition. Dostupno na: https://www.heavy.ai/technical-glossary/graphical-user-interface (pristupljeno 17.6.2022.)

[22] Rec Faces. (2020). How Facial Recognition Works: Technology Explained in Detail. Dostupno na: https://recfaces.com/articles/how-facial-recognition-works (pristupljeno 16.6.2022.)

[23] Jaiswal, A. (2022). Towards Data Science: Object Detection with Haar Cascade. Dostupno na: https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/ (pristupljeno 17.6.2022.)

[24] Smart Ray. (2022). Region of Interest (ROI). Dostupno na: https://www.smartray.com/glossary/region-of-interest-roi/ (pristupljeno 17.6.2022.)

[25] Singh, P. (2021). Analytics Vidhya: Understanding Face Recognition using LBPH algorithm. Dostupno na: https://www.analyticsvidhya.com/blog/2021/07/understanding-face-recognition-using-lbph-algorithm/ (pristupljeno 17.6.2022.)

POPIS SLIKA

SAŽETAK

Kristina Puhanić: Programiranje aplikacije za ulazak u društvene mreže putem AI identifikacije lica

Korištenje društvenih mreža u porastu je svakim danom, pa tako i potreba za imanjem veće količine korisničkih podataka potrebnih za zamornu prijavu, poput lozinke i korisničkog imena. Budući da je poznato kako je autentifikacija temeljena na lozinkama nesigurna, otvorio se prostor za programiranje i simulaciju aplikacije koja umjesto potrebe za upisivanjem lozinke i korisničkog imena prepoznaje lice osobe u stvarnom vremenu. Nakon klika na gumb za prijavu u upravitelja lozinki ulazi se u datoteku s autorizacijskim podacima, no samo u slučaju uspješne identifikacije korisnika putem kamere.

Programiranje i simulacija su napravljeni pomoću programskog jezika *Python* u integriranom razvojnom okruženju *PyCharm* koristeći se pritom algoritmom *Haar Cascade*, te uporabom *OpenCV* biblioteke. Izrada se sastojala od tri faze: ulaska u aplikaciju, prepoznavanja korisničkog lica u stvarnom vremenu i preusmjeravanja u aplikaciju u čijoj su datoteci pohranjeni linkovi društvenih mreža te je tako omogućen ulazak u bilo koju od društvenih mreža bez dodatne prijave, samo je potreban jedan klik koji korisnika direktno spaja na njegov profil željene odabrane društvene mreže.

Predloženi model, prvenstveno sam koncept mogu poslužiti i za druga područja koja bi značajno mogla profitirati od tehnologije identifikacije lica, ne samo područje društvenih mreža, te sigurnosti.

Ključne riječi: prepoznavanje lica, društvene mreže, Haar Cascade, OpenCV, AI

SUMMARY

Kristina Puhanić: Programming an application for social media entry via AI facial recognition

Social media usage is increasing daily, highlighting the need for multiple passwords and cumbersome login procedures. Since authentication with passwords is notoriously insecure, a space has opened up for programming and simulating an application that recognizes a user's face in real-time. Moreover, the user can use the application by simply pressing the login button on the password manager, which activates the camera instead of entering a password and username. Once the face is identified, the user can enter the application using the credentials.

Programming and simulation were done with the *Python* programming language in the *PyCharm* integrated development environment (*IDE*) using the *Haar cascade* method and the *OpenCV* package. The development is done in three steps. The first step involves opening the program, the second step involves real-time face recognition and redirecting the user to the application that contains the social media links. The last step refers to the entry into any social media without additional login; a single click is enough to establish a connection between the user and the social media he/she chooses.

The proposed method and especially the concept can be of great use not only in the field of social media and security, but also in other fields that could benefit from facial recognition technology.

Key words: facial recognition, social media, Haar Cascade, OpenCV, AI

ŽIVOTOPIS

Kristina Puhanić rođena je u Našicama 9. travnja 1998. godine, gdje je pohađala svoje osnovno i srednjoškolsko obrazovanje. 2017. godine završava opću gimnaziju i odlazi na studij na Grafički fakultet Sveučilišta u Zagreb. 2020. godine postaje Sveučilišna prvostupnica, inženjerka grafičke tehnologije, no uz navedenu struku, bavi se i glazbom.

Trenutno je studentica pete godine diplomskog studija na Grafičkom fakultetu Sveučilišta u Zagrebu, smjera Dizajn grafičkih proizvoda. Tijekom studija upoznala se s programima grafičke struke, ali i umjetne inteligencije. Stoga je upravo iz područja umjetne inteligencije izradila završni rad pod nazivom *Utjecaj umjetne inteligencije na unaprjeđenje ERP sustava u grafičkim poduzećima* koji se temelji na umjetnim neuronskim mrežama, a zatim i znanstveni rad, gdje umjetnu inteligenciju prikazuje kao sadašnjost, a ne budućnost.