Sigurnosno otvrdnjavanje algoritama za prepoznavanje lica

Cindori, Dominik

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Dominik Cindori

SIGURNOSNO OTVRDNJAVANJE ALGORITAMA ZA PREPOZNAVANJE LICA

ZAVRŠNI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Dominik Cindori

Matični broj: 0016109532

Studij: Primjena informacijske tehnologije u poslovanju

SIGURNOSNO OTVRDNJAVANJE ALGORITAMA ZA PREPOZNAVANJE LICA

ZAVRŠNI RAD

Mentor :

Doc. dr. sc. Tomičić Igor

Varaždin, rujan 2020.

Dominik Cindori

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Internet stvari (IoT) i umjetna inteligencija jedne su od najnovijih tehnologija, a njihova se primjena sve više širi. Olakšavaju svakidašnji život te ubrzavaju mnoge procese, a neki izvode procese čak bolje i od ljudi. Internet stvari ili IoT odnosi se na fizičke uređaje širom svijeta koji su povezani internetom, prikupljaju i dijele podatke. Zahvaljujući novim veoma jeftinim računalnim čipovima i širokoj prisutnosti bežičnih mreža, moguće je pretvoriti bilo što, od uređaja malog poput tablete do velikih poput aviona, u IoT. Umjetna inteligencija omogućava strojevima da uče iz iskustva, prilagođavaju se novim situacijama i novim ulaznim podacima i tako odrađuju zadatke na sličan način kao i ljudi. Završni rad sastoji se od dvije cjeline: teorijskog i eksperimentalnog dijela. U teoretskom dijelu navedeni su primjeri i primjena, povijest i nastanak, veličina, prednosti te sigurnost i arhitektura IoT uređaja. U teorijskom djelu također je obrađena i umjetna inteligencija, njena važnost, povijesti, tehnike strojnog učenja, prijetnje i sigurnosni problemi. Zaključno s primjenom umjetne inteligencije u IoT uređajima, za ovaj projekt korištena je detekcija i prepoznavanje lica, koje su ujedno i eksperimentalni dio rada. U eksperimentalnom dijelu rada razvijen je, poboljšan i testiran sigurnosni aspekt pametne brave. *Pametna brava* sama po sebi ranjiva je jer ju se lako može prevariti bez sigurnosno otvrdnjenog algoritma. Bilo je potrebno izabrati pravilnu primjenu sigurnosti, s obzirom na hardverska ograničenja računalne moći i same kamere, odnosno sveukupnog hardvera. Za projekt izabrani su detekcija živosti i otkrivanje treptaja kao metode koje su uspješno osigurale čitavi sustav.

Ključne riječi: IoT, AI, NN, Python, OpenCV, prepoznavanje, lice, algoritmi, sigurnost

Sadržaj

1. Uvod

U većini domova ima mnogo stvari koje su povezane internetom i komuniciraju međusobno, a te se stvari nazivaju **Internet stvari** (engl. *Internet of Things - IoT*). IoT je proces stavljanja računalnog procesora i *WiFi* veze u svakodnevne predmete kako bi se moglo koristiti računalno programiranje za kontrolu tih uređaja. To omogućuje stvaranje rješenja pomoću uređaja koji komuniciraju i rade zajedno. Namjera je omogućiti svim uređajima u domu da rade zajedno na automatizirani način i olakšaju život. Tako, na primjer, postoje automatska svjetla, termostati koji znaju kada su ljudi kod kuće i tako dalje. Totalni pametni domovi daleka su budućnost jer sve te sitnice trebaju razgovarati jedne s drugima, tako da se većina stvari odnosi na standarde i omogućavanje uređajima da se povezuju i govore istim jezikom.

Pametna je brava IoT uređaj koji približava tehnologiju totalnim pametnim domovima. Koncept je osmišljen tako da je ispred vrata postavljena kamera koja cijelo vrijeme traži lice te pri pronalasku lica program prepoznaje lice kao osobu. Ako je prepoznato lice pohranjeno u bazi, korisnik će biti autoriziran te će otključati vrata i omogućiti korisniku da slobodno uđe u prostoriju. Za prepoznavanje lica potrebne su programske biblioteke izgrađene strojnim učenjem (odnosno dubokim učenjem, koji je jedan aspekt strojnog učenja).

Strojno učenje oblik je **umjetne inteligencije** (engl. *artificial intelligence - AI*) u kojemu je program osmišljen za samostalno učenje. Stroj uči tako da prikuplja i pohranjuje podatke od kojih se izgrađuje grafikon. Nakon količine pokušaja i pogrešaka, stroj gleda trend grafova koje je razvio i pokušava predvidjeti ishod stvaranja duž trenda. Stroj je također potrebno ručno ispravljati da bi trend bio što točniji. Trenutno postoje već istrenirani modeli za prepoznavanje lica, a prepoznavanje se svodi na uzimanje karakteristika lica, kao što su: boja očiju, oblik usana, oblik nosa, širina brade, dubina obraza, oblik čeljusti i tako dalje. Stroj tada uspoređuje sa fotografijom koju ima pohranjenu za korisnika i ako dođe do podudaranja iznad unaprijed programirane sigurnosti, tada će shvatiti da se fotografija koju stroj vidi na kameri podudara s onom koju ima u svojoj pohrani.

2. Metode i tehnike rada

Za izvršavanje **prepoznavanja lica** potrebno je imati:

- OpenCV
- Python
- Deep learning.

Da bi se izvršilo prepoznavanje lica s Pythonom i OpenCV-om potrebno je instalirati dvije dodatne biblioteke:

- `dlib`
- `face_recognition`.

Biblioteka `dlib`, koju održava Davis King, sadrži implementaciju *dubokog metričkog učenja* koje se koristi za izradu ugrađenih lica (ugrađeno je lice vektor koji predstavlja značajke izvučene s lica) koji se koriste za postupak prepoznavanja [1].

Biblioteka `face_recognition`, koju je stvorio Adam Geitgey, omotava dlibovu funkciju prepoznavanja lica, olakšavajući rad s njom [2].

Prepoznavanje lica temeljeno na ovim bibliotekama vrlo je precizno i može se izvršiti u stvarnom vremenu.

Izvor može biti fotografija ili video. Za potrebe ovoga rada korišten je video *stream* s IP interfonske video kamere **Hikvision KD8 Series Pro Modular Door Station**.

Slika 1: Hikvision KD8 Series Pro Modular Door Station
[3]

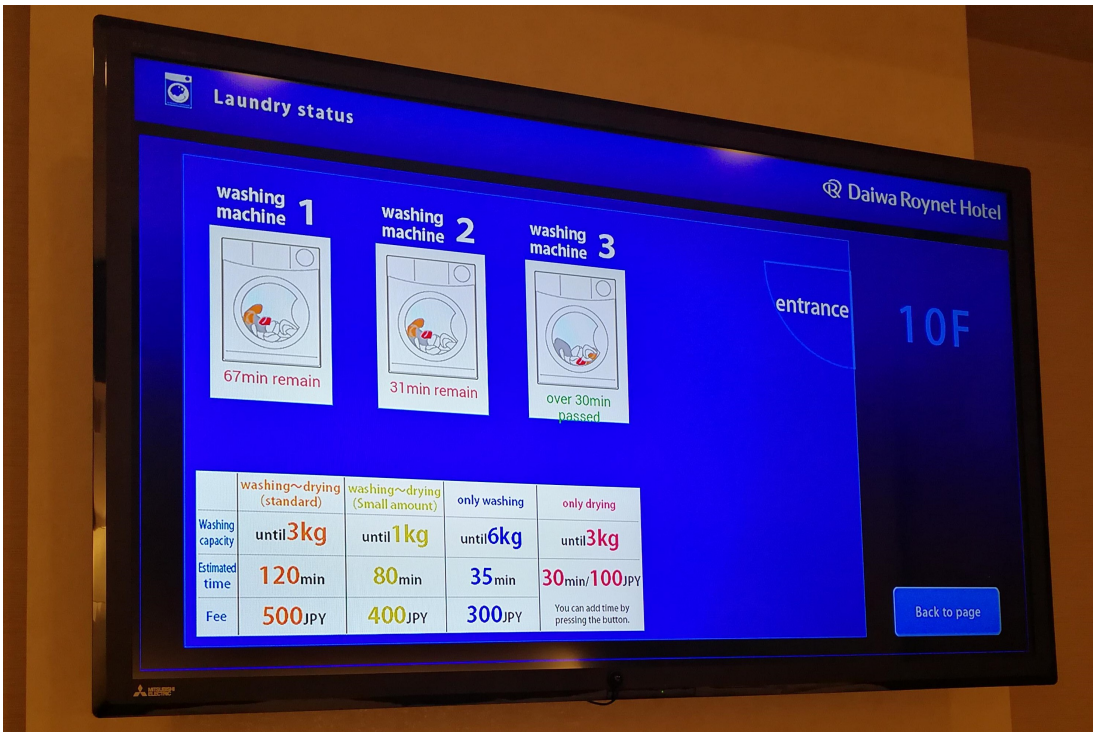
3. Internet stvari - IoT

Internet stvari ili IoT odnosi se na fizičke uređaje širom svijeta koji su povezani internetom i koji prikupljaju i dijele podatke. Zahvaljujući dolasku veoma jeftinih računalnih čipova i široke prisutnosti bežičnih mreža, moguće je u takav uređaj pretvoriti bilo što, od uređaja malog poput tablete do velikih poput aviona. **Povezivanje** svih tih različitih objekata i **dodavanje senzora** na njih dodaje razinu digitalne inteligencije uređajima koji bi inače bili "glupi", omogućujući im da komuniciraju u stvarnom vremenu bez uključivanja čovjeka. IoT čini okruženje čovjeka pametnijim i odgovornijim, spajajući digitalni i fizički svijet [4].

3.1. Primjeri i primjena IoT-a

Gotovo svaka fizička stvar se može transformirati u IoT uređaj ako ga se može povezati s internetom za kontrolu ili prenošenje informacija. Žarulja koja se može uključiti pomoću aplikacije na pametnom telefonu IoT je uređaj, kao što je IoT i senzor pokreta ili pametni termostat. Neki veći objekti mogu se sami napuniti s mnogim, manjim IoT komponentama, poput mlaznog motora koji je ispunjen tisućama senzora, koji prikupljaju i prenose podatke kako bi bili sigurni da djeluju učinkovito. U još većem obujmu projekti pametnih gradova pune čitave regije senzorima kako bi pomogli da ljudi bolje razumiju i kontroliraju okoliš. Izraz IoT uglavnom se koristi za uređaje za koje se obično ne očekuje da imaju internetsku vezu i koji mogu komunicirati s mrežom neovisno o ljudskom djelovanju. Iz toga se razloga PC općenito ne smatra IoT uređajem, a ni pametni telefon - iako je natrpan senzorima. Na primjer, pametni sat, *fitness* traka ili drugi nosivi uređaj mogu se računati kao IoT uređaji [4].

IoT može biti osoba s implantatom za nadzor srca, domaća životinja s biočip transponderom, automobil koji ima ugrađene senzore koji upozoravaju vozača kada je tlak u gumama nizak, odnosno bilo koja stvar kojoj se može dodijeliti adresa internetskog protokola (IP) i može prenositi podatke putem mreže [5].



Slika 2: Pametne perilice kojima se može pratiti status pranja
https://www.reddit.com/r/IoT/comments/68snk5/this_japanese_hotel_room_tv_has_a_channel_for/

IoT se koristi u **više mjesta no što bi prosječan čovjek mislio**:

- Grijanje, ventilacija i klimatizacija, požari, sigurnost, zaštita, rasvjeta, prijevoz
- Vjetrenjače, generatori, brojila, turbine, gorive ćelije, bušilice
- Uređaji, televizori, igraće konzole, alarmi, brave, rasvjeta
- Transporteri, montaža, pakiranje, pumpe, ventili, spremnici
- Oznake, prodajna mjesta, registri, natpisi, automati
- Hitna pomoć, vojna vozila, okoliš, nadzor
- Prekidači, usmjerivači, pohrana, telefonski sustavi, poslužitelji
- Implantati, kirurška oprema, monitori, pumpe, magnetska rezonancija, tablete [7].

3.2. Povijest IoT-a

Ideja o dodavanju senzora i inteligencije u obične stvari bila je predmetom rasprava tijekom 1980-ih i 1990-ih (a postoje i neke mnogo ranije), ali osim nekih ranih projekata (uključujući i automate za prodaju povezanih internetom) napredak je bio spor jer tehnologija nije bila spremna. Čipovi su bili glomazni te nije bilo načina da stvari učinkovito komuniciraju. Potrebni

su bili procesori, koji su jeftini i dovoljno štedljivi, prije nego što je to konačno postalo ekonomično za povezivanje milijardi uređaja. Dolaskom RFID oznaka (čipovi male snage koji mogu bežično komunicirati) riješio se dio problema, zajedno sa sve većom dostupnošću širokopojasnog interneta i mobilnoga i bežičnog umrežavanja. Adaptacijom IPv6, koji bi između ostalog trebao osigurati dovoljno IP adresa za svaki uređaj koji postoji na svijetu, također je bio nužan korak za širenje IoT-a [4].

Kevin Ashton skovao je frazu *Internet of things* 1999. godine, iako je bilo potrebno najmanje još jedno desetljeće da tehnologija dostigne njegovu viziju [8].

Autor navedene sintagme navodi da "IoT integrira međusobnu povezanost ljudske kulture naših stvari s međusobnom povezanošću našeg digitalnog informacijskog sustava interneta. To je IoT [9]."

Slika 3: Kevin Ashton, poznat kao *Otac IoT-a*
[10]

Dodavanje RFID oznaka na skupe dijelove opreme, radi praćenja njihove lokacije, bila je jedna od prvih IoT aplikacija. Ali od tada troškovi dodavanja senzora i internetske veze na stvari i dalje padaju, a stručnjaci predviđaju da bi ta osnovna funkcionalnost jednog dana mogla koštati svega 10 centi, što omogućava povezivanje gotovo svega na internet. IoT je u početku bio najzanimljiviji za poslovanje i proizvodnju, gdje se njegova primjena ponekad naziva i *machine-to-machine* (M2M), ali naglasak je sada na punjenju suvremenih domova i ureda pametnim uređajima. Rani prijedlozi za uređaje povezane s internetom uključuju *blogjects* (objekte koji *blogiraju* i bilježe podatke o sebi na internetu) [4].

3.3. Veličina IoT-a

Koliko je zapravo veliki IoT? Veliki je i sve veći; u svijetu ima više povezanih stvari nego ljudi. Tehnološka analitička tvrtka IDC predviđa da će do 2025. biti ukupno 41,6 milijardi po-

vezanih IoT uređaja. Također sugerira da industrijska i automobilska oprema predstavljaju najveću priliku povezanih "stvari", ali također takvu povezanost snažno usvajaju pametni domovi i nosivi uređaji bliske budućnosti [11].

Gartner, drugi tehnološki analitičar, predviđa da će automobilski sektor ove godine činiti 5,8 milijardi uređaja, što je gotovo četvrtina u odnosu na 2019. Uslužni će uređaji biti najveći korisnik IoT-a zahvaljujući kontinuiranom predstavljanju pametnih brojila. Sigurnosni uređaji, u obliku otkrivanja uljeza i internetskih kamera, bit će zaslužni za drugu najveću upotrebu IoT uređaja. Automatizacija zgrada, poput povezane rasvjete, bit će najbrže rastući sektor, a slijede ju automobilski (povezani automobili) i zdravstvena zaštita (nadzor kroničnih stanja) [12].

Tablica 1: IoT tržište po segmentima, 2018. -2020., Širom svijeta (milijarde jedinica)

Segment	2018	2019	2020
Usluge	0.98	1.17	1.37
Vlada	0.40	0.53	0.70
Automatizacija građenja	0.23	0.31	0.44
Fizičko osiguranje	0.83	0.95	1.09
Proizvodnja i prirodni resursi	0.33	0.40	0.49
Automobili	0.27	0.36	0.47
Pružatelji zdravstvenih usluga	0.21	0.28	0.36
Trgovina na malo i veliko	0.29	0.36	0.44
Informacije	0.37	0.37	0.37
Transport	0.06	0.07	0.08
UKUPNO	3.96	4.81	5.81

(Izvor: Gartner, Kolovoz 2019)

3.4. Prednosti IoT-a

Prednosti IoT-a za poslovanje ovise o konkretnoj implementaciji; fleksibilnost i efikasnost kao najčešće glavni aspekti. Ideja je da poduzeća trebaju imati veći pristup podacima o svojim proizvodima i vlastitim unutarnjim sustavima, kao i veću mogućnost uvođenja promjena kao rezultat korištenja IoT-a. Proizvođači dodaju senzore komponentama svojih proizvoda kako bi mogli natrag prenijeti podatke o njihovim performansama. Ovo može pomoći tvrtkama da uoče kada neka komponenta ne uspije i da je zamijene prije no što nanese štetu. Tvrtke također mogu koristiti podatke koje generiraju ti senzori, kako bi učinili svoje sustave i opskrbne lance učinkovitijim jer će imati puno točnije podatke o onome što se zaista događa. "Uvođenjem sveobuhvatnog prikupljanja i analize podataka u stvarnom vremenu, proizvodni sustavi mogu postati dramatično prilagodljiviji", kaže savjetnik McKinsey [13].

IoT obećava da će naše okruženje, naše domove, urede i vozila učiniti pametnijim te mjerljivijim. Pametni zvučnici, poput **Amazon Echo i Google Home**, olakšavaju reproduk-

ciju glazbe, postavljanje *timera* ili dobivanje informacija. Kućni sigurnosni sustavi olakšavaju nadgledanje onoga što se događa unutar i izvan nje ili razgovor s posjetiteljima, a pametni termostati mogu pomoći da se domovi zagriju prije nego što se ukućani vrate kući, a pametne žarulje mogu učiniti da domovi izgledaju kao da je netko kod kuće, čak i kada nema nikoga. Senzori mogu pomoći da se shvati koliko može biti bučno ili zagađeno određeno okruženje. Automobili koji voze samostalno i pametni gradovi mogli bi promijeniti način na koji se grade javni prostori i kako se njima upravlja. Međutim, mnoge od ovih inovacija mogle bi imati velike posljedice na osobnu privatnost pojedinca. Za potrošače, pametni je dom vjerojatno ondje gdje će vjerojatno doći u kontakt sa stvarima koje imaju internet, a to je jedno područje u kojemu se velike tvrtke (posebice Amazon, Google i Apple) natječu. Najočitiji su pametni zvučnici poput Amazon Echoa, ali tu su i pametni utikači, žarulje, kamere, termostati i pametni hladnjaci. No, osim što ljudi pokazuju svoje oduševljenje sjajnim novim uređajima, postoji i ozbiljnija strana aplikacija pametnih kuća. Možda će moći pomoći starijim osobama da ostanu neovisne i duže u svojim domovima tako što će obitelji i njegovateljima olakšati komunikaciju s njima i pratiti u kakvom su zdravstvenom stanju. Bolje razumijevanje načina funkcioniranja naših domova i mogućnost podešavanja tih postavki moglo bi uštedjeti energiju - na primjer, **smanjujući troškove grijanja** [4].

Slika 4: Pametan zvučnik Amazon Echo
[14]

3.5. Sigurnost i arhitektura IoT-a

Sigurnost je jedan od najvećih problema IoT-a. Senzori prikupljaju u mnogim slučajevima izuzetno **osjetljive podatke**, na primjer ono što se govori i radi u vlastitom domu. Čuvanje sigurnosti presudno je za povjerenje potrošača, ali do sada je sigurnosni put IoT-a bio izuzetno

loš. Previše IoT proizvođača premalo razmišlja o osnovama sigurnosti, poput šifriranja podataka u tranzitu i u mirovanju. Nedostaci softvera otkrivaju se redovno, ali mnogim IoT uređajima nedostaje mogućnost zakrpanja, što znači da su u stalnoj opasnosti. *Hakeri* aktivno ciljaju IoT uređaje, kao što su usmjerivači i internetske kamere, jer im nedostatak sigurnosti olakšava kompromis i pretapa se u divovske *botnete* [4].

Nedostaci su dopustili *hakerima* otvoren pristup uređajima pametnih kuća poput hladnjaka, pećnica i perilica posuđa. Istraživači su pronašli stotinu tisuća internetskih kamera koje se mogu lako *hakirati*, dok su neke pametne satove za djecu povezane s internetom otkrile sigurnosne ranjivosti koje omogućuju *hakerima* da prate lokaciju korisnika, prisluškuju razgovore ili čak komuniciraju s korisnicima. Takvi rizici sve više zabrinjavaju vlade diljem svijeta. Vlada Velike Britanije objavila je vlastite smjernice za sigurnost potrošačkih IoT uređaja. Očekuje da uređaji imaju jedinstvene zaporke, da će tvrtke pružiti javnu kontaktnu točku kako bi svatko mogao prijaviti ranjivost (i na to će se reagirati) i da će proizvođači izričito navesti koliko će dugo uređaji dobivati sigurnosna ažuriranja. To je skroman popis, ali to je početak [4].

Kad troškovi izrade pametnih predmeta postanu zanemarivi, ti će problemi postati sve rašireniji i neizreciviji. Sve to vrijedi i u poslu, ali u poslu su ulozi još veći. Spajanje industrijskih strojeva s IoT mrežama povećava potencijalni rizik hakera koji otkrivaju i napadaju te uređaje. Industrijska špijunaža ili razorni napad na kritičnu infrastrukturu potencijalni su rizici. To znači da će tvrtke morati osigurati da su te mreže izolirane i zaštićene, a potrebno je šifriranje podataka uz sigurnost senzora, pristupnika i drugih komponenata. Međutim, trenutačno stanje IoT tehnologije teže to osigurava, a postoji i nedostatak dosljednog IoT-ovog sigurnosnog planiranja u organizacijama. To je vrlo zabrinjavajuće s obzirom na to da su *hakeri* dokazano spremni dirati u industrijske sustave koji su povezani na internet, ali su ostali nezaštićeni [4].

IoT premošćuje jaz između digitalnog i fizičkog svijeta, što znači da *hakiranje* u uređaje može imati opasne posljedice u stvarnom svijetu. Prodiranje u senzore koji kontroliraju temperaturu u elektrani moglo bi navesti operatera na donošenje katastrofalne odluke, a preuzimanje kontrole automobila bez vozača moglo bi također završiti nesrećom [4].

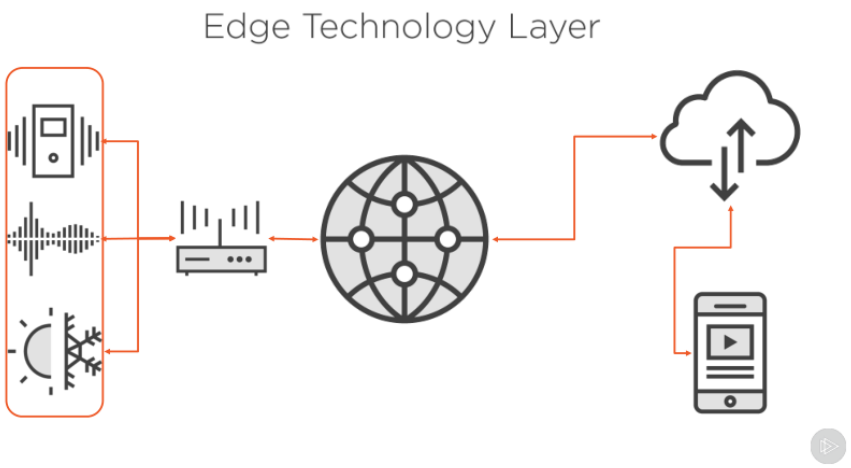
IoT uređaji često prodani sa zastarjelim operativnim sustav ili operativnim sustavom kojima fale zakrpe. Također, kupci često zaborave ili ne znaju promijeniti zadane šifre; u slučaju da promjene šifru, često su šifre slabe [7].

IoT uređaji tipično komuniciraju IoT prolazom (engl. *gateway*), ti IoT *gatewayovi* obično se koriste kao most između IoT uređaja i interakcije s krajnjim korisnikom. Mogu se zamisliti kao posrednika (engl. *middleman*). Uz to postoji i komunikacijski kanal, koji je zapravo internet, taj komunikacijski kanal spaja se s daljim *gatewayima* koji se nazivaju pohranom podataka i/ili serverima u oblaku (engl. *cloud server*). Tako se skupljaju podaci nakon što prođe kroz *gateway* i dođe na *cloud server*. Ti podaci pohranjuju se i analiziraju, te informacije mogu se tada prikazati krajnjem korisniku putem aplikacije koja se nalazi na njegovomu pametnom uređaju [7].

3.5.1. Arhitektura IoT-a

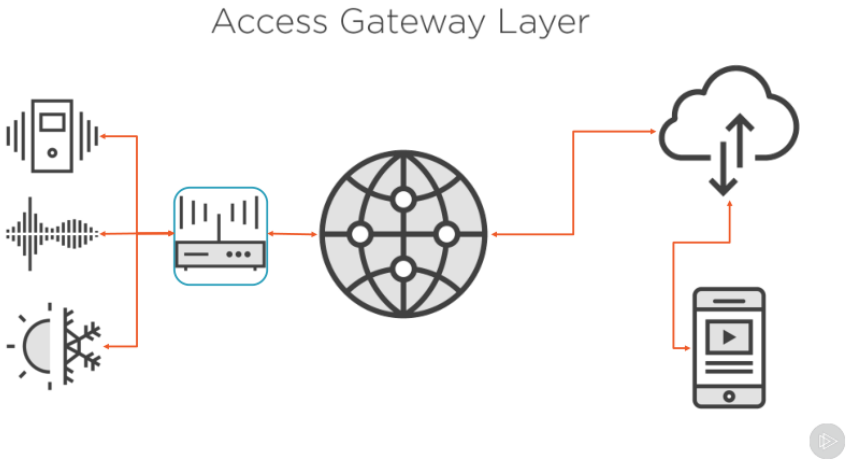
Iz perspektive hakera, postoji **mnogo mogućnosti napada** jer arhitektura IoT-a sastoji se od mnogo slojeva [7].

Prvi je sloj ***Edge Technology Layer***, u tom su sloju svi hardverski dijelovi poput senzora, čitača, softverski senzori, sami fizički uređaji, RFID oznake i tako dalje. Bilo koji uređaj koji bi koristili smješten je u *edge technology layer*. Uređaji su dizajnirani da bi skupili podatke koje će poslati natrag na *cloud server* ili na bazu podataka [7].



Slika 5: Arhitektura IoT-a, prvi sloj
[7]

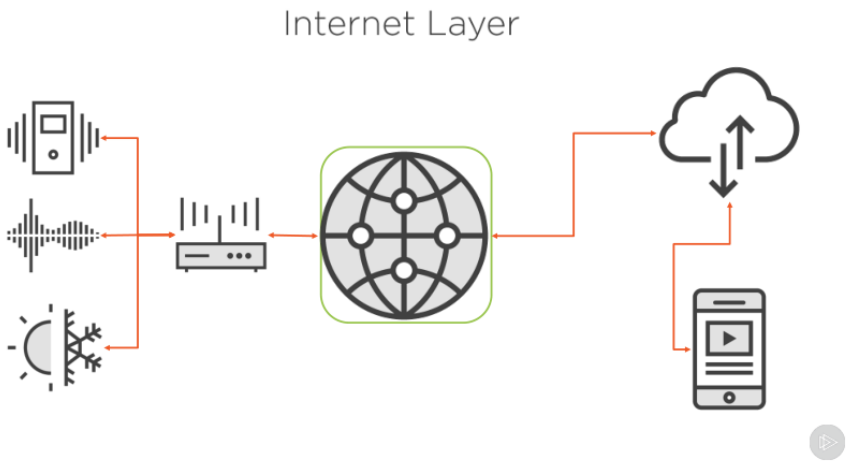
Sljedeći je drugi sloj, ***Gateway Layer***. On zapravo stvara most, odnosno konekciju između uređaja i interneta [7].



Slika 6: Arhitektura IoT-a, drugi sloj
[7]

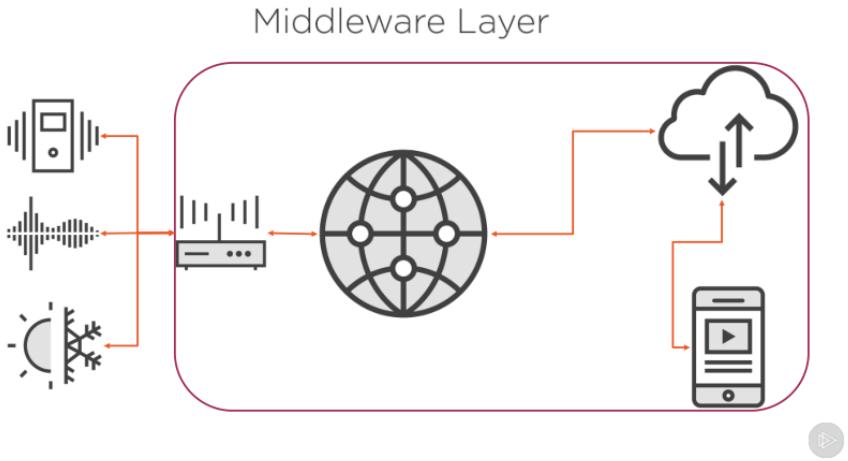
Treći je sloj ***Internet Layer*** koji je nužan za komunikaciju. Ovisno o proizvođaču, postoje

dvije različite krajnje točke (engl. *end points*) koje će *internet layer* koristiti [7].



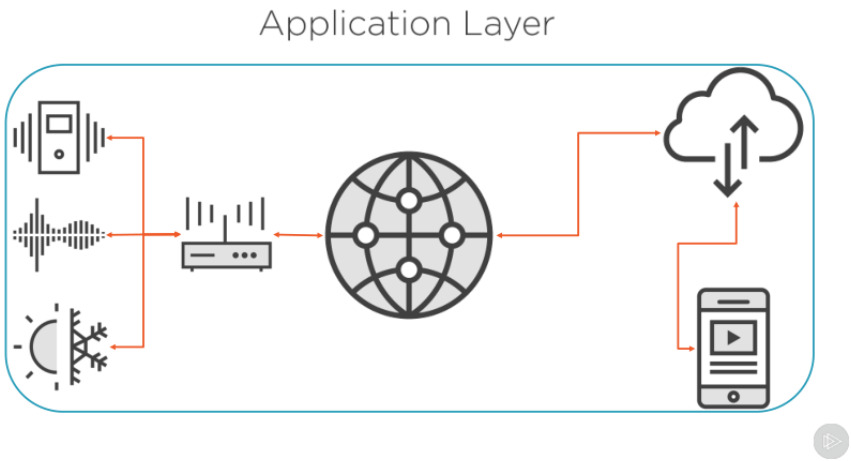
Slika 7: Arhitektura IoT-a, treći sloj
[7]

Također postoji i ***Middleware layer*** koji je obostrani komunikacijski kanal. Ovaj sloj stoji između aplikacijskog sloja i hardverskog sloja. Glavne su mu zadaće: upravljanje podacima, analiza podataka, agregacija podataka, filtriranje uređaja, otkrivanje uređaja i kontrola pristupa (korisnik može izabrati tko može pristupiti uređajima putem aplikacije) [7].



Slika 8: Arhitektura IoT-a, četvrti sloj
[7]

Aplikacijski sloj (engl. *Application Layer*) osigurava da se dostave servisi korisnicima iz različitih područja kako bi imali pristup potrebnim stvarima [7].



Slika 9: Arhitektura IoT-a, peti sloj
[7]

3.5.2. Prijetnje i prepreke koje stvara IoT

Problemi:

- **Vrijednost podataka:**

  Hvatanje podataka povećava rizik jer *hakeri* razumiju vrijednost informacije koju mogu uzeti. IoT okruženje dopušta *hakerima* da dobiju pristup drugim uređajima i mrežama na koji su spojeni uređaji. Mnoge tvrtke spajaju uređaje na istu mrežu kao i ostala računala, što je veoma opasno.

- **Agregacija podataka:**

  Opet postoji mnogo uređaja koji komuniciraju s drugim resursima, koji su često izvan kontrole korisnika.

- **Integracija:**

  Također postoji problem s integracijom. Često se na integrciju gleda kao pozitivnu stvar, što najčešće i je. Ali kada se počnu integrirati uređaji koji nisu u potpuno sigurnom okruženju radi pokušaja kontroliranja, kada se to omogući uređajima, oni nisu vrlo sigurni na postojećoj mreži i integriraju se s aplikacijama. Potrebno je provjeravati ima li dostupnih zakrpa ili nadogradnje *firmwarea* [7].

OWASP 10 glavnih ranjivosti IoT-a:

1. **Slabe, lako pogodive ili hardkodirane lozinke;**

   Lako je shvatiti zašto se ova briga pojavljuje na prvom mjestu OWASP IoT 10 glavnih ranjivosti. Većina IoT uređaja, posebno oni koji dolaze s intenretskim sučeljima, nisu rekonfigurirani kako bi korisnicima omogućili promjenu zadanih lozinki, što ih čini ranjivima

na mnoštvo napada lozinkom. Zašto bi napadač trošio vrijeme pokušavajući zaobići druge sigurnosne kontrole ako se lozinka može lako pogoditi ili nametnuti?

2. **Nesigurne mrežne usluge;**

   Alati mrežne sigurnosti poput vatrozida, sustava za otkrivanje upada/sustava za sprječavanje upada (IDS/IPS), objedinjena su rješenja za upravljanje prijetnjama (UTM-ovi) i tako dalje. Oni su i dalje relevantni čak i kad IoT uređaji uđu u igru. IoT sigurnost često je ugrožena zbog neovlaštenog pristupa (zbog zadanih lozinki, otvorenih portova i slično) i potencijalno može dovesti do toga da se ti uređaji koriste kao dio *botneta*. *Botneti* se često koriste za izvršavanje prijetnji poput napada distribuiranog uskraćivanja usluge (DDoS) na ciljane internetske stranice ili mrežne resurse.

3. **Nesigurna sučelja ekosustava;**

   Sučelja poput *weba*, oblaka, mobilnog ili pozadinskog sučelje za programiranje aplikacija (engl. *Application Programming Interface - API*), koji vam omogućuju interakciju s pametnim uređajem, mogu imati ranjivosti u provedbi provjere autentičnosti; ili još gore, potpuni nedostatak istih, slabosti u šifriranju, filtriranje podataka it tako dalje. Ove sigurnosne pogreške mogu naposlijetku dovesti do ugrožavanja uređaja ili bilo koje povezane komponente.

4. **Nedostatak sigurnih mehanizama za ažuriranje;**

   Ovdje je najveća opasnost činjenica da mnogim IoT uređajima nedostaje mogućnost sigurnog ažuriranja. Primjerice, u Ujedinjenom Kraljevstvu novi zakon predviđa da proizvođači IoT uređaja trebaju osigurati minimalno vrijeme tijekom kojega će njihovi uređaji dobivati sigurnosna ažuriranja.

5. **Korištenje nesigurnih ili zastarjelih komponenata;**

   Upotreba zastarjelog softvera ili pozivanje na nesigurne biblioteke u kodu mogu dovesti do ugrožavanja ukupne sigurnosti proizvoda. Od nesigurnih prilagodbi operativnog sustava do korištenja ranjivih hardverskih ili softverskih komponenata treće strane, IoT ranjivosti uključuje sve ono što ubrizgava slabosti u uređaj, a može se koristiti kao ulazna točka ili iskoristiti za produženje napada.

6. **Nedovoljna zaštita privatnosti;**

   Odnosi se na nesigurno čuvanje osobnih podataka, obradu ovih podataka ili njihovo otkrivanje bez odobrenja korisnika. Studija Sveučilišta Cornell iz 2017. godine istražuje informacije koje pasivni promatrači (poput ISP-a) mogu prikupiti samo analizom IoT mrežnog prometa, čak i kada je taj promet šifriran. Privatnost podataka, posebno kada je riječ o IoT-u, počinje se rješavati kroz zakonodavne akcije. Osim gore spomenutih zabrinutosti, prikupljanje potrošačkih podataka bez izričitog pristanka problematično je. Prekomjerno prikupljanje i prekomjerno zadržavanje takvih podataka, posebno danas, kada je IoT tako velik dio svakodnevice, također može dovesti do ugrožavanja sigurnosti korisnika u fizičkom svijetu.

7. **Nesiguran prijenos i pohrana podataka;**

   U ovom trenutku održavanje sigurnosti podataka moglo bi se činiti očitim prioritetom, s tim da stručnjaci neprestano upozoravaju na šifriranje, klasifikaciju podataka i pravilno postupanje s osjetljivim informacijama, ali s obzirom na prodor podataka koji se i dalje svakodnevno se vidjeti u medijima, nije ni čudo zašto je to još uvijek aktualna tema. Uz ograničenje pristupa osjetljivim podacima općenito, presudno je osigurati da se podaci šifriraju kada miruju, u tranzitu ili u obradi. Ako se šifriranje ne primijeni strogo, podaci postaju ranjivi te postaju glavni izvor moguće štete za IoT, u slučaju da nedostaju pametnim uređajima.

8. **Nedostatak upravljanja uređajima;**

   Kao što je važno znati koja su sredstva na određenoj mreži, jednako je važno njima učinkovito upravljati. Bez obzira na veličinu uređaja ili njihove pojedinačne troškove, ako oni komuniciraju s mrežom i imaju pristup njoj, metodičko upravljanje njima treba biti jedan od prioriteta. Uključivanje najboljih praksi upravljanja mrežnom sigurnošću, ali i upravljanje ažuriranjem radi osiguranja razgradnje, praćenja sustava i tako dalje, trebalo bi biti sastavni dio procesa. Neuspješno upravljanje IoT uređajima (oslanjanje na stare metode, poput praćenja sredstava pomoću Excel proračunskih tablica) može ugroziti cijelu mrežu.

9. **Nesigurne zadane postavke;**

   Zadane lozinke ili konfiguracije uređaja na pametnim uređajima često nisu sigurne. Iako je ponekad uzrok tomu nemar od strane korisnika i stoga ne mijenjaju zadane postavke. U drugim slučajevima nije moguće mijenjati postavke sustava, poput *hard* kodiranih lozinki, izloženih usluga s pokrenutim *root* dopuštenjima i tako dalje. Srećom, neki se zakonodavci bore protiv tih nesigurnih praksi. Primjerice, Kalifornija ima zakon koji od proizvođača IoT uređaja zahtijeva postavljanje jedinstvenih, unaprijed programiranih lozinki ili zahtijeva od korisnika da promijene lozinke prije upotrebe uređaja.

10. **Nedostatak fizičkog otvrdnjivanja;**

    Otvrdnjavanje uređaja od fizičkih napada štiti ga od pokušaja zlonamjernih korisnika da dobiju osjetljive podatke, koje kasnije mogu iskoristiti za pokretanje udaljenog *hakiranja* ili stjecanje kontrole nad uređajem. Na primjer: *portovi* za otklanjanje pogrešaka, koji se obično ne uklanjaju ili onemogućuju, čine uređaje ranjivima za pristup hakeru. Jednostavno uklanjanje memorijske kartice radi čitanja njenog sadržaja može otkriti lozinke ili druge osjetljive podatke. Korištenje sigurnog pokretanja pomaže u provjeri valjanosti *firmwarea* i osigurava da na uređaju može raditi samo pouzdan softver [15].

3.5.3. Protumjere za alate i metode hakiranja

Što se tiče IoT-a i protumjera koje se mogu provesti, prvo i očito moraju se **riješiti zadane postavke**, bilo to da su to zadane lozinke ili zadani računi, *guest račun* ili račun koji je naveden kao *demo*, a koji bi mogao biti problem [7].

Također treba osigurati da uređaj koristi **značajke zaključavanja**, tako da ako netko više puta pogrešno unese korisničko ime ili lozinku, automatski se zaključa [7].

Potrebno je koristiti jaku **provjeru autentičnosti**, treba biti siguran da se ona implementira na svim IoT uređajima [7].

Neke od osnova također uključuju stvari poput osiguranja da se **izoliraju i zaštite** ti uređaji od postojeće *production* mreže, a budući da su na vlastitim mrežama, trebat će im vlastiti sigurnosni uređaji poput **vatrozida** [7].

Naravno, predlaže se i da se na toj privatnoj mreži gdje su postavljeni IoT uređaji postave drugi uređaji, poput **sustava za otkrivanje upada** (engl. *Intrusion Prevention System - IPS*), kao i **sustave za sprječavanje upada** (engl. *Intrusion detection systems - IDS*) [7].

Šifriranje pomoću infrastruktura javnog ključa (engl. *Public Key Infrastructure - PKI*), kao i *end-2-end* enkripciju kroz komunikacijske kanale [7].

Implementiranje **VPN** rješenja, posebno za prelazak preko samog interneta do pristupnika ili čak do proizvođača, opet je cilj zbog osiguravanja komunikacije [7].

Valja pogledati i vidjeti je li moguće napraviti ***white* listu IP adresa** koje mogu ući u komunikaciju s ovim uređajima ili s kojim IP adresama ti uređaji smiju komunicirati. Ovo također neće nužno zaustaviti napadača jer postoji mogućnosti lažiranja IP adrese (engl. *IP spoofing*), ali to je samo jedan osnovni korak koji je u svakom slučaju dobro primijeniti [7].

Treba pripaziti i na **Telnet usluge**, mnogi od IoT uređaja koriste telnet usluge za daljinsko upravljanje njima. Telnet je aplikacijski protokol koji se koristi na internetu ili lokalnoj mreži za pružanje dvosmjerne interaktivne komunikacije, usmjerene na tekst pomoću virtualne terminalne veze [16]. Tu je problem s *portom* 48101. Taj *port* koristio je Mirai *botnet*. Mirai *botnet* bio je zlonamjerni softver (engl. *malware*) koji je korišten za krađu BusyBox sustava [17]. BusyBox je jednostavno dio softvera ili je softversko rješenje koje se može vidjeti na puno IoT uređaja, a riječ je o uklonjenoj verziji Unix alata, koji su kombinirani u jednu izvršnu datoteku (engl. *executable*)), a može se vidjeti puno BusyBoxa unutar Linuxa, Androida kao i u FreeBSD-u [18]. Zanimljivo je za ovaj određeni *malware* to što, čim je instaliran, odmah blokira *portove* 22, 23 i 80, koji bi u osnovi zaključali korisnika s vlastitih uređaja, a napadač je tada u stanju to kontrolirati i poslati tekst na neko određeno internetsko mjesto [7].

Stoga valja **isključiti univerzalni PnP ili *plug-in play***. Puno je ovih IoT uređaja napravljeno tako da budu prikladni ili jednostavni za upotrebu i tu se gubi sigurnosna strana. Uređaji poput video kamera zapravo koriste UPnP za razgovor s ruterom i prihvaćanje vanjskih veza [7].

Poželjno je osigurati **zaustavljanje fizičkog pristupa** jer ako napadač ima fizički pristup uređajima, vi više niste njihovi vlasnici [7].

Obavezno treba **instalirati zakrpe i ažuriranje softvera**. Mnogi pristupi ili napadi mogu se zaustaviti jednostavnim instaliranjem zakrpa i ažuriranja, ne samo na *gatewayove*, već i na fizičke IoT uređaje te naravno i na bilo koju vrstu ažuriranja za aplikacije na mobilnim uređajima [7].

3.6. Privatnost IoT-a

Sa svim senzorima koji skupljaju podatke o svemu što radimo, IoT je potencijalno veliki problem zbog mnogih slabosti privatnosti i sigurnosti. Neki nedostaci pametnog doma: on zna kada se osoba budi koju radio stanicu sluša (zahvaljujući pametnom zvučniku), kakvu vrstu hrane jede (zahvaljujući pametnoj pećnici ili hladnjaku), što njegova djeca misle (zahvaljujući njihovim pametnim igračkama) i tko posjećuje i prolazi pored kuće (zahvaljujući pametnom zvonu na vratima). Iako će tvrtke zaraditi od prodaje pametnih uređaja, njihov IoT poslovni model vjerojatno uključuje **prodaju** barem nekih podataka [4].

Što se događa s tim podacima, izuzetno je važno za privatnost. Ne grade sve tvrtke svoj poslovni model oko prikupljanja i prodaje podataka, ali neke to čine [4].

Valja zapamtiti da se IoT podaci mogu kombinirati s drugim dijelovima podataka kako bi se stvorila iznenađujuće detaljna slika o korisnicima. Iznenađujuće je lako saznati mnoštvo detalja o nekoj osobi iz nekoliko različitih očitavanja senzora. U jednom projektu, jedan istraživač je otkrio da su analizom podataka u kojima se bilježila samo potrošnja energije, razine ugljičnog monoksida i ugljičnog dioksida, temperatura i vlaga tijekom dana mogli utvrditi što je ta osoba imala za večeru [4].

Potrošači moraju razumjeti razmjenu koju obavljaju i jesu li zadovoljni s tim. Neka se ista pitanja odnose na poslovanje: bi li vaš izvršni tim rado raspravljao o spajanju u sobi za sastanke, na primjer, opremljenim pametnim zvučnicima i kamerama? Jedno nedavno istraživanje pokazalo je da četiri od pet tvrtki neće moći identificirati sve IoT uređaje na svojoj mreži [4].

Loše instalirani IoT proizvodi mogu lako otvoriti korporativne mreže za napad od strane hakera ili jednostavno mogu lakše **propustiti podatke**. To može izgledati kao trivijalna prijetnja, ali u slučaju da se *pametna brava* u uredu odbije otvoriti jedno jutro, *hakeri* to mogu iskoristiti za stvaranje *backdoora* na mreži [4].

3.7. Primjena umjetne inteligencije u IoT-u

IoT uređaji generiraju **ogromne količine podataka**. To mogu biti podaci o temperaturi motora, o tome jesu li vrata zatvorena ili podaci očitani s pametnog brojila. Sve ove IoT podatke treba prikupiti, pohraniti i analizirati. Jedan od načina na koji tvrtke maksimalno iskorištavaju te podatke jest njihovo unošenje u sustave umjetne inteligencije, koji će uzeti te IoT podatke i koristiti ih za predviđanje [4].

Na primjer, Google je postavio AI zadužen za svoj sustav hlađenja podatkovnog centra. AI koristi podatke iz tisuće IoT senzora koji se dovode u duboke neuronske mreže i koji predviđaju kako će različiti izbori utjecati na buduću potrošnju energije. Korištenjem strojnog učenja i AI-a, Google je uspio poboljšati svoje podatkovne centre i utvrdio da bi ista tehnologija mogla imati koristi u drugim industrijskim okruženjima [4].

4. Umjetna inteligencija - AI

Umjetna inteligencija (engl. *artificial intelligence - AI*) grana je računarstva koja pokriva široko područje razvoja pametnih uređaja sposobnih za odrađivanje zadataka koji obično zahtijevaju ljudsku inteligenciju. Umjetna je inteligencija multidisciplinarna znanost s više mogućih pristupa rješavanju problema, koja razvojem strojnog učenja i dubokog učenja mijenja paradigme u doslovno svakom području moderne tehnološke industrije [19].

Umjetna inteligencija omogućava strojevima da **uče iz iskustva**, prilagođavaju se novim situacijama i novim ulaznim podacima i tako odrađuju zadatke na sličan način kao i ljudi. Moguće je čuti o primjerima primjene UI – od igranja šaha do samostalnih vozila - koji se oslanjaju na strojno učenje i na prirodno jezično procesuiranje [20].

Upotrebom ovih tehnologija računala mogu biti istrenirana za odrađivanje određenih zadataka tako da procesuiraju ogromne količine podataka i prepoznaju uzorke i trendove unutar podataka [20].

Uglavnom je umjetna inteligencija potaknuta razvojem strojnog učenja, a za razlučivanje pojmova UI, strojnog učenja i dubokog učenja te za daljnje razumijevanje važno je znati:

- Umjetna je inteligencija **skup algoritama** koji pokušava imitirati ljudsku inteligenciju
- Strojno je učenje samo **jedan od algoritama**
- Dubinsko je učenje jedna od **tehnika strojnog učenja** [19].

Pojednostavljeno, strojno učenje **hrani računalo podacima** i pomoću statističkih tehnika pomaže mu da **"uči"** i s vremenom postaje sve bolje u izvršavanju zadataka, a da nije ciljano isprogramirano da izvrši ciljani zadatak, čime se izbjegava pisanje kompleksnih programski kodova. Strojno učenje može biti s nadzorom i bez nadzora, ovisno o tome kako se označavaju setovi podataka [19].

Duboko je učenje **tip strojnog učenja** koje je nadahnuto biološkom neuronskom mrežom. Neuronske mreže sadrže nekoliko skrivenih slojeva kroz koje su podatci procesuirani i time se omogućava računalo da ide "dublje" u svom učenju i povezivanje naočigled nepovezanih podataka [19].

Nekoliko najraširenijih primjera umjetne inteligencija danas su:

- Google Search
- Prepoznavanje fotografije (što je i tema ovog rada)
- Siri, Alexa i drugi osobni asistenti na mobilnim uređajima
- Autonomna vozila [19].

4.1. Važnost umjetne inteligencije

Umjetna inteligencija **automatizira** učenje iz ponavljanja i analize podataka. Umjetna inteligencija razlikuje se od automatizacije pogonjene sklopovima. Za razliku od sklopova koji automatiziraju jednostavne zadatke, UI odrađuje učestale, ubrzane računalne procese na velikoj količini podataka i bez zamora. Za ovaj način automatizacije i dalje je bitan ljudski način razmišljanja da bi se uspostavio sustav i postavila se prava pitanja [20].

Umjetna inteligencija postojećim uređajima **dodaje inteligenciju**, što znači da se na tržištu ne nudi kao samostalan proizvod, već kao nadogradnja postojećim uređajima i tako im unaprjeđuje uporabu, kao što se na primjer Siri dodaje na postojeće Apple uređaje. Sva postojeća rješenja mogu se kombinirati s velikom količinom podataka i tako unaprijediti tehnologije koje se koriste u kućanstvima i na radnim mjestima – od sigurnosti do analize financijskih tržišta [20].

Umjetna se inteligencija kroz napredne algoritme učenja prilagođava tako da sami podaci razvijaju programsko rješenje. UI prepoznaje strukturu i trendove unutar podataka i tako sam algoritam postaje sposoban za razvrstavanje ili predviđanje. Na taj se način algoritam sam može naučiti odrađivanju kompleksnih zadataka, kao igranju šaha ili koje proizvode nuditi *online*. Učenjem iz iskustava UI prilagođava se i vježba nove mogućnosti u slučajevima kada pogriješi iz prvog pokušaja [20].

Korištenje neutralnih mreža omogućuje umjetnoj inteligenciji dublje analize veće količine podataka i time odrađuje zadatke koji su još prije pet godina bili **nezamislivi**. Proboj UI počeo je porastom računalnih snaga i pojavom *Big Data* sustava, koji su omogućili računalima da uče direktno iz samih podataka. Što većom količinom podataka strojevi barataju, to su precizniji u izvršavanju zadataka. Sva moderna UI rješenja baziraju se na dubokom učenju i postaju sve točnija tijekom same uporabe [19].

Namjena umjetne inteligencije **nije da zamijeni ljude**. Ona proširuje ljudske mogućnosti, a time unaprjeđuje kvalitetu ljudskog rada. Zbog načina na koji strojevi uče, a takav je način drugačiji od ljudskog načina učenja, njihov je pristup rješavanju problema također drugačiji [20].

Strojevi vide uzorke i poveznice koje su nedokučive ljudskim kognitivnim sposobnostima, a time u suradnji s ljudima otvaraju prilike za suradnju na mnogim područjima:

- Uvode analitiku u grane industrije gdje podaci trenutno nisu potpuno iskorišteni
- Poboljšavaju trenutne tehnike analiza
- Ruše ekonomske i jezične prepreke
- Daju kvalitetniji uvid, razumijevanje i mogućnosti pamćenja velikih količina podataka [20].

4.2. Povijest umjetne inteligencije

Ideje o umjetnoj inteligenciji pojavljivale su se u književnosti i mitovima već od pradavnih vremena, prvi su pokušaji realizacije još u staroj Grčkoj, a njezin je razvoj započeo pojavom digitalnih računala u drugoj polovici 20. stoljeća. Među prve teoretske radove iz toga područja ubrajaju se radovi **A. M. Turinga**, dok se prvim računalnim programom umjetne inteligencije smatra **logički teoretičar** (engl. *The Logic Theorist; 1956*). Izraz umjetna inteligencija skovan je na kongresu na Dartmouth Collegeu 1956., koji je, okupivši pionire toga područja, potaknuo sustavna istraživanja. Taj je kongres organizirao John McCarthy, koji je potom postao autor osnovnoga programskog jezika umjetne inteligencije LISP-a (1958.), te ga se danas smatra začetnikom umjetne inteligencije. Nakon početnih optimističkih prognoza, u 1960-ima sva je složenost postizanja umjetne inteligencije koja bi bila mjerljiva s ljudskom izišla na vidjelo te su se istraživanja usmjerila prema parcijalnim rješenjima pojedinih problema. Tako je 1961. razvijen šahovski program koji je igrao na razini majstora, 1965. stroj koji se koristio rezolucijom kao metodom logičkoga zaključivanja, a iste godine započeo je rad na glasovitom ekspertnom sustavu Dendral [20].

Početkom 1970-ih razvijen je programski jezik PROLOG te potom i prvi zaista upotrebljivi ekspertni inteligentni sustavi, kakav je bio sustav MYCIN, koji je mogao dijagnosticirati bakterijske krvne infekcije i preporučiti liječenje. Taj je sustav u nekim slučajevima djelovao čak i bolje od liječnika, no zbog nedostatnih mogućnosti računala toga doba, znanje mu je bilo preusko. Rješenje je nađeno u umrežavanju više tisuća računalnih procesora, što je ostvareno projektom računala Connection Machine. To je računalo bilo prvi primjer umjetne neuronske mreže [20].

Početak 80-ih godina dvadesetog stoljeća i razvoj prvog komercijalnog uspješnog prototipa R1 (znanog i kao XCON) označava kraj oskudnog financiranja istraživanja i maleni napredak u umjetnoj inteligenciji, koji je poznat i kao *Zima UI-a*. Nakon toga slijedi uspon ekspertnih sustava – odnosno tehnologija umjetne inteligencije koje su obilježene specifičnim ciljevima realiziranih sintezom velikih količina podataka iz određenog područja ljudske djelatnosti, kao što je računalni šahovski šampion Deep Blue ili MYCIN program za medicinsku dijagnostiku [20].

Nakon ekspertnih sustava, dolazi period u kojemu se sada djeluje, a koje u jedan sustav povezuje pojmove poput strojnog učenja, vizualnog raspoznavanja znakova s mnogo širim područjima primjene, izgrađenih od raspodijeljenih komponenti. Danas su znanstvenici oprezniji u predviđanjima, pa se o UI-u govori kao o teoretskoj mogućnosti, dok se istovremeno mnogo radi na razvoju inteligentnih sustava koji upravljaju svakodnevnim ljudskim aktivnostima, poput pametnih agenata (robota ili softvera). Zato se može govoriti o sve većoj prisutnosti i praktičnoj primjeni tehnologija razvijenih u istraživanju umjetne inteligencije u današnjem društvu [20].

4.3. Tehnike strojnog učenja

Posljednjih je godina moderan pristup umjetnoj inteligenciji više usredotočen na *bottom-up* tehnike, odnosno na to da se uzmu osnovni gradivni blokovi inteligencije, koji se zatim pos-

tave zajedno u određene situacije te ih se ostavi da uče i da se razvijaju određeni vremenski period pa se pregledaju dobiveni rezultati. Situacijskim pristupom nastoji se ostvariti umjetna inteligencija koja je utjelovljena i smještena u stvarnom svijetu. Takav je pristup u istraživanju umjetne inteligencije nastao zadnjih dvadesetak godina, a temelji se na izgradnji **inteligentnih agenata (*botova*)**, koji se u svome okruženju ponašaju uspješno. Temelj ove metode dizajn je koji koristi elementarna ponašanja, koja se kombiniraju kako bi se ostvarila kompleksnija ponašanja. Novi pristup zagovara ideju da se inteligencija kod strojeva može ostvariti, ali kroz dovoljne motorne vještine i senzornu interakciju s okolinom, odnosno smještenost. Izraz smješten (engl. *situated*) nastao je u robotici i odnosio se na smještaj robota u okolini, no može se smatrati da se smještaj odnosi i na softverske agente, pod uvjetom da se oni nalaze u dinamičkom okruženju, da svojim ponašanjem mogu manipulirati i mijenjati okruženje koje su u stanju osjećati i percipirati. Naglasak je, dakle, na ponašanju i ne oslanja se na simbolički opis svijeta, već na kreiranju modela interakcija entiteta i njihove okoline. Smještajni pristup ulaže mnogo manje prioriteta apstraktnom zaključivanju ili vještinama koje zahtijevaju rješavanje problema. Inteligentno ponašanje entiteta postiže se u njegovoj interakciji s okolinom, kroz povezivanje jednostavnih procesnih elemenata koji rade paralelno (poput neurona u mozgu). To je temeljna ideja prema kojoj funkcioniraju i umjetne neuronske mreže pa se prema njihovoj implementaciji ovaj pristup naziva i konektivistički [21].

- Prvi temeljni koncept moderne umjetne inteligencije jest razmotriti način rada biološkog mozga u smislu osnovnih funkcija, razvoja i prilagodbe tijekom vremena
- Drugi se temelji na potrebi za dobivanjem relativno jednostavnih modela temeljnih elemenata, gradivnih elemenata, mozga
- Treće, te gradivne elemente potrebno je oponašati tehnološkim dizajnom – možda elektroničkim krugom, možda računalnim programom s ciljem da simuliraju gradivne blokove mozga. Umjetni gradivni blokovi tada se mogu zajedno priključiti na različite načine kako bi djelovali slično mozgu. Ovdje je u pitanju traženje inspiracije iz biološkog načina djelovanja kako bi ga se koristilo u tehnološkom dizajnu [21].

4.3.1. Umjetne neuronske mreže

Točan način na koji mozak omogućuje misao **jedna je od najvećih misterija znanosti**. Poznato je da je neuron ili živčana stanica osnovni funkcionalni dio tkiva živčanog sustava, uključujući mozak. Svaki neuron sastoji se od tijela stanice, koji se još naziva soma i u kojem je smještena stanična jezgra. Povezane spojnice neurona nazivaju se sinapse. Svaki neuron formira sinapse s ostalim neuronima, a može ih biti nekoliko desetaka do nekoliko stotina tisuća. Obično se neuron nalazi u stanju mirovanja, a signal koji prima u obliku je elektrokemijskog pulsa koji do njega dolazi od ostalih neurona. Svaki puls mijenja električni potencijal tijela stanice – neki doprinose signalnom potencijalu, dok ga neki umanjuju. Ako ukupni signal u bilo kojem trenutku dostigne vrijednost praga, tada će stanica „ispaliti“ elektrokemijski puls, koji se još naziva akcijski potencijal, u ostale neurone kako bi se i oni nakon toga pobudili. Kratko nakon toga neuron se vraća u stanje mirovanja i čeka da se ponovo izgradi puls. Obratno, ako

nije postignuta vrijednost praga, tada neuron neće „okinuti“. To je *sve ili ništa* proces u kojem neuron ili „okida“ ili ne [21]. Kako osoba uči, veze jačaju u njezinom mozgu (pozitivno) ili slabe (negativno) i tako čine da se osoba manje ili više ponaša na određeni način. Mozak je stoga izuzetno plastičan, tako da se prilagođava i funkcionira drugačije, ovisno o uzorcima signala koje prima i nagradama ili kaznama koje su s time povezani. Tako će mozak činiti manje pogrešaka. To je osnova na kojoj se temelje biološki rast i razvoj mozga i koja omogućuje mozgu izvođenje operacija. Ideje preuzete iz strukture biološke neuronske mreže i njezine metode učenja temeljne su sastavnice koje se primjenjuju u umjetnim neuronskim mrežama, kod kojih je cilj upotrijebiti tehnološka sredstva za ostvarenje nekih svojstava originalne biološke verzije. Opća definicija mogla bi glasiti:

- Neuronska mreža jest skup međusobno povezanih jednostavnih procesnih elemenata, jedinica ili čvorova, čija se funkcionalnost temelji na biološkom neuronu [21].

Neurofiziološka istraživanja, koja su omogućila bolje razumijevanje strukture mozga, daju naslutiti da je modelu mozga najsličniji model u kojem brojni procesni elementi podatke obrađuju paralelno. Područje računarstva koje se bavi tim aspektom obrade informacija zove se neuro-računarstvo, a paradigmu obrade podatka naziva se umjetnom neuronskom [21].

Posebna istraživanja rade se na području arhitekture računala koja bi, na način pogodniji od konvencionalne arhitekture, omogućila učinkovitu primjenu umjetne neuronske mreže. Pod postupkom učenja kod neuronskih mreža podrazumijeva se iterativan postupak predočavanja ulaznih primjera (uzoraka, iskustva) i eventualno očekivana izlaza. Ovisno o tome je li u postupku učenja unaprijed poznat i izlaz iz mreže, pri učenju mreže koristi se uz svaki ulazni primjer. Ili je točan izlaz nepoznat, razlikuju se dva načina učenja:

- učenje s učiteljem i
- učenje bez učitelja [21].

4.3.2. Duboko učenje

Tema o kojoj se sve više govori duboko je učenje. Ono predstavlja potkategoriju strojnog učenja, a realizira se umjetnim neuronskim mrežama koje pomažu u prepoznavanju govora, računalnog vida i obrade prirodnog jezika. Posljednjih godina duboko učenje doživljava **nagli rast**. Razvoj dubokog učenja pomogao je napretku u područjima poput percepcije objekata, strojnog prevođenja i prepoznavanja glasa – svaki je od njih predmet koji istraživači umjetne inteligencije dugo nastoje „probiti“. Ključni su faktori poboljšanja izvedbe algoritama dubokog učenja dostupnost velikih skupova podataka za treniranje, ostvareni masovnim povezivanjem računala, te povećanje kapaciteta memorije i brzine računala. Veće umjetne neuronske mreže mogu danas izvoditi kompleksnije zadatke s većom točnosti [21].

Pri klasičnom strojnom učenju računalo bi kreiralo znanje kroz nadzirano iskustvo, što znači da je ljudski operater pomagao stroju u učenju, dajući mu stotine ili tisuće praktičnih primjera za učenje, a greške su se ispravljale ručno. Razlika današnjih sustava s dubokim

učenjem u tome je što sada istraživači nastoje konstruirati sustav koji sam kreira svoje osobine, koliko god je to izvedivo. Dakle, duboko je učenje uglavnom bez nadzora. Uključuje, na primjer, neuronske mreže velikih razmjera koje omogućavaju računalu učenje i samostalno "razmišljanje", bez potrebe za izravnom ljudskom intervencijom [21].

4.3.3. Evolucijsko računalstvo

Tehnike pretraživanja, u smislu traženja rješenja, zadnjih su godina pronašle inspiraciju u proučavanju biološke evolucije. Evolucijsko računalstvo čini skup algoritama za pretraživanje, proizašlih iz tehnika umjetne inteligencije koji se temelje na teoriji evolucije. Ona pruža alternativnu, veoma moćnu strategiju kada se zahtjeva pretraživanje, po mogućnosti najboljeg, rješenja problema, odabirom iz niza mogućih rješenja. Moguće je, ako je potrebno, ostvariti nova rješenja koja nisu prije razmatrana, odnosno ostvariti kreativnost [21].

Oponašanjem (modeliranjem) prirodnog evolucijskog procesa u računalima, moguće je postići tehniku koja prilagođava rješenja problema iz populacije potencijalnih rješenja. Pri tome može se postići najbolje moguće rješenje ili barem rješenje koje funkcionira. Različita rješenja u jednoj generaciji pomiješana su genetički parenjem da bi proizveli novu, poboljšanu generaciju. Tako dobivena rješenja dalje se miješaju, na razne načine, da bi se ostvarila iduća generacija i tako dalje, sve dok se mnogo, ako je moguće tisuće, generacija kasnije ne dostigne mnogo bolje rješenje originalnog problema [21].

4.3.4. Inteligentni agenti

Agent je nešto što može percipirati svoju okolinu putem senzora (osjetila) i djelovati na tu okolinu putem efektora. Ljudski agent ima oči, uši i ostale organe za senzore te ruke, noge, usta i ostale dijelove tijela za efektore. Robotski agent zamjenjuje kamere i infracrvene lokatore za senzore, a različite motore za efektore. Softverski agent ima kodirane nizove bitova za perceptore i akcije koje izvodi [21].

Za interakciju agenta s okolinom putem senzora i efektora, koristi se izraz mjera izvedbe (engl. *performace measure - PM*) kod određivanja kriterija uspješnosti agenta. Pri tome se inzistira na objektivnom mjerenju, nametnutom od autoriteta. Drugim riječima, vanjski promatrači postavljaju standard o tome što znači biti uspješan u okolini i onda se taj standard koristi za mjerenje izvedbe agenata [21].

Ponašanje agenta ovisi isključivo o slijedu percepta do određenog trenutka. Tada je moguće opisati bilo kojeg agenta izradom tablica akcija koje se poduzimaju kao odgovor na svaki mogući slijed percepta. Takva se lista naziva mapa - od slijeda percepta do akcija. Idealno mapiranje tada opisuje idealnog agenta. Određivanjem akcija koje agent treba poduzeti kao odgovor na bilo koji slijed percepta omogućuje se kreiranje idealnog agenta. Ponašanje agenta može biti osnovano na njegovu vlastitom iskustvu, kao i na ugrađenom znanju prilikom izgradnje agenta za određeno okruženje u kojemu agent djeluje [21].

Zadaća tehnika umjetne inteligencije jest kreiranje agent programa (*botova*): funkcije

koja implementira mapiranje agenta od perceptora do akcija. Pretpostavka je da će se program izvoditi na nekoj vrsti računalne naprave koja se naziva arhitektura. Program se odabire tako da ga arhitektura prihvaća i izvodi. Arhitektura može biti jednostavno računalo ili može uključivati hardver za specijalne namjere, poput obrade fotografije iz kamere ili filtracije audio unosa. Može uključivati softver koji omogućava izolaciju temeljnog računala i agent programa, kako bi se moglo programirati na višoj razini [21].

Općenito; arhitektura čini perceptore iz senzora dostupnima programu, izvodi program, i dobavlja programske akcije efektorima, onako kako su one generirane. Pojavnost sveukupnog kompleksnog inteligentnog ponašanja postiže se kroz skup interakcija jednostavnih entiteta, koji su sami poluautonomni agenti [21].

Autonomnost sustava određena je mjerom kojom je njegovo ponašanje određeno vlastitim iskustvom. Zaista autonoman inteligentni agent trebao bi samostalno djelovati uspješno u različitim vrstama okruženja ako ima dovoljno vremena za prilagodbu. Autonomni je agent sustav koji je smješten u okolinu te je dio te okoline koju osjeća i na koju djeluje s ciljem ostvarenja vlastitih planova. Posljednja osobina koja je navedena razlikuje autonomne agente od ostalog softvera. Pretpostavka je da bi određeni generalno inteligentni sustav trebao biti autonomni agent jer je za generalizaciju znanja, među različitim i vjerojatno novim područjima, potrebno učenje [21].

Jedan pristup umjetne inteligencije posebno se fokusira na ideju agenata i njihove individualne identitete, kako bi se proizvelo ukupno nastalo ponašanje. Računalni agenti veoma su aktualna tema u umjetnoj inteligenciji, kao i prijelomna točka u razvoju softvera nove generacije. Postoji širok spektar mogućih softverskih agenata. Na primjer, takvi se agenti danas koriste pri promatranju financijskih tržišta [21].

4.3.5. Veliki podaci

U istraživanjima na području astronomije i genetike javlja se tijekom 2000-ih eksplozija velike količine sakupljenih podataka. Iz genomike nastao je koncept velikih podataka (engl. *Big Data*) i proširio se na sva područja ljudske djelatnosti [21].

U srži pojma *Big Data* nalazi se **predviđanje**. Iako se opisuje kao grana računalne znanosti i umjetne inteligencije, odnosno područja strojnog učenja, ova je osobina zavaravajuća. *Big Datom* ne pokušava se naučiti računalo da misli poput ljudi. Tu se radi o primjeni matematike na goleme količine podataka ne bi li se dobio zaključak temeljen na **vjerojatnosti**. Ključno je da ovakvi sustavi imaju dobre rezultate zbog toga što su hranjeni s mnogo podataka na kojima temelje svoja predviđanja. Štoviše, stvoreni su tako da sami sebe poboljšavaju s vremenom, da bilježe najbolje signale i uzorke za promatranje dok se hrane novim podacima [21].

4.4. Prijetnje umjetne inteligencije i sigurnosni problemi

Ironično, brzina je također glavni nedostatak umjetne inteligencije. *Hakeri* prihvaćaju algoritme strojnog učenja koji stoje iza uspjeha tehnologije kako bi stvorili napade personalizirane za određene pojedince. Budući da se AI može "podučavati" pomoću skupova podataka, *hakeri* mogu ili stvoriti vlastite programe ili manipulirati postojećim sustavima u zlonamjerne svrhe. Napadi izvršeni pomoću AI-a obično su uspješniji, možda zato što tehnologija olakšava razvoj zlonamjernog softvera s mogućnošću izbjegavanja čak i sofisticiranog otkrivanja prijetnji. Na primjer, uparivanje polimorfnog zlonamjernog softvera s AI omogućuje tim programima brzu promjenu koda, čineći ih gotovo neranjivima za postojeće sustave kibernetičke sigurnosti [22].

Hakeri također mogu modificirati algoritme strojnog učenja u poduzeću, mijenjajući ulaze kako bi promijenili način na koji sustav prepoznaje određene elemente. Ova se tehnika može koristiti kako bi sustav previdio prijetnje i omogućio *hakerima* da zaobiđu kontrole upravljanja identitetom i pristupom [22].

Ponašanja sustava također su potencijalne mete; s pravim izmjenama, *hakeri* mogu promijeniti način na koji uređaji reagiraju ili komuniciraju, što može rezultirati opasnim ishodima. Jednom kada se promijene informacije o sustavu, može biti vrlo teško ispraviti probleme i vratiti mrežu u prvotno stanje [22].

U svjetlu ovih prijetnji, važno je da se rukovoditelji poduzeća i IT profesionalci odupru iskušenju da budu zadovoljni. Iako AI postaje sve autonomniji, on nikako nije zamjena za ljudsku marljivost. Sustavi od početka zahtijevaju ispravno postavljanje i upravljanje, počevši od opsežnih skupova podataka, kako bi se spriječili lažni pozitivi, nastavljajući s dosljednim nadzorom i ažuriranjima da bi se održala jaka sigurnost [22].

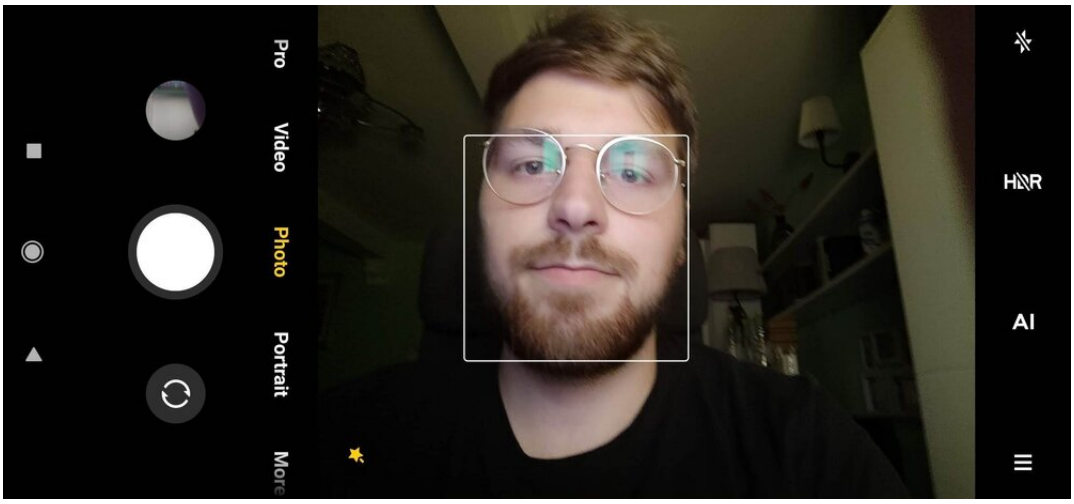
4.5. Prepoznavanje lica - korak po korak

U ovom projektu za izgradnju vlastitog *face recognition* sustava u **Pythonu** potrebne su `OpenCV` i `dlib` biblioteke. Prepoznavanje lica može se svesti ugrubo na par problema/koraka:

1. U slici je potrebno pronaći sva lica
2. Fokusirati se na svako lice i razumjeti da ako je lice okrenuto u bilo kojem smjeru ili ako je loše osvijetljeno, da je to svejedno ista osoba
3. Treba izabrati jedinstvene značajke lica kako bi se mogla razlikovati osoba od drugih osoba; na primjer veličina očiju, lica i slično
4. Sve jedinstvene značajke lica treba usporediti sa svim licima u bazi, kako bi se mogla prepoznati osoba [23]

4.5.1. Pronalaženje svih lica

Prvi je korak **otkrivanje lica**. Potrebno je pronaći lica na fotografiji prije no što se započne proces raspoznavanja! Otkrivanje lica na djelu može se vidjeti na većini fotoaparata danas. Otkrivanje lica izvrsna je značajka za kamere. Kada kamera može automatski razabrati



Slika 10: Otkrivanje lica na fotoaparatu

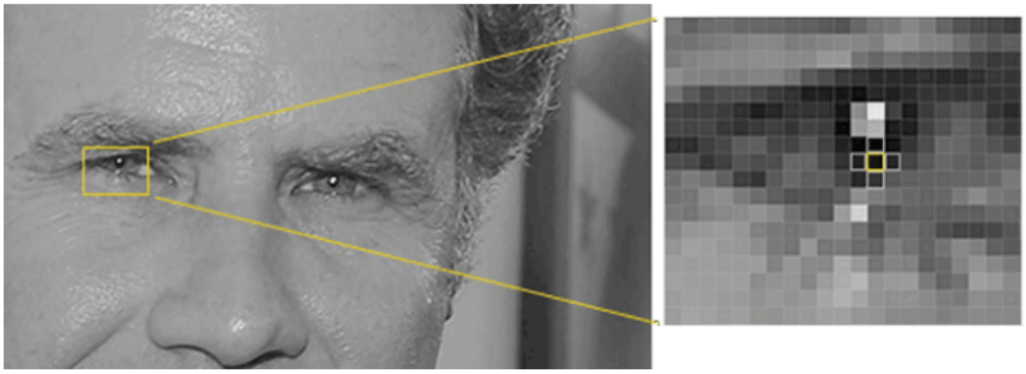
lica, može osigurati da su sva lica u fokusu prije nego što fotografira. Za ovaj projekt koristit će se ta značajka za pronalaženje područja fotografije te će se prenijeti na sljedeći korak [23].

Otkrivanje lica postalo je uobičajeno početkom 2000-ih kada su Paul Viola i Michael Jones izumili način otkrivanja lica koji je bio dovoljno brz da se može pokretati i na jeftinim kamerama. Međutim, sada postoje mnogo pouzdanija rješenja. Upotrijebit će se HOG metoda, izumljena 2005. godine [23].

Za pronalazak lica na slici potrebno je izraditi **crno-bijele** fotografije jer za pronalaženje lica nisu potrebni podaci u boji [23].

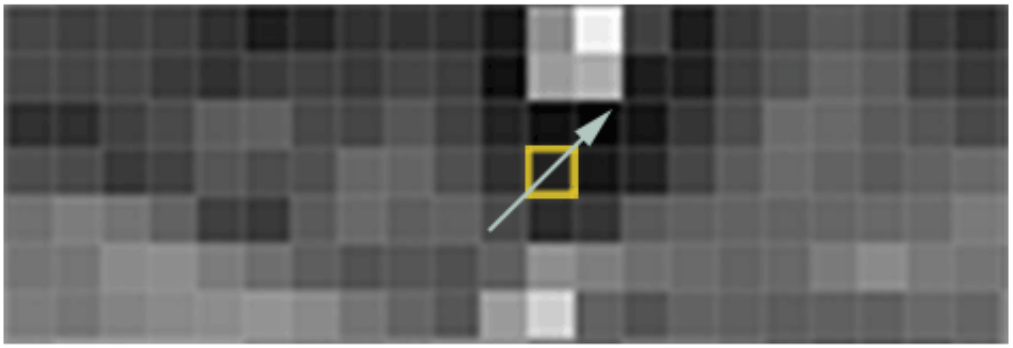
Slika 11: Will Farrell, test slika za detekciju lica (crno-bijela)
[23]

Zatim će algoritam tražiti svaki pojedini piksel na slici, jedan po jedan. Za svaki pojedini piksel pregledat će se pikseli koji ga izravno okružuju [23].



Slika 12: Detekcija susjednih piksela
[23]

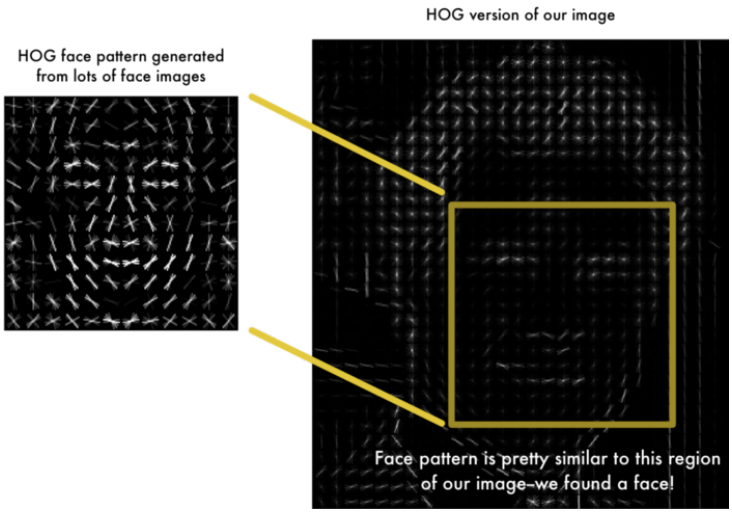
Cilj je shvatiti koliko je trenutni piksel taman u usporedbi s pikselima koji ga izravno okružuju. Zatim će se nacrtati strelica koja pokazuje u kojemu smjeru fotografija postaje tamnija [23].



Slika 13: Detekcija smjera tamnijeg susjednog piksela
[23]

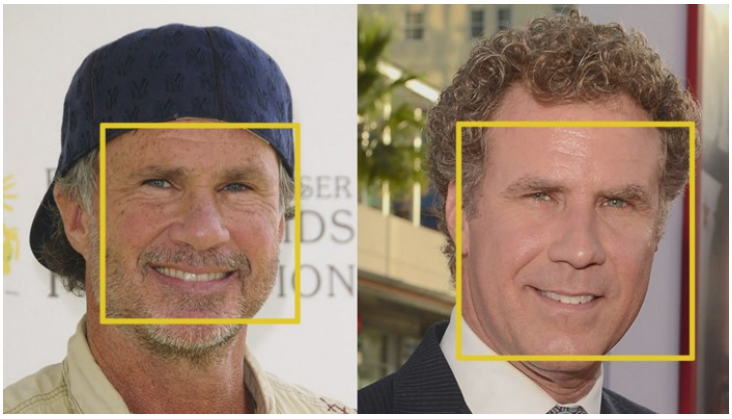
Ako se taj postupak ponovi za svaki pojedini piksel na slici, na kraju će svaki piksel biti zamijenjen strelicom. Te se strelice nazivaju gradijentima i prikazuju protok od svijetlog do tamnog preko cijele fotografije [23].

Fotografija se dijeli na male kvadrate od po 16x16 piksela. U svakom će se kvadratu izbrojati koliko gradijenata pokazuje u svakom od glavnih smjerova (koliko pokazuje gore, usmjerava gore-desno, pokazuje desno, i tako dalje). Tada će kvadrat na slici zamijeniti smjerom strelice koji je bio najjači. Krajnji je rezultat da se izvorna fotografija pretvori u vrlo jednostavan prikaz koji na jednostavan način bilježi osnovnu strukturu lica. Da bi se pronašla lica na ovoj HOG slici, algoritam će pronaći dio fotografije koji izgleda najsličnije poznatom HOG uzorku, koji je napravljen iz gomile drugih lica za trening [23]:



Slika 14: Detekcija lica u HOG verziji test slike
[23]

Detektirana se regija tada prenosi na originalnu fotografiju:

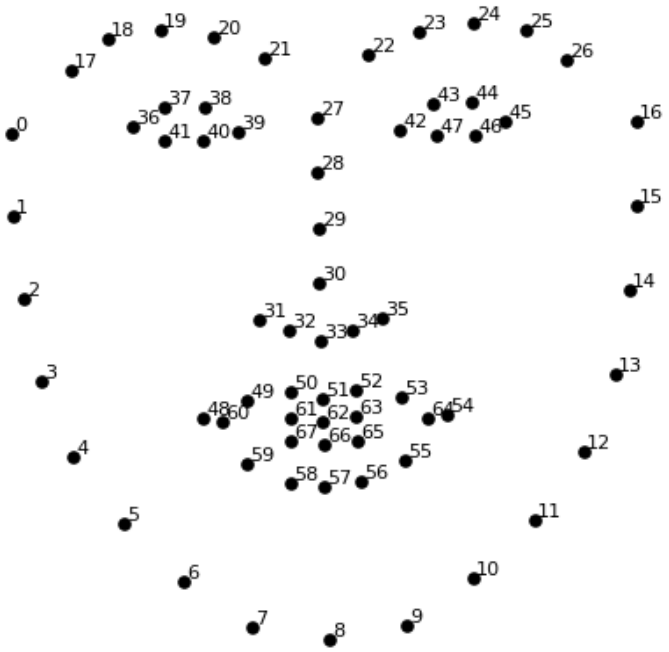


Slika 15: Detekcija lica u originalnoj verziji test slike
[23]

4.5.2. Poziranje i projiciranje lica

Zatim se algoritam nalazi pred komplikacijom lica okrenutih u različitim smjerovima koja izgledaju **potpuno drugačije** računalu.

Osnovna je ideja postaviti **68 specifičnih točaka (nazvanih orijentirima)** koje postoje na svakom licu, to su vrh brade, vanjski rub svakog oka, unutarnji rub svake obrve... Za to je potreban algoritam koji se naziva procjena orijentira lica, koji su izmislili 2014. godine Vahid Kazemi i Josephine Sullivan [24].



Slika 16: Generički prikaz orijentiri
[23]

4.5.3. Kodiranje lica

Najjednostavniji pristup prepoznavanju lica izravna je usporedba nepoznatog lica iz koraka 2 sa svim fotografijama ljudi koji su već označeni. Kada se nađe prethodno označeno lice koje izgleda vrlo slično nepoznatom licu, računalo misli da to mora biti ista osoba. Zapravo postoji ozbiljan problem s takvim pristupom. Internetska stranica poput Facebooka s milijardama korisnika i bilijunom fotografija nikako ne može proći kroz svako prethodno označeno lice da bi ga usporedila sa svakom novoučitanom fotografijom. To bi potrajalo predugo. Takvi programi moraju biti sposobni prepoznati lica u milisekundama, a ne satima. Potreban je način da se iz svakog lica izvuče nekoliko osnovnih mjera. Tada bi se moglo na isti način izmjeriti nepoznato lice i s najbližim mjerenjima pronaći poznato lice [23].

Čini se da mjerenja koja se ljudima čine očiglednima (poput boje očiju) nemaju smisla računalu koje gleda pojedinačne piksele na slici. Istraživači su otkrili da je najtočniji pristup omogućiti računalu da sam izračuna mjerenja. Dubinsko učenje čini bolji posao od ljudi u otkrivanju dijelova lica koji su važni za mjerenje [23].

Rješenje je osposobiti **duboku konvolucijsku neuronsku mrežu**, potrebno je trenirati da se generira 128 mjerenja za svako lice [23].

Proces treninga djeluje tako da se istodobno gledaju po tri fotografije lica:

1. Učitavanje fotografije lica poznate osobe
2. Učitavanje druge fotografije iste poznate osobe
3. Učitavanje fotografije potpuno druge osobe [23].

Algoritam pregledava mjerenja koja se generiraju za svaku od tri fotografije. Zatim se dotjeruje neuronska mreža tako što se osigurava da su mjere koje generira za fotografije 1 i 2 malo bliže, istovremeno osiguravajući da su mjere za fotografije 2 i 3 malo udaljenije [23].

Nakon ponavljanja ovog koraka milijunima puta i za milijune fotografija tisuća različitih ljudi, neuronska mreža uči pouzdano generirati 128 mjerenja za svaku osobu. Bilo kojih deset različitih fotografija iste osobe trebale bi dati približno iste mjere. 128 mjerenja iz trojnog učenja svakog lica nazivaju se ugrađivanjem (engl. *embedding*) [23].

Takav proces obuke konvolucijske neuronske mreže za izlaz ugrađenih lica zahtijeva mnogo podataka i računala. Čak i sa skupom Nvidia Tesla video karticom, potrebno je oko 24 sata kontinuiranog treninga da bi se dobila dobra točnost [23].

Ali nakon što je mreža obučena, može generirati mjerenja za bilo koje lice, čak i za ona koja nikad prije nije vidjela! Dakle, ovaj korak treba učiniti samo jedanput. Na internetu postoji nekoliko takvih, obučenih mreža koje se mogu izravno koristiti [23].

4.5.4. Pronalaženje imena osobe iz *encodea*

Posljednji je korak zapravo najlakši korak u cijelom procesu. Samo je potrebno pronaći u bazi podataka poznatu osobu koja ima najbliža mjerenja testnoj fotografiji nepoznate osobe[23].

To se može učiniti bilo kojim osnovnim algoritmom klasifikacije strojnog učenja. Sve što treba jest osposobiti klasifikator koji može mjeriti na novoj testnoj slici i utvrditi koja poznata osoba najbliže odgovara početnoj fotografiji. Pokretanje ovog klasifikatora traje tek nekoliko milisekundi. Rezultat klasifikatora ime je osobe [23].

5. Eksperiment *Pametna brava*

Eksperimentom je opisan i razvijen kod za korištenje umjetne inteligencije na IoT uređaju, točnije prepoznavanje lica, te testirana i osnažena sigurnost na strani IoT-a i na strani algoritma za prepoznavanja lica.

Kao što je navedeno u uvodu, za prepoznavanje lica prvo je potreban izvor, a za potrebe ovog rada korišten je ***video stream*** s IP interfonske video kamere Hikvision KD8 Series Pro Modular Door Station, koja je u ujedno i **IoT uređaj**.

Za izvršavanje prepoznavanja lica korišten je **Python** jer je lagan za naučiti, čak početnik može razumjeti kod na površnoj razini jer je sintaksa lako razumljiva, a Python omogućuje i korištenje mnogih biblioteka koje olakšavaju kodiranje. Jezik je zapravo korišten najviše za znanost podataka (engl. *data science*) i strojno učenje, što je i potrebno za ovaj eksperimentalni rad. Za prepoznavanje lice potrebno je izabrati jedan *machine learning* algoritam, dati mu podatke i dobiti rezultate.

Kod prepoznavanje važno je izabrati i isprobati pogodnu **biblioteku** za prepoznavanje lica (engl. *face recognition*) koji odgovara potrebama projekta. Python nudi razne biblioteke i svaki ima posebnu značajke i namjenu, a to su:

1. **Adam Geitgey-eva `face_recognition` biblioteka, koja je ujedno i API za naredbeni redak** (`https://github.com/ageitgey/face_recognition`);

   Ova biblioteka odabrana je za ovaj projekt jer je jako jednostavna i jer je općenito točnija od ostalih. **Idealna je za manju bazu lica**, nije potrebna ogromna računalna snaga, te koristi `dlib` biblioteku kao temelj za prepoznavanje lica, koju održava Davis King, sadrži implementaciju *dubokog metričkog učenja* koje se koristi za izradu ugrađenih lica (ugrađeno je lice vektor koji predstavlja značajke izvučene s lica) koja se koriste za postupak prepoznavanja [1].

   Ima mogućnost prebacivanja između dva glavna algoritma, histograma orijentiranih gradijenata (engl. *histogram of oriented gradients - HOG*) i detektora prepoznavanja lica zasnovanog na konvolucijskim neuronskim mrežama (engl. *convolutional neural network - CNN*). Povijesno gledano, HOG je bila najčešće korištena metoda mnogih algoritama za otkrivanje, nakon što su Dalal i Triggs 2005. godine objavili svoj rad o otkrivanju pješaka. Pristupi neuronskim mrežama su raširenija posljednjih godina zbog značajnih poboljšanja performansi, zahvaljujući jednostavnosti treninga na grafičkim karticama [25].

   Model ima **točnost 99, 38%** na mjerilu (engl. *benchmark*) Labeled Faces in the Wild (`http://vis-www.cs.umass.edu/lfw/`) [26].

   Prepoznavanje lica temeljno na ovoj biblioteci je **vrlo je precizno i može se izvršiti u stvarnom vremenu**.

2. **Ramiz Rajaeva biblioteka**

   (`https://github.com/informramiz/opencv-face-recognition-python`);

Koristi **stare** algoritme iz OpenCV-a (Eigenfaces, Fisherfaces, LBPH). Povijesno gledano, eigenfaces je bio prvi pristup koji je dobro funkcionirao za praktično prepoznavanje lica u ranim 90-ima, a Fisherfaces bili su napredak povrh eigenfacesa [25].

3. **Multi-Task Cascaded Convolutional Neural Networks - MTCNN;**

   Daljnji napreci u posljednje dvije godine koriste specifičnu vrstu neuronske mreže, nazvanu višesatnom kaskadnom konvolucijskom neuronskom mrežom (engl. *Multi-Task Cascaded Convolutional Neural Networks - MTCNN*). Ona je rješila neke slabosti CNN-a poput one koju koristi dlib, koja propušta neka lica [27].

   Sve u svemu, samo zato što je algoritam najnoviji, **to ne znači da je najbolji**.

4. **Google FaceNet;**

   Googleov FaceNet već se nekoliko godina smatra **vrhunskim** u prepoznavanju lica u usporedbi s već spomenutim LFW *benchmarkom*. FaceNet je **najpopularniji** i ima mnogo implementacija otvorenog koda. Jedna je takva implementacija OpenFace, sjajna i zaista jednostavna za upotrebu (iako ponekad teška za instalaciju).

   FaceNet i njegove implementacije, bez obzira na odlične rezultate prepoznavanja, nisu dobar odabir za ovaj projekt jer zahtijevaju ogromnu računalnu moć.

Često se mora napraviti **kompromis** između računalnih zahtjeva, vremena za obradu, lažnih pozitiva i lažnih negativa. **Ovisno o zahtjevima** proizvoda ovisit će i koji je najbolji. Iz prakse se može utvrditi da dlibov CNN dobiva više lažnih negativa (lice neće biti prepoznato, iako je lice u bazi), MTCNN dobiva više lažno pozitivnih rezultata (prepoznat će lice krivo što za potrebu ovog projekta je jako loše jer može odobriti ulaz stranoj osobi u prostor), dok će HOG biti najbrži, ali mu je niža točnost u usporedbi s pristupima neuronskih mreža.

5.1. Prepoznavanje lica s OpenCV-om, Pythonom i dubokim učenjem

Da bi se izvršilo prepoznavanje lica **Python**om i **OpenCV**-om moraju se instalirati dvije dodatne biblioteke:

- dlib (`http://dlib.net/`)
- face_recognition (`https://github.com/ageitgey/face_recognition`).

Biblioteka dlib, koju održava Davis King, sadrži implementaciju *dubokog metričkog učenja* koje se koristi za izradu ugrađenih obraza, koje se koriste za stvarni postupak prepoznavanja. face_recognition biblioteka, koju je stvorio Adam Geitgey, omotava dlibovu funkciju prepoznavanja lica, olakšavajući rad s njom [2].

OpenCV, dlib i face_recognition moduli mogu su instalirati pomoću Python *installer* paketa (engl. *Python Package Installer - PIP*).

Potrebno je napuniti **skup podataka** za prepoznavanje lica. Skup fotografija izraditi će 128-d *embedding* za svako lice u skupu podataka, te ih upotrijebite za prepoznavanje lica na video *streamu*.

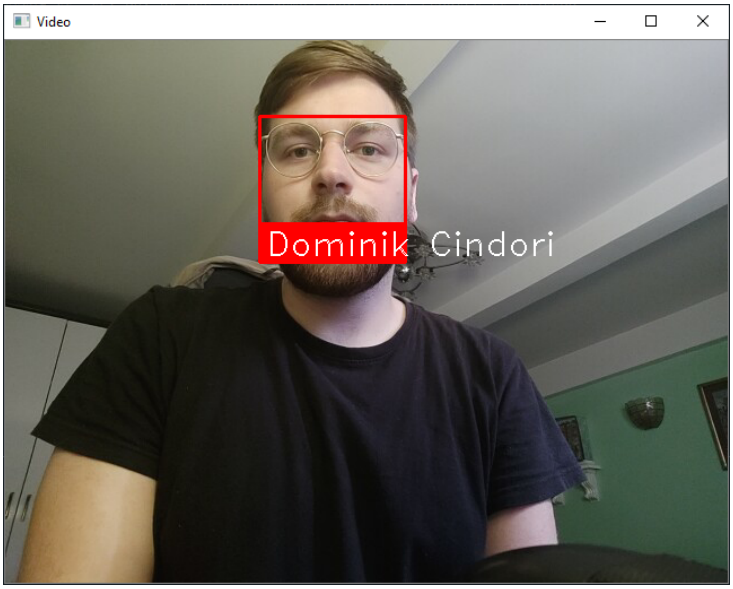
Prije no što se prepoznaju lica na fotografijama i videozapisima, prvo se moraju kvantificirati lica u setu treninga. Za projekt nije bilo potrebno trenirati mrežu, nego je mreža već obučena za stvaranje 128-d *encoded* podataka.

U Python kod potrebno je uvesti pakete iz biblioteka. A to su:

- face_recognition (biblioteka za prepoznavanje lica)
- cv2 (OpenCV)
- numpy
- dodatne biblioteke za *multi-threading*, opcionalno za bolje performanse.

Također je potrebno inicijalizirati dva popisa koja će sadržavati *encoded* lica i odgovarajuća imena za svaku osobu u skupu podataka.

U projektu je korišten **video *stream*** s video interfona, zato je potrebno napraviti proces koji će snimati svaku N sličicu videa (engl. *frame*) za potrebe prepoznavanje nepoznate osobe. Kada je snimljen *frame*, potrebno ga je procesirati kroz više manjih procesa. *Frame* se učita, zatim se pretvara iz BGR boje (koju koristi OpenCV) u RGB model boje (koju koristi face_recognition). U tom *frameu* sada se pronalaze sva lica i *encodingsi* lica. Nakon toga ulazi se u iteraciju kroz sva prepoznata lica pojedinačno te se provjerava odgovara li poznatim licima. Python crta regiju prepoznatog lica i ispisuje ime osobe koja je prepoznana i tako u krug.



Slika 17: Uspješno izvršeno prepoznavanje lica

Nakon uspješnog prepoznavanja lica potrebno je poraditi na **sigurnosti**, jer program može prepoznati osobu i s ekrana mobitela ili isprintano lice na papir. To je važno eliminirati

tako da neautorizirana osoba ne iskoristi lice iz baze jer će joj u tom slučaju biti odobren ulaz u prostor. Potrebno je poraditi na sigurnosnom otvrdnjavanju algoritama za prepoznavanje lica.

5.2. Sigurnosno otvrdnjavanje algoritama za prepoznavanje lica

Kao i većina novih tehnologija, i ova uvodi **nove mogućnosti napada** na sigurnost. Jedan je od najpopularnijih načina za zavaravanje mehanizma za prepoznavanje lica *face spoof*. **Spoofing** napad predstavlja pokušaj stjecanja tuđih privilegija ili prava pristupa korištenjem fotografije, videozapisa ili druge zamjene za lice ovlaštene osobe [28].

Neki primjeri napada:

- **Napad ispisom;**

  Napadač koristi nečiju fotografiju, ona se ispisuje ili prikazuje na digitalnom uređaju

- **Napad ponovnim reproduciranjem video zapisa;**

  Sofisticiraniji način zavaravanja sustava, obično zahtijeva petlju videozapisa lica žrtve. Takav pristup osigurava da ponašanje i pokreti lica izgledaju *prirodnije* u usporedbi s držanjem nečije fotografije

- **Napad isprintanom 3D maskom;**

  Tijekom ove vrste napada maska se koristi kao odabrani alat za lažno predstavljanje. To je još sofisticiraniji napad od reprodukcije videozapisa s lica. Uz prirodne pokrete lica, omogućuje načine zavaravanja nekih dodatnih slojeva zaštite, poput senzora dubine [28].

Potrebno je zapitati se kako prepoznati pravo lice od lažnog lica. Postoje mnogi načini kojima se može prepoznati je li u pitanju prava osoba ili pokušaj *spoofing* napada. Najčešći načini prevencije *spoofing* napada su:

- **Otkrivanje živosti lica**(engl. *Face Liveness Detection*);

  Mehanizam zasnovan na analizi koliko je testno lice *živo*. To se obično radi provjerom pokreta očiju, poput treptanja i pokreta lica [28].

- **Otkrivanje treptaja oka;**

  Otkrivanje treptaja oka jedan je test za otkrivanje živosti koji je veoma precizan. Prirodno treptanje jednostavan je način da se utvrdi je li lice u tomu trenutku prisutno uživo ili nije. Prosječni čovjek trepće 15 do 30 puta u minuti. Oči ostaju zatvorene oko 250 milisekundi tijekom treptanja [29].

- **Interakcija korisnika;**

  Aktivan način prevencije, zahtjeva korisnika da izvrši radnju (okretanje glave lijevo/desno, smiješak, treptanje očiju) kako bi stroj mogao otkriti je li radnja izvršena na prirodan način koji nalikuje ljudskoj interakciji [28].

- **3D kamera;**

  3D kamere najpouzdanije su sredstvo protiv lažnog predstavljanja. Precizne informacije o dubini piksela mogu pružiti visoku točnost protiv napada prezentacije jer mogu razlikovati lice od ravnog oblika. 3D napadi mogu uzrokovati poteškoće, ali kamere su i dalje jedna od najpouzdanijih dostupnih tehnika protiv lažnog predstavljanja i unatoč dostupnosti kamera, nemaju ih svi korisnici na svojim računalima [29].

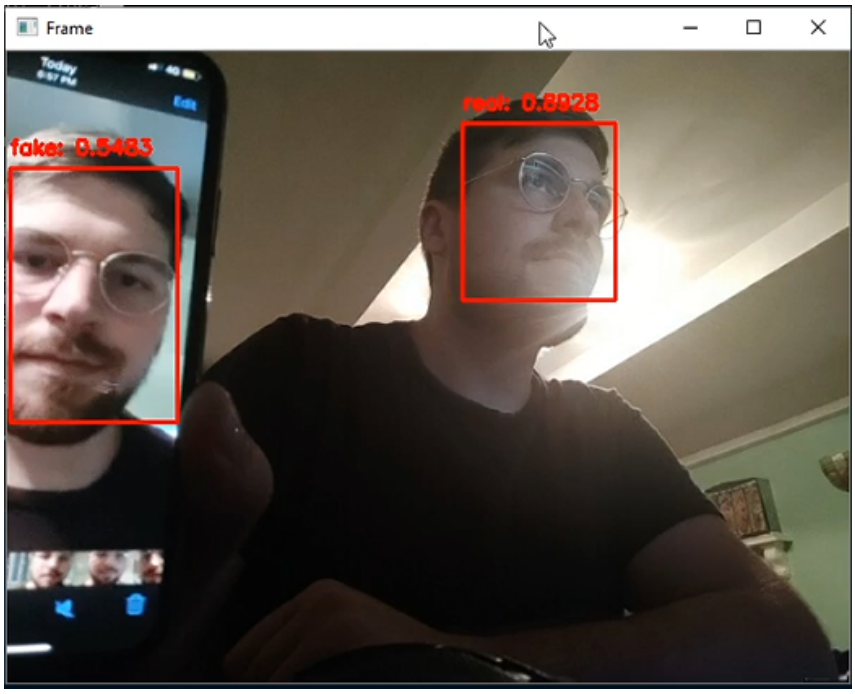
- **Aktivan *flash*;**

  Aktivna bljeskalica zanimljiva je tehnika za koju se smatra da obećava. Ovo rješenje omogućilo je otkrivanje lažiranja pomoću refleksija svjetlosti na licu. Ideja uključuje korištenje promjenjivog svjetlosnog okruženja koje pruža dodatno svjetlo, koje dolazi sa zaslona uređaja. Bijelo svjetlo stvara odgovarajući odraz na licu [29].

Za potrebe ovog projekta koristiti će se ***face liveness detection* i otkrivanje treptaja oka**. Postoji niz pristupa otkrivanju živosti, jedan je od njih analiza teksture, uključujući izračunavanje lokalnih binarnih uzoraka (LBP) preko područja lica i korištenje SVM-a za klasifikaciju lica kao stvarnih ili lažnih. Zatim analiza frekvencije, poput ispitivanja Fourierove domene lica, pa analiza promjenljivog fokusiranja, poput ispitivanja varijacije vrijednosti piksela između dva uzastopna okvira. Postoje i euristički algoritmi koji uključuju kretanje oka, pokret usana i otkrivanje treptaja. Ovaj skup algoritama pokušava pratiti kretanje očiju i treptaje kako bi osigurao da korisnik ne drži fotografiju druge osobe (jer fotografija neće treptati ili pomicati usnama). Algoritmi optičkog protoka ispitivanje je razlika i svojstava optičkog protoka generiranog od 3D objekata i 2D ravnina. 3D oblik lica, sličan onome koji se koristi na Appleovom iPhone sustavu prepoznavanja lica, omogućuje sustavu prepoznavanja lica da razlikuje stvarna lica od ispisa fotografija druge osobe. Kombinacije gore navedenog, omogućavaju inženjeru sustava za prepoznavanje lica da odabere modele za otkrivanje živosti koji odgovaraju njihovoj određenoj primjeni [30].

Za potrebe ovog projekta otkrivanje živosti tretirat će se kao **binarni problem klasifikacije**.

Kako bi se stvorio otkrivanje živosti, treba trenirati duboku neuronsku mrežu sposobnu za razlikovanje stvarna od lažnih lica. Potrebno je izgradite skup podataka, zatim primijeniti CNN sposoban za izvođenje detektora živosti, obučiti tu mrežu detektora živosti te na kraju stvoriti Python + OpenCV skriptu koja je sposobna uzeti obučeni model detektora živosti i primijeniti ga na video u stvarnom vremenu.



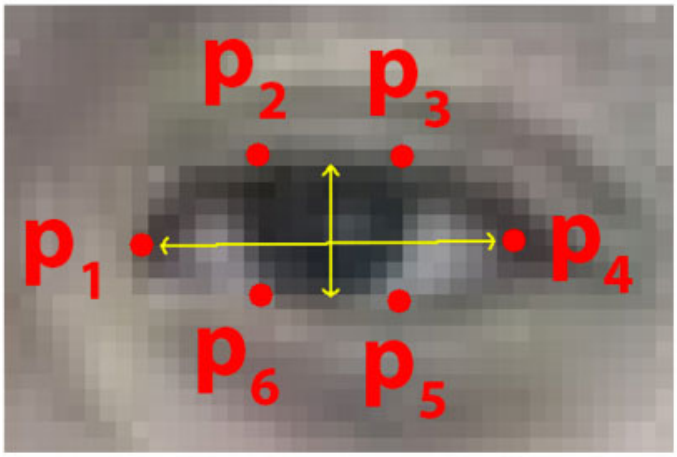
Slika 18: Otkrivanje živosti pomoću Pythona i OpenCV

Za otkrivanje treptaja oka potrebni su **orijentiri** spomenuti u poziranju i projiciranju lica. Potrebno je nadograditi te orijentire i razviti aplikaciju za računalni vid, koja je sposobna detektirati i brojati treptaje u video *streamovima* pomoću orijentira na licu i OpenCV-a. Da bi se razvio takav detektor treptaja, treba izračunati metriku koja se naziva **omjer oka** (engl. *Eye Aspect Ratio - EAR*), koju su Soukupová i Čech uveli u svom radu 2016. godine. Kod detekcije treptaja pomoću orijentira lica u stvarnom vremenu [31].

Orijentiri su bolji za prepoznavanje treptaja; za razliku od tradicionalnih metoda obrade fotografija za računanje treptaja, koje su kombinacije lokalizacija oka, praga za pronalaženje bjeloočnica i utvrđivanja nestaje li "bijelo" područje očiju na određeno vrijeme (što ukazuje na treptaj) [32].

EAR je umjesto toga mnogo elegantnije rješenje koje uključuje vrlo jednostavan izračun koji se temelji na omjeru udaljenosti između orijentira lica očiju. Ova metoda za otkrivanje treptaja oka brza je, učinkovita i jednostavna za primjenu [32].

Svako oko predstavljeno je sa 6 (x, y) -koordinata, počevši od lijevog kuta oka (kao da gledate osobu), a zatim, radeći u ostatku regije, u smjeru kazaljke na satu:



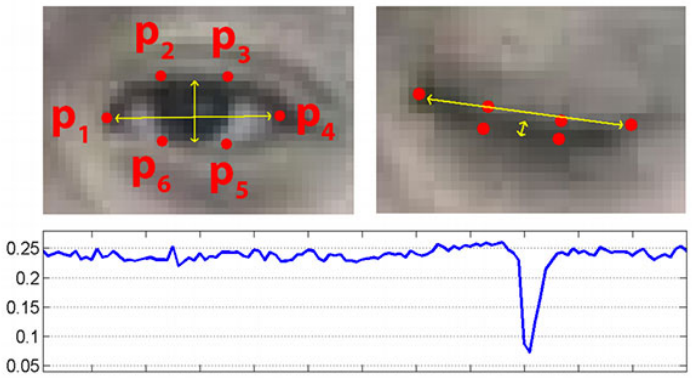
Slika 19: 6 orijentira lica povezanih s okom
[32]

Postoji **veza između širine i visine ovih koordinata**. Na temelju rada Soukupová i Čecha u radu iz 2016. godine, Detekcijom treptaja očiju u stvarnosti pomoću orijentira na licu u stvarnom vremenu može se izvući jednadžba koja odražava sljedeću vezu [31]:

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

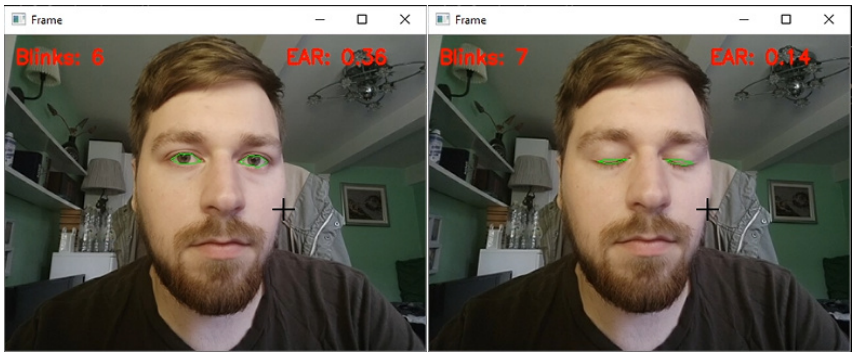
Slika 20: Jednadžba omjera oka
[31]

Koristeći ovu jednostavnu jednadžbu, mogu se izbjeći tehnike obrade fotografija i jednostavno se pouzdati u omjer udaljenosti orijentira oka, kako bi se utvrdilo trepće li osoba. Da bi to bilo jasnije, treba pogledati slikovni prikaz Soukupová i Čecha [31]:



Slika 21: Jednadžba omjera oka
[31]

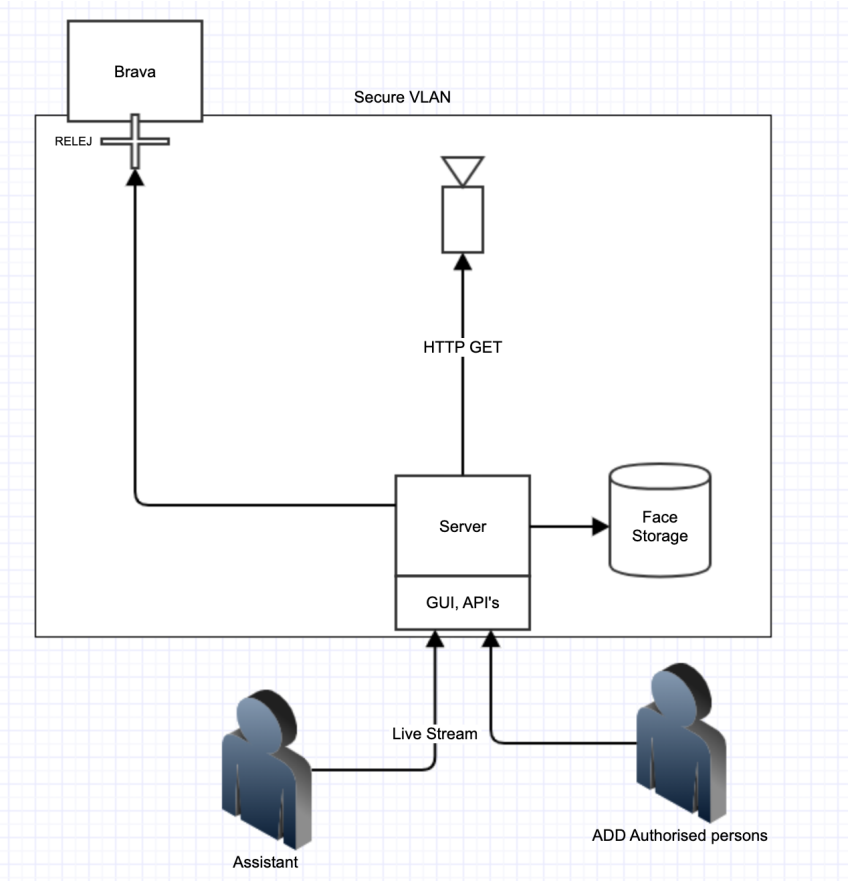
Nakon što se dobiju orijentiri na licu za oba oka, izračunava se omjer oka za svako oko, što daje jedinstvenu vrijednost, povezujući udaljenosti između okomitih orijentira oka i udaljenostima između vodoravnih točaka orijentira. Jednom kada se dobije omjer oka, može ga se premašiti kako bi se utvrdilo trepće li osoba - omjer oka ostat će približno konstantan kada su oči otvorene, a zatim će se brzo približiti nuli tijekom treptanja, a zatim će se ponovo povećavati kako se oko otvara. Kako bi poboljšali otkrivanje treptanja, Soukupová i Čech preporučuju izgradnju 13-prigušenog vektora obilježja omjera oka (N-ti okvir, N - 6 okvira i N + 6 okvira), nakon čega slijedi unos vektora značajke u Linearni SVM za klasifikaciju [32].



Slika 22: Prepoznavanje treptaja pomoću Pythona i OpenCV

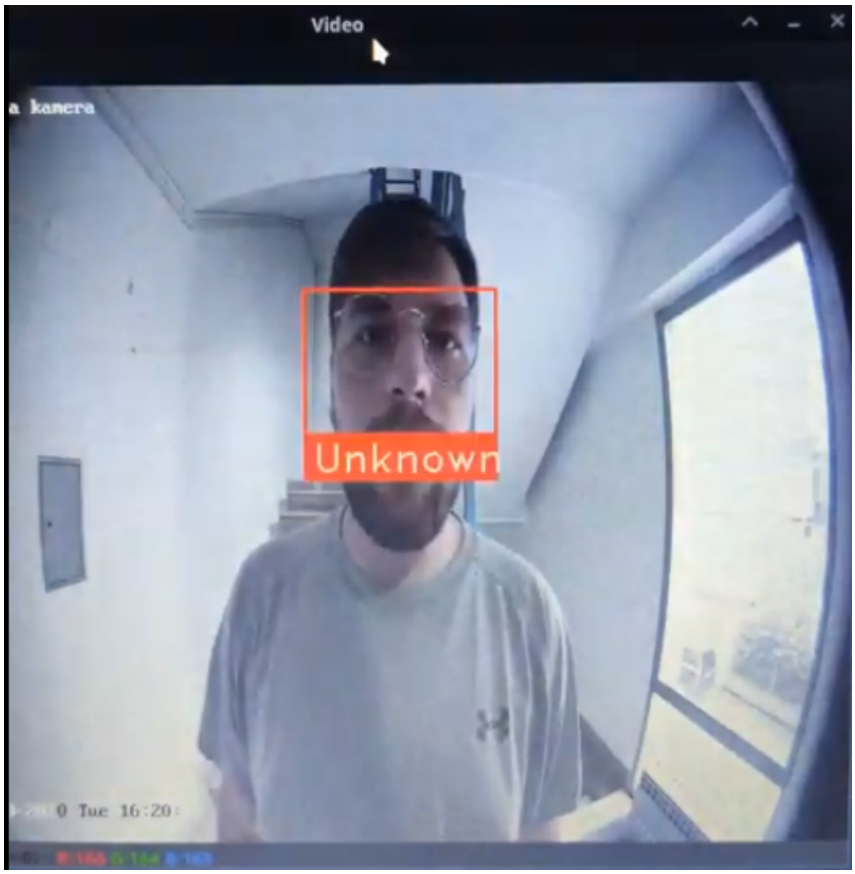
5.3. Integracija IoT-a i prepoznavanje lica

U projektu je obrađen svaki aspekt, sada je potrebno sve zajedno **integrirati u kohe-rentan/povezan sustav**. Sama ideja izgleda ovako:



Slika 23: Shematski prikaz sustava pametne brave

Umjesto korištenja običnog video *streama* s internetske kamere laptopa, koristit će se Hikvision IP kamera s video interfona. Za dohvaćanjem video *streama* s video kamere potrebno je uzeti prikladan Hikvision **RTSP** URL format. S obzirom na to da navedena kamera ima takozvani ***fish-eye efekt***, orijentiri lica malo su drugačiji nego kada je bez efekta jer *-eye efekt* napravi distorziju fotografije i zbog toga je potrebno napraviti novu bazu lica. Za izgradnju nove baze lica potrebno je svakog korisnika fotografirati Hikvisionovom kamerom, umjesto običnom, kako bi se orijentiri lica s video *streama* i baze lica bolje podudarali.



Slika 24: Neuspješno prepoznavanje zbog *fish-eye efekta*

Nakon testiranja prepoznavanja lica, moguće je podesiti **toleranciju** da bi se dobilo više ***false positive/negatives***, što zavisi o potrebi. S obzirom na to da se *pametna brava* koristi za ulaz u radni prostor (ured), tolerancija je snižena kako bi se izbjegao *false negative* rezultati jer u slučaju *false negative* rezultata neautorizirana bi osoba mogla ući jer bi ju sustav mogao prepoznati kao zaposlenika zgrade (ako, na primjer, približno sliči zaposleniku koji se nalazi u bazi lica).

Ako je lice uspješno prepoznano, osoba prođe dodatne sigurnosne mjere (*face liveness* i otkrivanje treptaja oka), sustav otključava vrata te omogućuje osobi da ude u radni prostor zgrade.

6. Zaključak

Predmet je rada sigurnosno otvrdnjavanje algoritama za prepoznavanje lica, a izabrane su metode koje su testirane i iskorištene otkrivanje živosti i otkrivanje treptaja oka. Detekcija je živosti dokazana kao odlična metoda sigurnosnog otvrdnjavanja, kao i otkrivanje treptaja oka, koji dodatno osigurava u kombinaciji s prethodnom metodom.

Testovi su bili uspješni nakon integracije pojedinačnog sustava, a puno sigurniji kao kombinacija oba. Bez ovih metoda moguće je takozvano *spoofati* lice i prevariti prepoznavanje lica s ispisanom fotografijom na papiru ili digitalnom fotografijom na nekom mobilnom uređaju, što predstavlja mogućnost zloupotrijebe. Stroga je preporuka koristiti neku metodu sigurnosnog otvrdnjavanja algoritma kod korištenja IoT uređaja i umjetne inteligencije. Tehnologija je još nova i, kao kod nastanka svake nove tehnologije, otvaraju se nove mogućnosti napada. Ostale metode, koje uključuju 3D kameru i aktivni *flash*, nije bilo moguće dokazati zbog hardverskih ograničenosti kamere. Python je bio glavni izbor za programski jezik i bazu jer je lako razumljiv, ima bogat izbor biblioteka i najčešći je izbor za umjetnu inteligenciju. Od potrebnih biblioteka koje su dodatno popunile Python bile su: dlib, OpenCV i face_recognition, koju je stvorio Adam Geitgey, omotava dlib-ovu funkciju prepoznavanja lica, olakšavajući rad s njom. Sami sustav ne zahtijeva ogromnu računalnu moć te je moguće pokrenuti ga kod čak i na Raspberry Piu. Za brži rad moguće je adaptirati programski kod da koristi *multi-threading* procesor i CUDA jezgru NVIDIA grafičkih kartica. Također se kao dobra praksa pokazalo treniranje modela umjetne inteligencije sa što više lica, kako bi umjetna inteligencija imala uspješnije rezultate prepoznavanja lica. Kod prepoznavanja lica moguće je podesiti toleranciju prepoznavanja, koja pri povećavanju vrijednosti daje više *false-positive* rezultata, a kod smanjivanja vrijednosti više *false-negative* rezultata. To znači da, kada bi tolerancija bila jako visoka, moguće bi bilo da se lice prepozna, iako nije u bazi, dok će se kod male tolerancije dogoditi da će lice koje je u bazi biti teže prepoznano, ili uopće neće biti. Toleranciju je potrebno regulirati i testirati po potrebi primjene prepoznavanja lica, a u slučaju rada trebala je biti što niža jer ukoliko osoba koja nije u bazi lica bude prepoznana, svejedno će joj biti autoriziran ulaz u ured.

Ovaj sigurnosno otvrdnjen sustav može se odlično iskoristiti s IoT uređajima koji olakšavaju svakodnevicu, kao na primjer s pametnom bravom, koja je ujedno bila i eksperimentalni dio rada.

Popis literature

[1] *dlib C++ Library - Introduction*, en. adresa: http://dlib.net/intro.html (pogledano 20. 8. 2020).

[2] *An interview with Adam Geitgey, creator of the face_recognition Python library*, en, Section: Interviews, srpanj 2018. adresa: https://www.pyimagesearch.com/2018/07/11/an-interview-with-adam-geitgey-creator-of-the-face_recognition-python-library/ (pogledano 20. 8. 2020).

[3] *DS-KD8003-IME1/Surface | Pro Series | Hikvision*, en. adresa: https://www.hikvision.com/en/products/Video-Intercom-Products/IP-Series/Pro-Series/DS-KD8003-IME1-Surface/ (pogledano 7. 7. 2020).

[4] *What is the IoT? Everything you need to know about the Internet of Things right now | ZDNet*, en. adresa: https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-IoT-right-now/ (pogledano 7. 7. 2020).

[5] *What is IoT (Internet of Things) and How Does it Work?*, en. adresa: https://internetofthingsagetechtarget.com/definition/Internet-of-Things-IoT (pogledano 7. 7. 2020).

[6] *This Japanese hotel room TV has a channel for monitoring the status of the laundry machines.* en. adresa: https://www.reddit.com/r/IoT/comments/68snk5/this_japanese_hotel_room_tv_has_a_channel_for/ (pogledano 7. 7. 2020).

[7] *Ethical Hacking: Hacking the Internet of Things (IoT) | Pluralsight*, en. adresa: https://app.pluralsight.com/library/courses/ethical-hacking-hacking-internet-of-things/table-of-contents (pogledano 21. 7. 2020).

[8] *Internet of Things (IoT) History*, en. adresa: https://www.postscapes.com/IoT-history (pogledano 7. 7. 2020).

[9] C. Barker, *If you want to succeed you must fail first, says the man who dreamt up the Internet of Things*, en. adresa: https://www.zdnet.com/article/if-you-want-to-succeed-you-must-fail-first-says-the-man-who-dreamt-up-the-internet-of-things/ (pogledano 7. 7. 2020).

[10] *Kevin Ashton (@Kevin_Ashton) / Twitter*, en. adresa: https://twitter.com/Kevin_Ashton (pogledano 7. 7. 2020).

[11] *The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025, According to a New IDC Forecast*, en. adresa: https://www.idc.com/getdoc.jsp?containerId=prUS45213219 (pogledano 8. 7. 2020).

[12] *Gartner Says 5.8 Billion Enterprise and Automotive IoT Endpoints Will Be in Use in 2020*, en. adresa: https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io (pogledano 8. 7. 2020).

[13] *How the Internet of Things will reshape future production systems | McKinsey*, en. adresa: https://www.mckinsey.com/business-functions/operations/our-insights/how-the-internet-of-things-will-reshape-future-production-systems (pogledano 8. 7. 2020).

[14] *Amazon.com: Echo (2nd Generation) - Smart speaker with Alexa and Dolby processing - Charcoal Fabric: Amazon Devices*, en. adresa: https://www.amazon.com/all-new-amazon-echo-speaker-with-wifi-alexa-dark-charcoal/dp/B06XCM9LJ4 (pogledano 8. 7. 2020).

[15] *The OWASP IoT Top 10 List of Vulnerabilities*, en, travanj 2020. adresa: https://sectigostore.com/blog/owasp-IoT-top-10-IoT-vulnerabilities/ (pogledano 21. 7. 2020).

[16] *Telnet*, en, Page Version ID: 972680982, srpanj 2020. adresa: https://en.wikipedia.org/w/index.php?title=Telnet&oldid=972680982 (pogledano 21. 7. 2020).

[17] P. Geenens, *BusyBox Botnet Mirai – the warning we've all been waiting for?*, en, Section: Security, listopad 2016. adresa: https://blog.radware.com/security/2016/10/busybox-botnet-mirai/ (pogledano 21. 7. 2020).

[18] *BusyBox*, en. adresa: https://busybox.net/about.html (pogledano 21. 7. 2020).

[19] *What is Artificial Intelligence? How Does AI Work? | Built In*, en. adresa: https://builtin.com/artificial-intelligence (pogledano 14. 8. 2020).

[20] *Artificial Intelligence – What it is and why it matters*, en. adresa: https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html (pogledano 14. 8. 2020).

[21] K. Lorena, „UMJETNA INTELIGENCIJA DANAS", hr, str. 81,

[22] *Artificial Intelligence Threats and Security Issues*, en. adresa: https://www.identitymanagementi
org/artificial-intelligence-threats-and-security-issues/ (pogledano 14. 8. 2020).

[23] A. Geitgey, *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning*, en, studeni 2018. adresa: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78 (pogledano 27. 8. 2020).

[24] V. Kazemi i J. Sullivan, „One millisecond face alignment with an ensemble of regression trees”, en, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH: IEEE, lipanj 2014, str. 1867–1874, ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.241. adresa: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909637 (pogledano 27. 8. 2020).

[25] *Face Recognition python performance suggestion*, en. adresa: https://www.reddit.com/r/computervision/comments/96pvjk/face_recognition_python_performance_suggestion/ (pogledano 27. 8. 2020).

[26] A. Geitgey, *ageitgey/face_recognition*, en, original-date: 2017-03-03T21:52:39Z, rujan 2020. adresa: https://github.com/ageitgey/face_recognition (pogledano 27. 8. 2020).

[27] K. Zhang, Z. Zhang, Z. Li i Y. Qiao, „Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks”, en, *IEEE Signal Processing Letters*, sv. 23, br. 10, str. 1499–1503, listopad 2016, arXiv: 1604.02878, ISSN: 1070-9908, 1558-2361. DOI: 10.1109/LSP.2016.2603342. adresa: http://arxiv.org/abs/1604.02878 (pogledano 27. 8. 2020).

[28] YND, *Anti-Spoofing Mechanisms in Face Recognition Based on DNN*, en, srpanj 2019. adresa: https://medium.com/swlh/anti-spoofing-mechanisms-in-face-recognition-based-on-dnn-586011ccc416 (pogledano 27. 8. 2020).

[29] S. Maksymenko, *Anti-Spoofing Techniques For Face Recognition Solutions*, en, srpanj 2020. adresa: https://towardsdatascience.com/anti-spoofing-techniques-for-face-recognition-solutions-4257c5b1dfc9 (pogledano 27. 8. 2020).

[30] S. Chakraborty i D. Das, „An Overview of Face Liveness Detection”, en, *International Journal on Information Theory*, sv. 3, br. 2, str. 11–25, travanj 2014, ISSN: 23208465, 23197609. DOI: 10.5121/ijit.2014.3202. adresa: http://www.airccse.org/journal/ijit/papers/3214ijit02.pdf (pogledano 27. 8. 2020).

[31] T. Soukupova, „Real-Time Eye Blink Detection using Facial Landmarks”, en, str. 8,

[32] *Eye blink detection with OpenCV, Python, and dlib*, en-US, Section: dlib, travanj 2017. adresa: https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/ (pogledano 27. 8. 2020).

Popis slika

Popis tablica

1. Prilog 1

2. Prilog 2