

# Analiza i usporedba neuronskih mreža za prepoznavanje lica

---

Sedlarević, Daniel

Undergraduate thesis / Završni rad

2023

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:211:928333>

*Rights / Prava:* [Attribution-NonCommercial-NoDerivs 3.0 Unported / Imenovanje-Nekomercijalno-Bez prerada 3.0](#)

*Download date / Datum preuzimanja:* **2024-07-15**



*Repository / Repozitorij:*

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN**

**Daniel Sedlarević**

**ANALIZA I USPOREDBA NEURONSKIH  
MREŽA ZA PREPOZNAVANJE LICA**

**ZAVRŠNI RAD**

**Varaždin, 2023.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Daniel Sedlarević**

**Matični broj: 0016136444 0**

**Studij: Informacijski sustavi**

**ANALIZA I USPOREDBA NEURONSKIH MREŽA ZA  
PREPOZNAVANJE LICA**

**ZAVRŠNI RAD**

**Mentor/Mentorica:**

Izv. Prof. dr. sc. Petra Grd

**Varaždin, rujan 2023.**

Daniel Sedlarević

### **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/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi*

---

## Sažetak

U ovom radu se opisuje procese prepoznavanja lica, prednosti i nedostatke korištenja tehnologije za prepoznavanje lica. Opisuju se i neuronske mreže, od njihove strukture i načina rada, glavnih tipova neuronskih mreža i kako one rade te na kraju se radi pregled literature o neuronskim mrežama. Kod opisa strukture neuronske mreže je opisana podjela mreže na slojeve, povezanost čvorova ili neurona, te jačina konekcije između konekcija pomoću posebnog broja zvanog težina.

Opisani tipovi neuronskih mreža su Feed Forward neuronska mreža, višeslojni perceptron i konvolucijska neuronska mreža. Za svaki tip su dani primjeri korištenja i koje prednosti ima mreža u usporedbi s drugim tipovima mreža. Isto tako se opisuje njihova primjena za prepoznavanje lica.

Sljedeći dio pregledava znanstvene članke o neuronskim mrežama za prepoznavanje lica, gdje su podijeljeni prema tipu neuronske mreže. Prva dva članka su za tip konvolucijske neuronske mreže, te opisuju više različitih implementacija mreže za svrhu prepoznavanja lica. Drugi tip mreže je višeslojni perceptron s povratnim širenjem te se opisuju dvije različite implementacije mreže za prepoznavanje lica.

Zadnji dio rada opisuje vlastitu implementaciju dviju neuronskih mreža opisanih u prijašnjem dijelu te prijenos učenja za testiranje na novom skupu podataka. Nakon toga se opisuju i analiziraju rezultati dobiveni testiranjem i daje zaključak na temu.

**Ključne riječi:** prepoznavanje lica; neuronske mreže; privatnost; biometrija; strojno učenje; python; klasifikacija slika

# Sadržaj

Sadržaj .....	v
1. Uvod.....	1
2. Metode i tehnike rada.....	2
3. Prepoznavanje lica .....	3
3.1. Način rada tehnologije za prepoznavanje lica .....	3
3.2. Faktori prepoznavanja lica .....	4
3.3. Upotrebe tehnologije za prepoznavanje lica.....	5
3.4. Prednosti i nedostaci .....	7
3.4.1. Prednosti .....	7
3.4.2. Nedostaci .....	8
4. Neuronske mreže za prepoznavanje lica .....	10
4.1. Neuronske mreže .....	10
4.2. Struktura neuronskih mreža.....	10
4.2.1. Težina .....	11
4.3. Tipovi neuronskih mreža.....	12
4.3.1. Feed forward neuronska mreža .....	12
4.3.2. Višeslojni perceptron.....	12
4.3.3. Konvolucijska neuronska mreža .....	13
4.4. Primjena neuronskih mreža za prepoznavanje lica.....	14
5. Pregled literature – Neuronske mreže.....	16
5.1. Konvolucijske neuronske mreže.....	16
5.1.1. DeepFace .....	18
5.1.2. DeepID.....	19
5.1.3. FaceNet.....	20
5.1.4. ResNet50.....	21
5.1.5. SENet50.....	22
5.2. Višeslojni perceptron .....	23
6. Implementacija neuronske mreže za prepoznavanje lica .....	26
6.1. Implementacija neuronskih mreža ResNet50 i SENet50 u Python-u .....	26
6.2. Pregled rada programa .....	29
6.3. Testiranje neuronske mreže .....	31
7. Zaključak.....	34
Popis literature .....	35

Popis slika .....	38
Popis tablica .....	38

# 1. Uvod

Lice je jedno od prvih načina prepoznavanja osobe, i još uvijek se koristi na osobnim iskaznicama i putovnicama za verifikaciju identiteta. Zbog ubrzanog razvoja tehnologije i socijalnih mreža dolazi do velike rasprave o korištenju te tehnologije za prepoznavanje lica, te gdje i kako bi se koristila. Dani kada su takve tehnologije bile dostupne samo velikim institucijama i vladama su prošlost, te su sve više i više dostupnije običnim ljudima.

Dostupnost takve tehnologije povlači za sobom pitanja o sigurnosti, privatnosti i etici korištenja, koja je potrebno dobro razmatrati i pregledati radi zaštite korisnika od zloupotrebe te tehnologije. Ako se pravilno korištenje takvih tehnologija ne stavi u fokus, problemi oko krive identifikacije ili diskriminatornih sklonosti sustava mogu uzrokovati velike probleme za manjinske grupe.

Važno je i spomenuti neuronske mreže koje su u zadnje vrijeme eksplodirale u popularnosti zbog MidjourneyAI i ChatGPT, koji su omogućili osobama da kreiraju slike ili komuniciraju s umjetnom inteligencijom. Velika popularnost umjetne inteligencije je dovela do važnih pitanja o vlasničkim pravima kreiranog sadržaja.

Slično se dogodilo i s biometrijskom verifikacijom, kada su mobiteli počeli koristiti otiske prsta ili lica za otključavanje, te zbog velike popularnosti kod prvih modela su se počeli koristiti u svim novim modelima. Time se uvelike povećao utjecaj na svakodnevni život ljudi, radi čega sam odabrao ovu temu. Kroz istraživanje teme naučio sam o načinu rada takvih sustava i razlikama između njih.



## 2. Metode i tehnike rada

Metode korištene za izradu ovog rada su pregledavanje i čitanje znanstvenih i stručnih članaka o temi, analiza i uspoređivanje tih članaka. Stranice korištene su Google Scholar za pronalazak znanstvenih članaka i Google za pronalazak stručnih članaka.

Kod implementacije neuronske mreže za prepoznavanje lica sam prvo pronašao poželjnu implementaciju pretraživanjem na internetu, te sam zatim koristio Google Colab za pisanje koda u python-u. Odabrao sam već treniranu neuronsku mrežu za prepoznavanje lica, te sam je odlučio testirati tako što sam odabrao 15 osoba od skupa i pogledao da li ih sustav prepoznaje.

## 3. Prepoznavanje lica

Prepoznavanje lica je metoda na koju je bazirana tehnologija biometrije koja može prepoznati ili verificirati osobu preko slike, videa ili drugih dijelova lica. Takva tehnologija se koristi u vezi neke aplikacije, tehnologije ili sustava i radi kao skener lica. Tehnologija koristi biometrijske uzorke i podatke lica, koji su u većini slučajeva unikatni za osobu. [1]

### 3.1. Način rada tehnologije za prepoznavanje lica

Sustav za prepoznavanje lica radi tako da pomoću kamere snimaju sliku ili video lica, bilo to dvodimenzionalno ili trodimenzionalno ovisno o uređaju za snimanje, te detektira da li i gdje postoje lica. Nakon toga obrezuje lice iz slike ili videa te ih matematički analizira i verificira uspoređujući s bazom podataka. [1]

Postoji više načina prepoznavanja lica, te su oni sljedeći [35]:

- Statistički pristup
  - o Ovom metodom su uzorci lica izraženi kao značajke, te je svaka slika skup značajki koje možemo obraditi pomoću statističkog alata. Neki od tih alata su:
    - Analiza glavnih komponenti (PCA)
    - Linearna diskriminantna analiza (LDA)
    - Projekcije očuvanja lokaliteta (LPP)
- Utemeljena na izgledu
  - o Kod ove metode se koriste više slika te se pojedina slika smatra višedimenzionalnim vektorom. Pokušava se iz slika izvesti set značajki.
- Utemeljena na modelu
  - o Ovom metodom se radi model lica prema primljenom licu, te se prema parametrima modela prepoznaje lice.
- Preklapanje predložaka
  - o Kod ove metode lice podudara s uzorcima i predlošcima, te se mjeri razina podudaranja.
- Neuronske mreže
  - o Neuronske mreže se mogu koristiti za obradu slike, prepoznavanje uzorka i klasifikaciju, te su zbog toga vrlo koristan alat za prepoznavanje lica.

Sustav pretvara rezultate u niz brojeva ili vektor koji se zove otisak lica, te kao otisak prsta smatra se unikatnim za svaku osobu. Nakon što se pretvori može se s njime uspoređivati i verificirati po potrebi. [2]

## 3.2. Faktori prepoznavanja lica

Zbog velike moguće razlike između slika ili videa za uspoređivanje s podacima u bazi podataka, potrebno je uspostaviti određene uvjete kojim bi se mogli standardizirati unosi za bolje performanse sustava i poboljšati točnost. Uvjeti ili faktori se mogu podijeliti na intrinzične i ekstrinzične.

Intrinzični faktori se odnose na načine koje osoba može promijeniti izgled lica, te su oni:

- Starenje
  - o Kako osoba stari tako se mijenja izgled lica, od isušavanja kože do pojava bora i madeža, lice se mijenja s odrastanjem i starenjem. [26]
- Izraz lica
  - o Promjenom izraza lica se mijenja lokacija različitih dijelova lica koji se koriste za prepoznavanje, čime se uvodi neizvjesnost kod sustava za prepoznavanje lica. [26]
- Plastična operacija
  - o Plastičnom operacijom se mijenjaju određeni dijelovi lica, čime se može uvesti neizvjesnost u sustav. [26]

Ekstrinzični faktori se odnose na načine na koje se može promijeniti izgled lica u slici, od vanjskih izvora. Ti faktori su:

- Skrivenost lica
  - o Ako osoba ima nešto na licu, poput sunčanih naočala ili maske, time se smanjuje vidljivost lica i smanjuje točnost prepoznavanja. [26]
- Niska rezolucija slike
  - o Ako slika ima nisku rezoluciju, teže je sustavu prepoznati dijelove lica potrebne za prepoznavanje. [26]
- Buka
  - o Slike mogu imati određene distorzije koje nastaju kod kreiranja slike, čime se otežava prepoznavanje. [26]
- Osvjetljenje

- Osvjetljenje ima veliki utjecaj na izgled lica, te postoje različiti načina pristupa osvjetljenju. [26]
- Poza osobe
  - Različite poze mogu otežati dobivanje slika cijelog lica za prepoznavanje, te su često sustavi trenirani na samo prednjim pogledima. [26]

Postoji i više faktora koji utječu na brzinu prepoznavanja lica, koji se odnose na cijeli sustav. Ti faktori su:

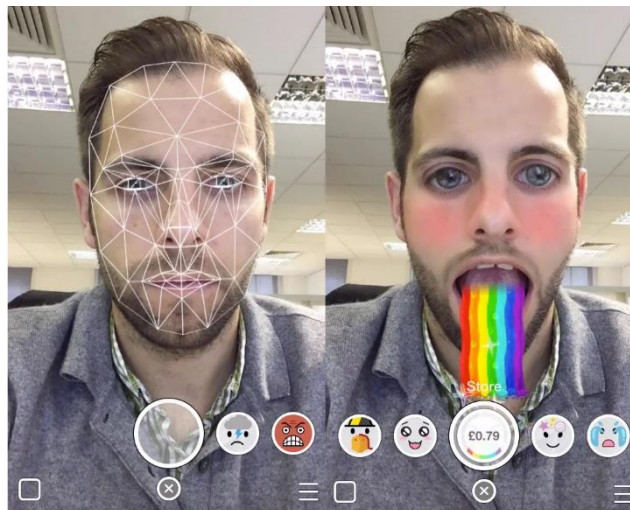
- Okolina
  - Sličnost okoline olakšava sustava da lakše uspoređuje lice, umjesto da se zbunjuje zbog okoline. [4]
- Dob slike
  - Starost slike je važna za prepoznavanje, jer što je aktualnija slika u bazi podataka točnije će je sustav prepoznati tijekom ispitivanja. [4]
- Konzistentna kamera
  - Konzistentne optičke karakteristike kamere koja se koristi za početnu sliku i sliku za uspoređivanje povećavaju točnost sustava. [4]
- Veličina baze podataka
  - Što više slika postoji za osobu sa manjim razlikama među njima (više kutova slikanja, različita osvjetljenja) precizniji će sustav biti s pogađanjem. [4]

### 3.3. Upotrebe tehnologije za prepoznavanje lica

Razvojem tehnologije dolazi do ubrzanog i olakšanog korištenja prepoznavanja lica za verifikaciju identiteta. Neke od tih upotreba su:

- Pametni mobiteli
  - 2017. Apple je predstavio iPhone X i Face ID koji je omogućio vlasniku da koristi svoje lice za otključavanje mobitela. Ta značajka je brzo postala vrlo popularna te se proširila i na druge proizvođače poput Samsung-a, Huawei-a i OnePlus. [3]
- Kontrola pristupa

- Mogućnost verifikacije lica se može koristiti za efikasnije blokiranje pristupa neželjenim osoba određenim mjestima, poput ureda ili prostorije za zaposlene, gdje se smanjuje mogućnost napada pomoću curenja lozinki ili izgubljenih kartica za ulaz. [27]
- Društvene mreže
  - Facebook je bila jedna od prvih velikih mreža koja koristi prepoznavanje lica za označavanje ljudi u fotografijama, dok je snapchat počeo koristiti tu tehnologiju za filtere koji omogućuju ljudima da mijenjaju izgled svojih crta lica, poput kontura lica, oblike jagodica itd. [3]
  - Na slici vidimo kako aplikacija Snapchat detektira lica i obilježja lica, te ih izmjenjuje ovisno o odabranom filteru.



Slika 1: Primjer snapchat filtera

Izvor: <https://medium.com/@anidaro/how-snapchats-filters-work-86973c3e2e9f>

- PolICIJA i nacionalna sigurnost
  - Sigurnosne kamere i video sigurnosni sustavi počinju se ugrađivati sa sustavima za prepoznavanje lica, koji pomoću policijske baze podataka pokušavaju prepoznati poznate zločince i ostale poznate kriminalce u većim grupama. [3]
  - Može se i koristiti za prepoznavanje osoba koje se ne mogu same prepoznati, poput starijih osoba s demencijom ili alzheimerovom bolesti, te osobe koje pate od amnezije. [27]
- Reklame

- Praćenjem ekspresije lica, sustav može odabrati reklame koje su primjerene za prepoznate osjećaje, te omogućuje promjenu reklama u slučaju dosade. [3]
- Isto tako može prepoznati registrirane kupce i prikazati im ponude bazirane na prijašnjim kupovinama ili koje bih željeli vidjeti. Ako osoba nije registrirana u trgovini onda se mogu prikazivati reklame ovisno o dobi, spolu ili osjećaju. [27]
- Granice
  - Na zrakoplovnim lukama i granicama su se počeli koristiti sustavi za prepoznavanje lica, koji provjeravaju točnost slike na putovnicama i provjeravaju za poznate zločince. [3]
  - Isto tako se koriste za ubrzavanje ukrcavanja na avione i kontrolu imigracije. [27]

### 3.4. Prednosti i nedostaci

Kao i svaka nova tehnologija, pogotovo u svrhe verifikacije i identifikacije, ona ima određene prednosti i nedostatke. Prednosti su vezane uz pronalazak osoba i udobnosti, dok nedostaci govore o privatnosti i sigurnosti.

#### 3.4.1. Prednosti

Postoji više prednosti korištenja tehnologije za prepoznavanje lica, koje ovise o području i načinu primjene. Najveća domena primjene bi bila u pronalasku i identifikaciji osoba, dok ima prednosti kroz više različitih domena.

- Pronalaženje nestalih osoba i kriminalaca
  - Kao što je već bilo spomenuto, skeniranjem lica mogu se prepoznati nestale osobe ili kriminalci na raznim mjestima, što omogućava preciznije lociranje tih osoba. To se može raditi pomoću slika na internetu ili pomoću video sigurnosnih sustava te može uvelike ubrzati potragu. [5]
  - Koristeći video sigurnosnih sustav na mjestima sa velikim prometom ljudi, poput zračnih luka ili trgovačkih centara, može se omogućiti veliko skeniranje ljudi na već postojećim sustavima. [27]
- Zaštita od krađe
  - Ako već poznati kriminalci dođu u trgovinu, osiguranje se može fokusirati na njih da se zaštiti roba. Isto tako ako dođe do krađe može

se dodati nova osoba u bazu podataka što bi sprječavalo ponavljanje zločina. [5]

- Olakšavanje kupnje
  - o Sustav u trgovinama može naplaćivati na račun kupca verifikacijom pomoću lica. Time se ubrzava proces kupnje i smanjuje šansa za nelegitimne kupnje. [5]
  - o Može se koristiti i za dvostruku autentikaciju uz pomoć kartice ili osobnom za veće kupovine. [27]
- Manje dodira
  - o Prepoznavanje lica se obavlja bez dodira, čime se može uvelike smanjiti prijenos bakterija i virusa za provjeru. Pandemija u 2020. je pokazala moguću brzinu širenja te važnost smanjivanja dodira kod većih grupa ljudi. [5]
  - o Može se koristiti i da se provjeri da li osoba nosi pravilno masku, te čak i provjeriti temperatura osobe, što može biti vrlo korisno za određene sektore u bolnici ili tijekom pandemija. [27]

### 3.4.2. Nedostaci

Svaka nova tehnologija sa sobom donosi određene nedostatke, bilo to u vezi same tehnologije ili implementacije te tehnologije. Prepoznavanje lica nije iznimka, te treba pristupiti takvoj tehnologiji s određenom mjerom opreza i pripreme za kvalitetnu implementaciju.

- Privatnost
  - o Sustavi za prepoznavanje lica imaju bazu podataka s velikim brojem podataka i lica u njemu, čime se javlja problem privatnosti. Povezivanjem lica s imenom i drugim osobnim podacima dolazi do važnih pitanja za korištenje takve tehnologije. Takva pitanja su vrlo važna za sigurnost i potrebni bi bili kvalitetni odgovori da bi se riješio problem privatnosti. Neka od tih pitanja su:
    - Da li su sudionici dali svoj pristanak za korištenje slika u bazi podataka za identifikaciju?
    - Tko sve ima pristup bazi podataka?
    - Jesu li politike za obradu podataka navedene i dobro promišljene?
    - Da li se informacije koje su dobivene koriste za što su namijenjene? [4]

- Važno je napomenuti da se kod korištenja takve tehnologije u javnim mjestima osoba može osjećati neugodno znajući da ih prati kamera koja ih prepoznaje. Time se može uzrokovati efekt „velikog brata“ koji prati svaki pokret osobe, što uzrokuje velike neugode ljudima. [5]
- Pravednost
  - Kod implementacije tehnologije za prepoznavanje lica dolazi do pitanja o jednakosti sustava u prepoznavanju, te mogućnosti diskriminacije. Ako neke grupe ljudi dobivaju veći broj lažnih pozitivnih rezultata ili se više puta prepoznaju može doći do profiliranja ili diskriminiranja na osnovi lošeg sustava. Radi toga je potrebno ekstenzivno testiranje i treniranje sustava na većem broju različitih ljudi s različitim pozadinama (različite rase, spola, godina itd.). [4] [5]
- Sigurnost
  - Zbog rada s verifikacijom ljudi, sigurnost za sustava prepoznavanje lica bi trebala biti vrlo visoka, slično kao i kod drugih biometrijskih metrika. Postavljaju se pitanja o zaštiti podataka, te zaštiti u slučaju curenje podataka. Pošto se slika osobe spaja s identitetom osobe, zlonamjerni upad u bazu podataka može imati velike posljedice za žrtve. Isto tako je potrebno spomenuti opasnost od uhođenja pogotovo u javnim prostorima koji koriste sustav za prepoznavanje lica. [5]
  - Važne su i posljedice u slučaju da osoba ne daje svoj pristanak na korištenje lica u svrhe prepoznavanje, te posljedice u slučaju ako je u pitanju sigurnost više osoba. Važno je i pitanje posljedica za lažne pozitivne rezultate ili diskriminatornih trendova, te kako osigurati da sustav povećava sigurnost. [4]



## 4. Neuronske mreže za prepoznavanje lica

Od izuma računala i softvera koji se koristi na računalu ljudi su zamišljali robote i umjetnu inteligenciju koja bi bila pametna kao najpametniji ljudi. Iako su ideje kako bi se izradila takva neuronska mreža postojale duže vrijeme, nisu se implementirale dok tehnologija nije dostigla određenu razinu performansi.

### 4.1. Neuronske mreže

Neuronske mreže je metoda umjetne inteligencije koja uči računala da procesuiraju podatke na način kao i ljudski mozak. Koristi međusobno povezane čvorove (ili neurone) u slojevitoj strukturi za učenje, što je modelirano prema ljudskom mozgu. [6]

Time se stvara adaptivni sustav koji uči na pogreškama i kontinuirano se poboljšava, tj. uči. Pomoću toga možemo naučiti (istrenirati) sustav za obavljanje složenijih problema, od kreiranja sažetka dokumenata do vizualnog prepoznavanja.

### 4.2. Struktura neuronskih mreža

Osnova neuronskih mreža je čvor ili neuron koji procesuiraju ulazni podatak ili informaciju i njegove veze s drugim čvorovima. Duboke neuronske mreže mogu imati više milijuna međusobno povezanih čvorova. Oni su podijeljeni na 3 sloja:

- Ulazni sloj
  - o Sloj koji prima podatke i šalje ih sljedećim slojevima za daljnju obradu. Strukturu ovog sloja se određuju mogući ulazi, te kako će se poslati sljedećim slojevima. [6]
- Skriveni sloj
  - o Čvorovi u skrivenom sloju primaju ulazne podatke iz ulaznog sloja ili drugih skrivenih slojeva, te analizira i procesuiraju primljene podatke i prosljeđuje ih sljedećem sloju. Neuronske mreže mogu imati velik broj skrivenih slojeva. [6]
- Izlazni sloj
  - o Izlazni sloj je konačni rezultat procesuiranja te ovisno o tipu neuronske mreže može imati više čvorova. U ovom sloju se određuje kakvi će biti izlazi i što možemo s njima raditi. [6]

### 4.2.1. Težina

Zbog velike važnosti veza između čvorova, neuronske mreže imaju poseban broj, zvan težina, koji se dodjeljuje svakom čvoru da reprezentira jačinu konekcije između dva čvora. Što je veća težina, time čvor ima bolju povezanost i veći utjecaj na te čvorove, dok manji broj signalizira manji utjecaj i jačinu signala. [6]

Čvor je dakle funkcija koja procesira težine od svih veza s čvorovima od prijašnjeg sloja, te time određuje težine za svoje konekcije. Taj proces se ponavlja za svaki čvor u svakom sloju, te se prema rezultatima određuje koje težine su ispravne. Taj proces može se raditi manualno, ali je to vrlo nepraktično zbog ogromnog broja veza i čvorova. Radi toga se češće koristi metoda povratnog širenja (back propagation), gdje se uspoređuje predviđeni odgovor s točnim, te ovisno o razlici mreža mijenja težine da se dođe bliže točnom odgovoru.

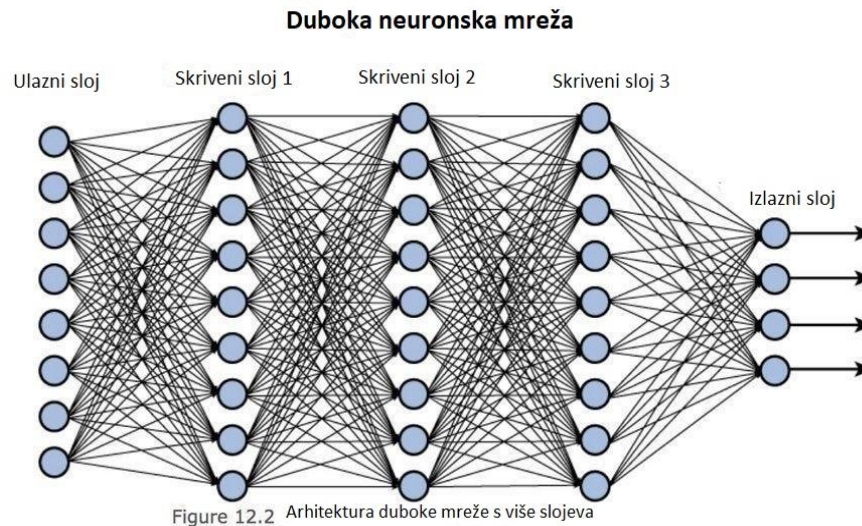
Važno je još i spomenuti aktivacijsku funkciju, pomoću koje se određuje da li će se neuron „upaliti“, tj. uključiti i poslati vrijednost sljedećem sloju. Postoje dvije vrste funkcija [25]:

- Linearne funkcije: izlaz funkcije je rezultat izračuna, te se takav šalje. Rijetko se koriste zbog svoje nefleksibilnosti.
- Nelinearne funkcije: izlaz funkcije može biti različit te ovisi o kojoj funkciji se radi. Postoji više vrsta:
  - o Sigmoidne, Tanh, ReLU itd.

Na primjer, ako imamo neuronsku mrežu za prepoznavanje geometrijskih oblika. Ulazni sloj je oblik koji želimo prepoznati, te kroz skriveni sloj mreža kalkulira koji bih to bio oblik. Na kraju na izlaznom sloju mreža predviđa s raznim razinama sigurnosti koji je oblik bio uneseni. Ovisno o rezultatima i predviđanjima, neuronska mreža mijenja težine između čvorova da točni odgovori budu što češće odabrani sa velikom razinom sigurnosti.

To znači da svaki ulazni podatak prolazi kroz čvorove u skrivenom sloju, te ovisno o vezama i težinama između čvorova sustav dolazi do određenog izlaznog čvor. Time se omogućuje neuronskim mrežama da „misle“, tj. da naprave odluku ovisno o tome što smatra točnim. Radi toga duboke neuronske mreže zahtijevaju velik broj testnih podataka i treninga prije nego budu korisne.

Na slici 2 se prikazuje struktura duboke neuronske mreže, te vidimo ulazni, skriveni i izlazni sloj, te njihovu međusobno povezanost.



Slika 2: Duboka neuronska mreža s više slojeva

Izvor: <https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964>

### 4.3. Tipovi neuronskih mreža

Postoji više vrsta neuronskih mreža, od osnovnih mreža sa jednim slojem do dubokih sa mnogo više slojeva. Isto tako svrha i način procesuiranja mijenjaju i kreiraju nove tipove neuronskih mreža.

#### 4.3.1. Feed forward neuronska mreža

Najjednostavnija neuronska mreža je ona gdje podaci putuju u samo jednom smjeru kroz skriveni sloj. U ovoj mreži, zbog svoje jednostavnosti, skriveni sloj ne mora ni postajati. Težina je statična, te su čvorovi dosta jednostavni, sa jednostavnim funkcijama i procesima. Prednosti ovog tipa neuronske mreže je njena jednostavnost i brzina obrade. Nedostatak je da se ne može koristiti za duboke neuronske mreže zbog svoje jednostavnosti. [7]

#### 4.3.2. Višeslojni perceptron

Kompleksnija neuronska mreža ima barem 3 sloja u skrivenom sloju, gdje su svi čvorovi međusobno povezani. Podaci mogu putovati u oba smjera, te je najčešća varijanta neuronske mreže s povratnim širenjem (backpropagation), pomoću koje se omogućava samo korekciju težina. Time se omogućava neuronskoj mreži da sama uči pomoću testnih podataka, gdje se težine ispravljaju od strane sustava, gdje nema ljudskog unosa. To se radi tako da sustav uspoređuje izlazne vrijednosti koje je predvidio i testne vrijednosti. [7]

Prednosti ovog tipa neuronske mreže je mogućnost korištenja za duboke neuronske mreže, dok su nedostaci kompleksnost razvijanja takvog sustava i potrebno vrijeme razvijanja, kao i brzina obrade.

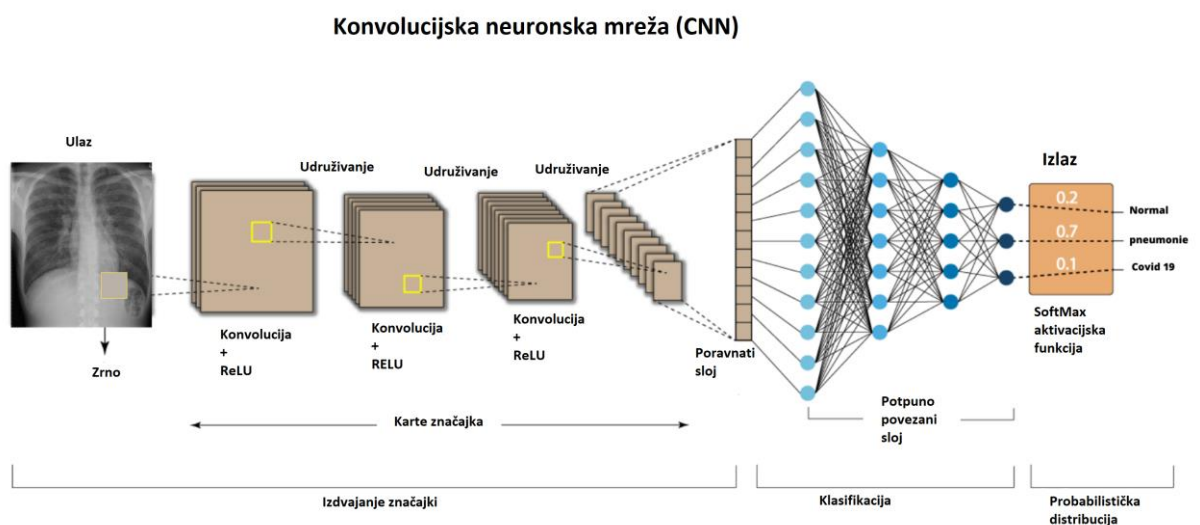
### 4.3.3. Konvolucijska neuronska mreža

Konvolucijska neuronska mreža sadrži trodimenzionalnu konfiguraciju čvorova gdje svaki čvor ima svoj filter koji prolazi kroz određeni dio ulaza(slike) dok ne prođe kroz cijelu sliku. Svaki filter je namješten da prepozna neki uzorak u slici te ako ga je prepoznao filter to označava. Slojevi bliži ulaznom sloju su najčešće vrlo apstraktni, te provjeravaju rubove i krivulje, dok su dublji slojevi specifičniji i provjeravaju složenije elemente poput lica ili oblika objekta. [7]

Rezultat prolaska filtera kroz sliku je matrica ili vektor koja pokazuje da li se prepoznao uzorak filtera i na kojem mjestu, te se zove mapa značajki. Nakon što se izračuna mapa značajki, da se smanji broj kalkulacija potrebnih za sljedeći sloj potrebno je udružiti mape značajki od sloja. Svako udruživanje ima veličinu matrice ili vektora koji služi kao filter, te korak za koji se filter pomiče po mapi značajki. [24]

To se može učiniti na dva načina:

- Maksimalno udruživanje(max pooling): ovom metodom udruživanja se odaberi najveći element koji filter nađe u dijelu mape značajke. [24]
- Udruživanje srednje vrijednost(average pooling): ovom metodom udruživanja filter izračunava srednju vrijednost od elemenata u dijelu mape značajke. [24]



Slika 3: Konvolucijska neuronska mreža

Izvor slike: <https://www.mdpi.com/2079-9292/11/11/1775>

Na slici 3 se prikazuje rad konvolucijske neuronske mreže. Na ulazu mreža prima sliku, u ovoj slučaju sliku pluća. Nakon toga mreža obrađuje sliku pomoću konvolucija sa ReLU aktivacijskom funkcijom, nakon kojih se odrađuje udruživanje. Nakon što slika prođe kroz sve konvolucije, prenosi se na potpuno povezani sloj koji pomoću SoftMax aktivacijske funkcije klasificira sliku, u ovoj slučaju prepoznavanje bolesti pluća.

## 4.4. Primjena neuronskih mreža za prepoznavanje lica

Da bi sustav mogao prepoznati lice na slici ili videu, sustav prvo mora pripremiti i pretvoriti sliku u standardizirani oblik koji on prepoznaje. Taj proces možemo podijeliti na 4 procesa:

### Detektiranje lica

- Prvi korak prepoznavanja lica, za koji sustav treba prepoznati lice od okoline ili drugih dijelova tijela. Ovo može biti vrlo zahtjevno za računalo zbog velike varijacije lica koje ljudi mogu imati, te slika ili video može imati različite uvjete (poput osvjetljenja, kuta slike, naočale, kape itd.) čime se dodatno otežava posao. Zbog toga je potrebno osmisliti i testirati sustav koji će raditi i u težim uvjetima. Cilj je da sustav prepozna lice iz različitih poza, neovisno o kutu snimanja ili uvjetima okoline. [8]
- Postoje 4 metode detektiranja lica, ali algoritmi za detektiranje lica mogu koristiti više metoda. Te metode su:
  - Utemeljene na znanju:
    - Bazira se na ljudskom znanju o licu, poput „lice mora imati nos, usta, oči na određenim udaljenostima“. Takav pristup ima velikih poteškoća ovisno da li su pravila previše striktna ili previše slobodna. [32]
  - Utemeljene na značajkama:
    - Ova metoda se bazira na ekstrakciji strukturnih značajki lica. Nakon treniranja klasifikator može razlikovati lica od ne-lica. [32]
  - Podudaranje predložaka
    - Koristi se predložak prema kojem se uspoređuje slika, te se tako pokušava detektirati lice na slici. Ova metoda ima slične nedostatke kao i metoda utemeljena na znanju, tj. vrlo rigidna i uska definicija za lice. [32]
  - Utemeljene na izgledu

- Metoda se temelji na skupu lica za treniranje sustava da nađe modela lica. Metoda se oslanja na tehnike analize i strojnog učenja da pronade karakteristike lica na slici. [32]

#### Poravnavanje lica

- Nakon što je sustav prepoznao lice, potrebno je to lice poravnati i standardizirati u oblik koji sustav može prepoznati i procesuirati. Time se omogućava uspoređivanje jer će slike lica biti u istom ili sličnom formatu. [8]

#### Izdvajanje značajki lica

- Nakon što je sustav standardizirao sliku lica, on može izdvojiti najvažnije značajke koje provjerava (ili one značajke koje su prisutne), koje će zatim biti pretvorene u otisak lica za provjeravanje. [8]

#### Prepoznavanje lica

- Zadnji korak je samo prepoznavanje lica, koji može imati različite procese ovisno o tipu i vrsti sustava i za što se koristi. Tako možemo prepoznati nekoliko glavnih upotreba za takav sustav, koje su:
  - Verifikacija osobe: sustav provjerava je li korisnik osoba za koju se predstavlja.
  - Identifikacija osobe: sustav provjerava osobu s bazom podataka da se sazna tko je ta osoba.
  - Transformacija lica: sustav treba transformirati dobiveno lice (dodati ili promijeniti karakteristike lica). [8]

## 5. Pregled literature – Neuronske mreže

Postoji velik broj literature o korištenju neuronskih mreža za prepoznavanje lica, te se generalno mogu podijeliti na preglede literature i implementacije različitih neuronskih mreža. Od više različitih neuronskih mreža, najčešće su korištene konvolucijske mreže zbog svoje snage kod obrade slika.

Efektivnost neuronske mreže leži u ne linearnom načinu rada, čime je ekstrakcija osobina efikasnija i brža od linearnih načina. Konstrukcija neuronske mreže je vrlo važna za efikasno i točno prepoznavanje, te se u tu svrhu se najčešće koriste konvolucijske neuronske mreže i višeslojni perceptroni s povratnim širenjem. [14] [15]

Prepoznavanje lica se smatra jednim od najpopularnijih načina korištenja neuronskih mreža u svrhe biometrijske identifikacije, zbog toga jer tako većina ljudi prepoznaju druge osobe. [14] [15]

Jedan od velikih limita za prepoznavanje objekata i lica sa slika je potrebna veličina ulaznog sloja, gdje bi slika veličine 28x28 piksela trebala preko 784 neurona u ulaznom sloju i 784 težina za svaku poveznicu sa skrivenim slojem. Ako bi koristili slike veličine 64 x 64 piksela bilo bi 12 288 neurona u ulaznom sloju i njihovih težina, čime dolazi do prevelikog povećanja kompleksnosti neuronske mreže. [16]

### 5.1. Konvolucijske neuronske mreže

Odabrana su dva članka za konvolucijsku neuronsku mrežu za prepoznavanje lica. Prvi znanstveni članak prolazi kroz više trenutnih modela za prepoznavanje lica te uspoređuje njihovu točnost, te zatim njihovu implementaciju pomoću već treniranih konvolucijskih neuronskih mreža AlexNet i ResNet-50 s potpornim vektorskim strojem za klasifikaciju. [17]

Drugi članak opisuje strukturu implementacije konvolucijske neuronske mreže, koja dijeli CNN na 9 slojeva, 3 konvolucijska, 2 objedinjujuća sloja, 2 potpuno povezana sloja i jedan softmax regresijski sloj. [20]

Prije nego se može početi s prepoznavanjem lica, potrebno je promijeniti veličinu slike da bude prikladna za CNN model, te se crno-bijele slike pretvaraju u RGB slike. Sljedeći proces je prikaz lica, gdje se koriste dvije CNN mreže AlexNet i ResNet-50 za izdvajanje crta lica koja se mogu koristiti u klasifikaciji sa sljedećim CNN. Na kraju su koristili dva pristupa klasifikacije,

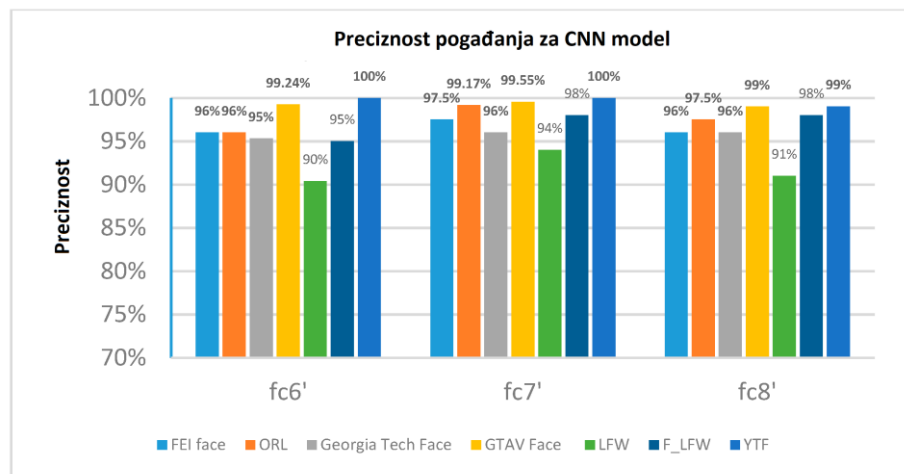
prvi s potpornim vektorskim strojem za klasifikaciju, a drugi s prenesenim učenjem od AlexNet CNN-a. [17]

Za svaki skup podataka je potrebno promijeniti veličinu slike na 227 x 227 za ulaz na AlexNet i 224 x 224 za ulaz na ResNet-50. Performanse su mjerene prema točnosti prepoznavanja lica. [17]

Korišteni skupovi podataka su ORL, GTAV, Georgia Tech face database, FEI face, LFW, F\_LFW(Frontalized Labeled Faces in the Wild), YTF, DB\_collection.

Rezultati CNN mreže AlexNet s potpornim vektorskim strojem za klasifikaciju:

- AlexNet ima više slojeva za prepoznavanje, ali nisu svi potrebni za izdvajanje osobnosti lica. Članak gleda rezultate za slojeve fc6, fc7 i fc8.
- Osobnosti sa sloja fc7 su imale najtočnije rezultate, dok je fc6 imao najlošije rezultate, što vidimo na slici 4. [17]



Slika 4: Rezultati CNN mreže AlexNet s potpornim vektorskim strojem za klasifikaciju

Izvor: Almabdy, S., & Elrefaei, L. (2019). Deep convolutional neural network-based approaches for face recognition. *Applied Sciences*, 9(20), 4397.

Rezultati CNN mreže ResNet-50 s potpornim vektorskim strojem za klasifikaciju [17]:

- Koristeći duboke slojeve za prepoznavanje posebnih značajka je davalo bolje rezultate, te su se u zadnjem sloju izdvojile značajke.
- Rezultati za ORL, GTAV i YTF iznose 100% točnosti, dok ih slijedi točnost za skup podataka FEI sa 98.5%. GTAV i Georgia Tech Face imaju istu točnost od 96%. Najlošije rezultate je skup podataka LFW sa 94% točnosti.

Rezultati prijenosa treninga AlexNet za izdvajanje osobnosti i klasifikaciju [17]:



- Dodavanjem posebnog sloja za klasifikaciju na kraj AlexNet slojeva, omogućava klasifikaciju slike unutar neuronske mreže.
- Rezultati za Georgia Tech Face, GTAV i YTF iznose 100% točnosti. Sljedeći najbolji rezultat je na skupu podataka F\_LFW, sa 99.89% točnosti. Nakon toga slijedi ga skup podataka ORL sa 99.17% točnosti te FEI sa 98.7%. Najlošiji skup podataka je LFW sa 95.63% točnosti.

U drugom članku se koristi konvolucijska neuronska mreža sa 9 slojeva, od kojih su 3 konvolucijska sloja, 2 slojeva udruživanja(objedinjivanja), 2 potpuno povezana slojeva i 1 SoftMax regresijski sloj. Svaki sloj predstavlja linearno mapiranje različitih vrsta podataka. Svaki konvolucijski i objedinjujući sloj se sastoji od više mapa značajki, a svaka mapa značajki se sastoji od više neurona(čvorova). Svaka mapa značajki je ulaz za sljedeći sloj. [20]

Konvolucijski slojevi obrađuju sliku pomoću konvolucijskih jezgra gdje se stvara mapa značajke, dok u objedinjujućem sloju se smanjuje veličina te mape značajki. Potpuno povezani slojevi igraju ulogu klasifikatora. Zbog kompleksnosti lica, softmax klasifikator se koristi za klasificiranje na zadnjem sloju. [20]

Skupovi podataka na kojem je sustav testirani je ORL skup podataka, koji sadrži 400 slika od 40 osoba; i AR skup podataka, koji sadrži 2600 slika za 100 osoba.

- Točnost prepoznavanja lica za skup podataka ORL iznosi 99.82%. [20]
- Točnost prepoznavanja lica za skup podataka AR iznosi 99.78%. [20]

### 5.1.1. DeepFace

DeepFace je neuronska mreža za prepoznavanje lica napravljena od strane Facebook-a. Koristi model dubokog učenje, te radi tako da za poravnavanje lica kreira 3D model toga lica. [9]

DeepFace je od početka bio povezan s kontroverzama, od načina skupljanja slika za treniranje mreže do korištenja mreže za prepoznavanje. Nakon velikog pritiska od korisnika i vlada neuronska mreža je ugašena i omogućeno je korisnicima brisanje podataka povezanih uz neuronsku mrežu. Facebook je tako jedan od prvih medijskih divova koji je prestao promovirati i ulagati u tehnologiju prepoznavanja lica zbog problema s privatnosti. [33]

Arhitektura se bazira na tome da kada se poravnanje izvrši kreira se 3D model, gdje je lokacija svakog područja lica fiksna na razini piksela. Zbog toga je moguće naučiti RGB vrijednosti piksela, čime se smanjuje broj konvolucija u konvolucijskoj neuronskoj mreži. [9]

Proces poravnavanje lica počinje s 2D poravnavanjem, gdje nakon što se lice detektira i obreže iz slike, sustav traži 6 referentnih točka na licu. Te točke su centar očiju, vrh nosa i

lokacija usta. Nakon toga se za izradu 3D modela lokalizira dodatnih 67 referentnih točaka u 2D slici, prema kojima se izrađuje 3D model. Važna je još i frontalizacija, gdje ako lice nije ravno ispred kamere, potrebno je učiniti neko savijanje da bih se mogao napraviti bolji 3D model. [9]

Mreža ima 8 slojeva, te su podijeljeni na sljedeći način [9]:

- Nakon detekcije i frontalizacije, prvi sloj je konvolucijski sloj sa 32 filtera veličine 11x11x3.
- Sljedeći sloj je maksimalno udruživanje, veličine 3x3 sa korakom od 2.
- Nakon toga slijedi konvolucijski sloj sa 16 filtera veličine 9x9x16.
- Sljedeća 3 sloja su lokalno povezani, te svaka lokacija za mape značajke nauči drugi set filtera.
- Zadnja dva sloja su potpuno povezani, te mogu uhvatiti korelacije između značajka udaljenih dijelova slika lica.

Treniran je na skupu podataka SFC(Social Face Classification) koji uključuje preko 4.4 milijuna označenih lica od 4030 ljudi. Ljudska greška kod označavanja tih podataka iznosi 3%. Nakon što je sustav bio istreniran, testiran je na 2 skupa podataka:

- LFW(Labelled Faces in the Wild) koji uključuje preko 13 000 slika od 5749 poznatih osoba. [9]
- YTF(YouTube Faces) koji skuplja preko 3400 videa od preko 1595 osoba(neke od tih osoba su i dio LFW skupa podataka) [9]

Rezultati za skup podataka je točnost od 97.35%, dok ljudska točnost iznosi 97%. Bez 3D poravnavanja točnost iznosi 94.3%. [9]

Rezultati za skup podataka YTF je točnost od 91.4%, te ako se uračunavaju poprave pogreške za nazive videa točnost dolazi do 92.5%. [9]

### **5.1.2. DeepID**

DeepID je osnova neuronske mreže za prepoznavanje lica, koja ima više različitih implementacija, te svaka ima svoj način da poboljša točnost prepoznavanja.

M<sup>2</sup> Deep-ID:

- Kod ovog modela se više 2D slika iste osobe pod različitim uvjetima dodaje u neuronsku mrežu i pusti da mreža izračuna apstraktniji prikaz značajki lica. Time se omogućava bolje prepoznavanje lica iz različitih kuteva slikanja. [10]

- Model ima 4 glavna modula. Prvi modul sadrži reprezentacijsku metodu učenja zvanu „multi-view representation“. U drugom modulu je agregirajući sloj koji agregira informacije iz različitih pogleda i kreira 2D reprezentaciju svih pogleda. Treći sloj uči agregirane značajke dok je zadnji modul softmax sloj za identificiranje klase. [10]
- Testiran je na skupu podataka IUST (Iran University of Science and Tehnology) koji ima 6552 slika lica od 504 osobe). Za svaku osobu ima slike s 5 poza sa neutralnim osvjetljenjem. [10]
- DeepID bez nadogradnja ili modifikacija je dobio točnost od 97%, dok sa multi-view Deep Face Recognition implementacijom je točnost 99% na skupu podataka. [10]

H-ELM-LRF-DeepID (Hybrid Locally Connected Extreme Learning Machine With DeepID)

- Hibridizacija ELM(extreme learning machine) okvira za uspoređivanje lica i DeepID mreže za izdvajanje značajki lica. Simplificirani algoritam učenja čija je strategija prilagođavanje slobodnih neurona čak i kada su izlazni oblici i funkcije čvorova su nepoznati. Time se omogućava lakša implementacija i manje procesne zahtjevnosti. [11]
- U mreži se koristi ELM sa lokalnim receptivnim poljima, te korištenjem tih lokalno receptivnih polja za dobivanje ELM. Takav ELM se zove ELM-LRF, te sadrži 2 dijela. Prvi dio je mapiranje značajki bez podešavanja, drugi dio je ELM učenje bazirano na reguliranom rješenju najmanjih kvadrata. [11]
- Mreža ne koristi duboki CNN okvir za verifikaciju lica, čime se smanjuju troškovi računanja tijekom treniranja. Umjesto toga, mreža koristi ELM kao podlogu za okvir verifikacije lica što olakšava implementaciju. [11]
- Testiran je na dva skupa podataka, YTF i LFW, te je postigao sljedeće rezultate [11]:
  - o Za skup podataka YTF točnost je 90.32%
  - o Za skupo podataka LFW točnost je 97.47%

### 5.1.3. FaceNet

FaceNet je unificirana neuronska mreža za verifikaciju, identifikaciju i grupiranje lica. FaceNet radi tako da uzme sliku osobe i koristi Euklidski prostora u koji se preslikavaju značajke, u kojem su udaljenosti povezane s mjerom sličnosti lica. Nakon što kreiramo taj prostor, zadatke identifikacije, verifikacije i grupiranja je moguće lagano implementirati koristeći standardne tehnike sa FaceNet ugrađivanjima kao vektorima značajka. [12] [13]

FaceNet uči na sljedeći način:

- Prvo selektira sliku kao sidro
- Nakon toga odaberi sliku iste osobe kao usidrena slika. Ta slika će služiti kao pozitivni primjer.
- Odabire sliku osobe različite od usidrene slike. Ta slika služi kao negativni primjer.
- Prilagodi parametre tako da je pozitivni primjer bliži usidrenoj slici od negativnog primjera.
- To se ponavlja dok su slike iste osobe što je moguće bliže jedna drugoj dok ostale su udaljene. [13]

Mreža je testirana na skupu podataka LFW i YTF.

- Rezultati na skupu podataka LFW iznosi 98.87%, te dostiže 99.63% ako se korisni dodatno poravnavanje lica(u tom slučaju bih se koristio dodatni detektor lica). [12]
- Rezultati na skupu podataka YTF iznosi 95.12%. [12]

#### 5.1.4. ResNet50

ResNet50 je rezidualna mreža koja je posebna vrsta konvolucijske neuronske mreže i ima 50 slojeva. Ime dobiva prema engleskom nazivu *Residual network* i broju slojeva. Bazirana je na ResNet-34, koja ima 34 sloja te koristi spajanje prečacima radi rješavanja problema nestajanja gradijenta, koji opisuje problem kod povratnog širenja. Problem koji se događa je smanjivanje vrijednosti gradijenta tijekom povratnog širenja čime dolazi do minimalne promjene težina, te se time usporava treniranje neuronske mreže. ResNet rješava taj problem pomoću spajanja slojeva prečacima(može se isto zvati preskakanje veze). [29]

Prvi sloj mreže je konvolucija sa znom veličine 7x7 i 64 različitih zrna sa korakom veličine 2, čime dobivamo prvi sloj. Nakon toga slijedi maksimalno udruživanje sa veličinom koraka 2. Sljedeća konvolucija sadrži 3 sloja sa 1x1, 64 znom, 3x3, 64 znom i zadnji sloj sa 3x3, 256 zrno. Ova 3 sloja se ponavljaju 3 puta čime dobivamo 9 slojeva. Nakon toga ponovo slijede 3 slojeva sa znom 1x1,128, 3x3,128 i 1x1,512 slojevi. Oni se ponavljaju 4 puta čime dobivamo 12 slojeva. [29]

Zatim slijede 3 slojeva sa znom 1x1,256 ; 3x3, 256 ; 1x1, 1024 koji se ponavljaju 6 puta, čime dobivamo 18 slojeva. Nakon toga ponovo slijede 3 slojeva sa znom 1x1, 512 ; 3x3, 512 ; 1x1,2048 koji se ponavljaju 3 puta za konačni broj slojeva od 9. Na kraju radi prosječno udruživanje i završava sa potpuno povezanim slojem od 1000 čvorova i softmax funkcijom, što znači da je broj slojeva za ovaj dio 1. [28]

Zbrajanjem ukupnih slojeva dobivamo 50 slojeva, ne računajući slojeve udruživanja. [28]

### 5.1.5.SENet50

SENet50 ima arhitekturu koja je bazirana na bloku „Squeeze-and-Excitation“, SE (kraće). SE blok adaptivno rekalibrira odgovore značajki eksplicitnim modeliranjem međuovisnosti između kanala. [30]

Squeeze dio:

- Problem kod konvolucijskih neuronskih mreža je lokalnost filtera koji prolaze kroz sliku zbog čega nisu u mogućnosti koristiti kontekstualne informacije izvan te regije.
- Radi toga se uvodi *stiskanje(squeeze)* globalne informacije u opis kanala. To se radi koristeći globalno prosječno udruživanje čime se generira statistika po kanalu.
- Izlaz je kolekcija lokalnih deskriptora čija statistika je ekspresivna za cijelu sliku. [31]

Excitation dio:

- Da bismo mogli iskoristiti informaciju skupljenu u *squeeze* dijelu, potrebna je sljedeća operacija čiji je cilj potpuno uhvatiti ovisnosti o kanalu. Da bi se to moglo izvršiti funkcija treba ispuniti dva uvjeta:
  - o Fleksibilnost – Mora biti sposobna naučiti nelinearne interakcije između kanala
  - o Mora naučiti ne-međusobno-isključivu vezu, jer želimo osigurati da više kanala bude naglašeno. [31]

Takav SE blok se može integrirati u standardne arhitekture umetanjem nakon nelinearnosti poslije konvolucije. Štoviše, fleksibilnost SE bloka znači da se može primijeniti na transformacije izvan standardnih konvolucija. Time se može SE blok staviti prije, poslije ili paralelno. [31]

Na slici 5 može vidjeti slojeve ResNet50 mreže i SE-ResNet-50.

Veličina izlaza	ResNet-50	SE-ResNet-50	SE-ResNeXt-50 (32 × 4d)
112 × 112	conv, 7 × 7, 64, stride 2		
56 × 56	maksimalno udruživanje , 3 × 3, stride 2		
	$\begin{bmatrix} \text{conv}, 1 \times 1, 64 \\ \text{conv}, 3 \times 3, 64 \\ \text{conv}, 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1 \times 1, 64 \\ \text{conv}, 3 \times 3, 64 \\ \text{conv}, 1 \times 1, 256 \\ \text{fc}, [16, 256] \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 256 \\ \text{fc}, [16, 256] \end{bmatrix} C = 32 \times 3$
28 × 28	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 512 \\ \text{fc}, [32, 512] \end{bmatrix} \times 4$	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 512 \\ \text{fc}, [32, 512] \end{bmatrix} C = 32 \times 4$
14 × 14	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 1024 \\ \text{fc}, [64, 1024] \end{bmatrix} \times 6$	$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 1024 \\ \text{fc}, [64, 1024] \end{bmatrix} C = 32 \times 6$
7 × 7	$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 2048 \\ \text{fc}, [128, 2048] \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1 \times 1, 1024 \\ \text{conv}, 3 \times 3, 1024 \\ \text{conv}, 1 \times 1, 2048 \\ \text{fc}, [128, 2048] \end{bmatrix} C = 32 \times 3$
1 × 1	Globalno prosječno udruživanje , 1000-d fc, softmax		

Slika 5: Pregled strukture ResNet-50 i SE-ResNet-50

Izvor: Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132-7141).

## 5.2. Višeslojni perceptron

Da bi se omogućila bolje efikasnost učenja sustava uvodi se povratno širenje, čime neuronska mreža može samostalno učiti prema testnim materijalima. Takva neuronska mreža se zove neuronska mreža povratne propagacije (back propagation neural network) ili BPNN ukratko. Ona obavlja samo korekciju tako da dobije povratne informacije o dobivenom rezultatu i prema tome ojačava točne konekcije ili oslabljuje ostale. Time se omogućava veći broj slojeva što poboljšava učinkovitost.

Prvi odabrani članak se bazira na podjeli slike na mreže, dok drugi članak spaja neuronsku mrežu s neizrazitom logikom. [18] [19]

Pretvaranjem slike lica u niz i uspoređivanjem s ostalim nizovima u bazi podataka može prepoznati lice. Problem se pojavljuje s veličinom takvog niza, gdje zbog svoje veličine smanjuje efikasnost sustava. Zbog toga su korištene različite metode za smanjivanje nedostatka implementacije ta članak odabire metodu dijeljenja slike prozore. [18]

Jedna od tih metoda je dijeljenje slike na prozore veličine 4x4 ili 8x8, gdje se slike iz baze podataka veličine 112 x 92 piksela. Nakon toga se podijele prozori koji su srednja vrijednost svakog prozora te se dobiva vektor značajki. Vektori značajki dolaze u dvije veličine, 16 x 400 za 4x4 prozore i 64 x 400 za 8x8 prozore. Nakon toga je svaka grupa značajki standardizirana da bude između 0 i 1 da se može procesuirati kroz neuronsku mrežu. Zbog toga, ulazne matrice su veličine 16 x 320 i 64 x 320, dok su testne matrice 16 x 80 i 64 x 80. [18]

Metoda je testirana na skupu podataka ORL, tako što su uzete dvije slike od osobe koje nisu korištenje za treniranje. Rezultati korištenjem te metodu su točnost od 94%. [18]

TABLE III. Rezultati iteracija

	iteracija	<i>lr.</i>	<i>mc.</i>	<i>%acc(max)</i>	<i>%acc(mean)</i>
4×4	500	0.1	0.9	65	61,5
	500	0.5	0.9	68,75	63,75
	500	0.9	0.9	72,5	60
	500	2.0	0.9	63,75	59
4×4	1000	0.1	0.9	78,75	68
	1000	0.5	0.9	78,75	69
	1000	0.9	0.9	70	64,75
	1000	2.0	0.9	75	69
4×4	2000	0.1	0.9	81,25	74,5
	2000	0.5	0.9	86,25	72,75
	2000	0.9	0.9	83,75	78
	2000	2.0	0.9	81,25	71,25
4×4	5000	0.1	0.9	85	79,75
	5000	0.5	0.9	85	77,5
	5000	0.9	0.9	88,75	79,5
	5000	2.0	0.9	81,25	76,25
8×8	500	0.1	0.9	81,25	74
	500	0.5	0.9	77,5	71
	500	0.9	0.9	78,75	74,25
	500	2.0	0.9	77,5	74
8×8	1000	0.1	0.9	95	85,25
	1000	0.5	0.9	90	84,25
	1000	0.9	0.9	92,5	86,75
	1000	2.0	0.9	86,25	81,75
8×8	2000	0.1	0.9	93,75	92,25
	2000	0.5	0.9	93,75	91,5
	2000	0.9	0.9	93,75	91,5
	2000	2.0	0.9	93,75	92,75
8×8	5000	0.1	0.9	96,25	94,25
	5000	0.5	0.9	96,25	93,25
	5000	0.9	0.9	93,75	92,25
	5000	2.0	0.9	93,75	92

Tablica 1: Rezultati BPNN neuronske mreže i metode prozora kroz iteracije

U tablici su prikazani rezultati kroz iteracije neuronske mreže, te se vidi da sa više tisuća iteracija mreža dostiže točnosti od 96.25%.

U drugom članku se spaja višeslojni perceptron s neizrazitom logikom(fuzzy logic) za prepoznavanje lica. Neuronske mreže su odabrane zbog svoje adaptivne mogućnosti i prepoznavanje uzorka. S druge strane neizrazita logika ima veliki potencijal zbog svoje mogućnosti rukovanja neizvjesnostima, nepreciznostima i neodređenostima. Spajanjem i hibridizacijom tih dvaju sustava oni postaju komplementarni te vrlo efektivni za rješavanje problema prepoznavanja lica. [19]

Spoj neizrazite logike i neuronske mreže je implementiran kao neizrazita neuronska mreža, gdje su čvorovi neizraziti umjesto čista vrijednosti. Tijekom prepoznavanja klasa, ako dolazi do preklapanja klasa, očito da postoje uzorci koji se testiraju u obje dvije klase, neovisno o stupnju pripadanja. Za identifikaciju granica između tih klasa, stupnjevi pripadanja uzorka

klasama igraju važnu ulogu. U ovom slučaju neizrazita logika je vrlo važna za opis različitih stupnjeva pripadanja istih uzorka klasama. [19]

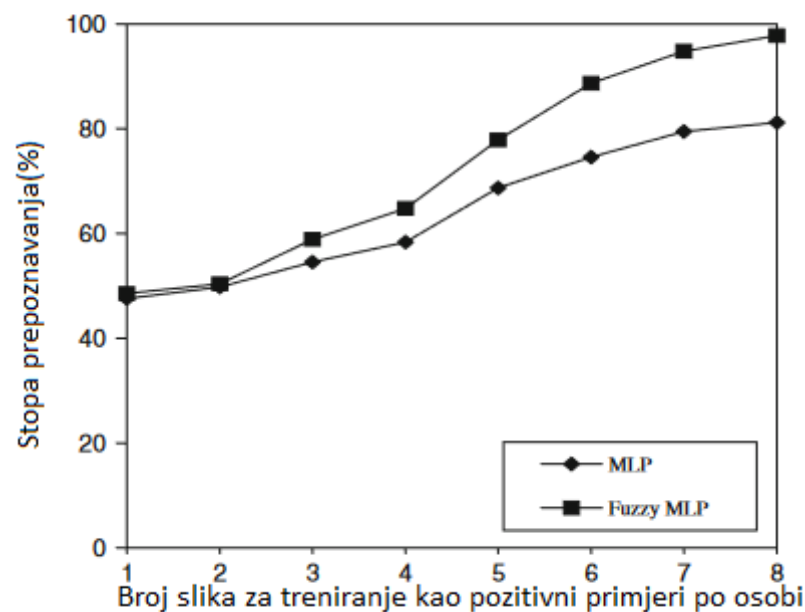
Baza podataka je Cambridge ORL, sa 400 slika od 40 osoba, a druga je njihova baza podataka sa 200 slika od 10 osoba, što im daje sveukupan broj od 600 slika za 50 osoba.

Kada sustav primi sliku lica, pomoću Gaborovog valića dobiva se 24 transformacije za svaku sliku. Zbog varijacija između te 24 transformacije kreira se vektor veličine 48, koji se fuzificira(eng. Fuzzified) i primjeni neizraziti MLP. [19]

- Osnova fuzificiranja vektora je da što je broj bliži 1, time je vrijednost pripadanja čiste vrijednosti veća, dok što je bliži 0 time je vrijednost pripadanja manja. [19]

Rezultati:

- Za ORL bazu podataka dobivena je točnost od 97.875% i 95.48% za sveukupnu bazu podataka.



Slika 6: Usporedba običnog višeslojnog perceptrona i neizrazitog perceptrona

Na slici 6 vidimo performanse od višeslojnog perceptrona i neizvjesnog perceptrona, te vidimo da stopa prepoznavanja lica se više povećava za neizvjesni perceptron sa više pozitivnih primjera po osobi.



## 6. Implementacija neuronske mreže za prepoznavanje lica

Za implementaciju sam odabrao Keras API za Tensorflow platformu za duboko učenje, zbog velike kompatibilnost s python-om, što je programski jezik u kojem sam radio implementaciju. Keras je API(Application Programming Interface) za duboko učenje napravljeno u python-u. Keras se ističe svojom jednostavnošću, fleksibilnosti i snagom. Korišteni su modeli ResNet50 i SENet50. [21]

Implementacija je rađena u python-u preko Google Colab, koji omogućava pisanje i izvršavanje programskog koda python-a u web pregledniku, čime se uklanja potreba za posebnim konfiguracijama i omogućava pristup računalstvu u oblaku.

### 6.1. Implementacija neuronskih mreža ResNet50 i SENet50 u Python-u

U mojoj implementaciji se koristi integracija s ResNet-50 koja se zove SE-ResNet-50, ili skraćeno SENet50.

Implementacija se bazira na dva članka koji opisuju kako implementirati keras u python, te kako koristiti tu implementaciju za prepoznavanje lica. [22] [23]

Koristio sam te članke jer su se istaknuli svojom jednostavnošću i širinom objašnjenja, te je završni program bio što sam tražio. Iako su članci dobro opisali kako implementirati neuronsku mrežu, svejedno sam naišao na probleme koje sam trebao ispraviti i popraviti da bih dobio funkcionirajući program.

Počinjemo s pripremom okruženja python-a za implementaciju, za što je potrebno koristiti upravitelj paketa *pip*. Za ovu implementaciju potrebno je instalirati sljedeće pakete:

```
!pip install keras_vggface
!pip install keras_applications
!pip install mtcnn
```

Pakete koje instaliramo prije nego možemo početi s radom aplikacije su *keras\_vggface*, implementacija Keras sučelja za prepoznavanje lica. Sljedeći paket je

*keras\_applications*, koji sadrži modul za trenirane modele neuronske mreže. Zatim je potrebno izmijeniti liniju 20 u *keras\_vggface models.py* i

```
from keras.engine.topology import get_source_inputs
```

u

```
from keras.utils.layer_utils import get_source_inputs
```

Zatim možemo instalirati zadnji paket *mtcnn*. Nakon što smo pripremili okruženje za python, možemo krenuti s pisanje programa što počinje s uvozom potrebnih modula.

```
import tensorflow as tf
import keras
import keras_vggface
from keras.utils import img_to_array
from keras_vggface.vggface import VGGFace
import mtcnn
import numpy as np
import matplotlib as mpl
import cv2
from keras_vggface.utils import preprocess_input
from keras_vggface.utils import decode_predictions
```

Prvi uvoz je Tensorflow, na čemu je baziran Keras, gdje je Keras sučelje za Tensorflow.

Sljedeći uvoz je sam Keras, čime možemo koristiti različite Keras funkcionalosti.

Nakon toga uvozimo *keras\_vggface*, što je dio Keras-a koji se koristi za prepoznavanje lica, što nama treba za ovu implementaciju.

Zatim uvozimo funkciju *img\_to\_array* iz *keras.utils*, da ju možemo koristiti za pretvaranje slike u niz.

Isto tako uvozimo funkciju *VGGFace* iz modula *keras\_vggface.vggface*, što nam omogućuje korištenje *VGGFace* arhitekture i korištenje različitih modela za prepoznavanje lica.

Sljedeći modul za uvoz je *mtcnn*, koji se koristi zbog svoje izvrsne sposobnosti detekcije lica.

Nakon toga uvozimo modul *numpy* kao *np*, koji je važan zbog svoje mogućnosti za rad s nizovima i računanja.

Predzadnji modul koji uvozimo je `matplotlib`, kojeg uvozimo kao `mpl` i koji nam omogućava kreiranje vizualizacije.

Zadnji modul koji nam treba je `cv2`, koji je verzija OpenCV modula za procesuiranje slika prilagođenija korisniku.

Na kraju uvozimo dvije funkcije iz `keras_vggface.utils`, `preprocess_input` koji procesuirala ulaze u potreban oblik; i `decode_predictions`, koji dekodira predviđanja i pogađanja modela.

Nakon što smo uključili sve potrebne module i funkcije, možemo krenuti s izradom programa. Prvo što sam napravio je kreirao instancu modela kojega ću koristiti za prepoznavanje lica, koji može biti jedan od 3 modela. Instancu sam nazvao `vggface_resnet` radi lakšeg prepoznavanja.

```
vggface_model=VGGFace(model='placeholder')
```

Nakon toga trebamo učitati sliku iz trenutnog direktorija, što radimo s linijom:

```
photo=cv2.imread('placeholder.jpg')
```

Zatim koristimo `mtcnn` detektor lica da provjerimo da li postoji lice na slici. To radimo tako da prvo kreiramo instancu `mtcnn`-a, čime dobivamo na raspolaganje funkcije toga modula. Nakon toga koristimo funkciju instance `detect_faces` koja detektira lica na slici, koju spremamo u varijabli `face_roi`.

```
face_detector=mtcnn.MTCNN()
```

```
face_roi=face_detector.detect_faces(photo)
```

Ako je program prepoznao da postoji lice na slici, onda možemo krenuti na sljedeći korak, što je obrezivanje slike, da se ukloni okolina i da nam ostane samo lice koje možemo procesuirati dalje.

```
x1, y1, width, height = face_roi[0]['box']
```

```
x2, y2, = x1+width, y1+height
```

```
lice=photo[y1:y2, x1:x2]
```

Nakon što smo dobili sliku obrezanog lica, potrebno je promijeniti sliku u veličinu koju primaju modeli, što iznosi 224x224. To radimo pomoću `cv2` modula i njegove funkcije `resize`. Time dobijemo sliku potrebne veličine, ali je slika u formatu BGR. Radi toga je potrebno pretvoriti sliku u format RGB, i spremiti pretvorenu sliku u varijablu `face`.

```
face_bgr = cv2.resize(lice, (224, 224))
```

```
face = cv2.cvtColor(face_bgr, cv2.COLOR_BGR2RGB)
```

Nakon što smo dobili sliku lica u pravilnom formatu i veličini, potrebno je pretvoriti sliku u polje da ga neuronska mreža može uspoređivati. To radimo tako da koristimo funkciju `img_to_array`

koju smo uveli na početku. Nakon toga dodajemo novu os u niz pomoću numpy funkcije *expand\_dims*, te nakon toga imamo polje lica koje možemo dati neuronskoj mreži za prepoznavanje.

```
numpty_lice=img_to_array(face)
image_batch=np.expand_dims(numpty_lice, axis=0)
```

Na kraju koristimo ResNet50 za prepoznavanje tako da pozovemo instancu modelu *vggface\_model* i koristimo njegovu funkciju *predict*, koja za ulaz prima niz u određenom formatu. Rezultate spremamo u varijablu *predictions*, ili predviđanja na hrvatskom. Prije nego možemo saznati što je neuronska mreža prepoznala, potrebno je dekodirati predviđanja, a to radimo pomoću funkcije *decode\_predictions* koju smo uveli na početku. Rezultate spremamo u varijablu *names*.

```
predictions=vggface_model.predict(image_batch)
names=decode_predictions(predictions)
```

Sada je još preostalo ispisati dekodirane rezultate, što se može raditi pomoću *print(names)*, ali radi bolje preglednosti ja sam odlučio to napraviti pomoću petlje, gdje za svako ime koje je neuronska mreža prepoznala, program ispisuje ime i razinu sigurnosti u postocima.

```
for name in names[0]:
    print('%s: %.3f%%' % (name[0], name[1]*100))
```

## 6.2. Pregled rada programa

Pošto koristimo već treniranu neuronsku mrežu, nije ju potrebno trenirati te možemo krenuti na prikaz rada aplikacije na primjeru. Prije nego možemo krenuti s pokretanjem aplikacije potrebno je dodati sliku(poželjno sliku lica) koju želimo prepoznati. Sliku spremamo u datoteke s naredbom *sample\_data*. Nakon toga u liniji koda gdje čitamo sliku promijenimo *placeholder* u ime slike koju želimo učitati. U ovom primjeru sam odabrao slike Bob Iger-a, trenutnog direktora Disney-a.

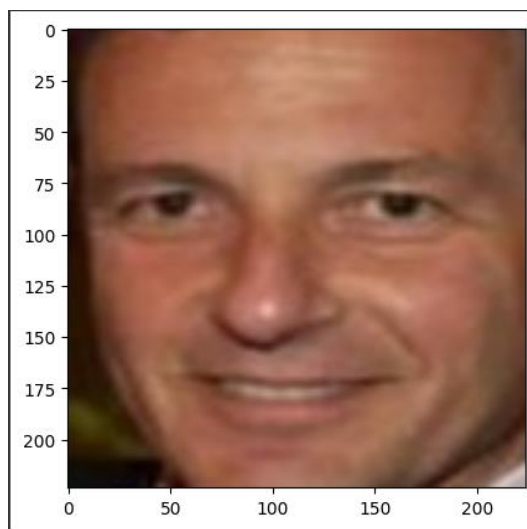
Pokretanjem programa, ako smo dodali sliku lica, program počinje ispisom detektiranja lica. Ta funkcija prolazi kroz cijelu sliku tražeći lice, te ako ga pronađe kreira pravokutnik oko lica i ispisuje lokaciju u pikselima za glavne značajke(nose, desni dio usta, desno oko, lijevo oko, lijevi dio usta). Za lakše razumijevanje dodao sam naredbu za ispis rezultata detekcije lica:

```
print(face_roi)
```

```
1/1 [=====] - 0s 99ms/step
1/1 [=====] - 0s 95ms/step
1/1 [=====] - 0s 20ms/step
1/1 [=====] - 0s 20ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 102ms/step
1/1 [=====] - 0s 112ms/step
[{'box': [88, 65, 97, 121], 'confidence': 0.9998988164527893, 'keypoints': {'left eye': (103, 112), 'right eye': (149, 111), 'nose': (121, 141), 'mouth left': (106, 157), 'mouth right': (146, 157)}}
```

Slika 7: Ispis programa kod detekcije lica

Nakon što je program pronašao lice, potrebno je obrezati i promijeniti veličinu slike i format da bude u dobrom formatu za pretvaranje u niz koji neuronska mreža prima. Nakon što je program to izvršio, možemo prikazati lice u dobrom formatu pomoću *matplotlib* modula. Rezultat toga vidljiv je na Slici 8:



Slika 8: Slika lica nakon obrade

Nakon što program ima sliku u dobrom formatu, može ju pretvoriti u niz koji neuronska mreža prima. Niz koji dobijemo pretvorbom možemo ispisati pomoću naredbe *print*.

```

[[[ 50. 21. 15.]
 [ 49. 28. 15.]
 [ 46. 19. 14.]
 ...
 [ 98. 72. 61.]
 [ 92. 67. 57.]
 [ 90. 66. 56.]]]

[[[ 48. 19. 14.]
 [ 48. 19. 14.]
 [ 45. 18. 14.]
 ...
 [ 98. 72. 61.]
 [ 93. 68. 58.]
 [ 91. 67. 57.]]]

[[[ 46. 17. 12.]
 [ 45. 16. 12.]
 [ 43. 16. 13.]
 ...
 [ 99. 72. 61.]
 [ 94. 69. 59.]
 [ 92. 68. 58.]]]

...

[[[ 13. 15. 14.]
 [ 13. 15. 14.]
 [ 13. 15. 14.]
 ...
 [181. 169. 169.]
 [181. 171. 170.]
 [181. 171. 170.]]]

[[[ 13. 15. 14.]
 [ 13. 15. 14.]
 [ 13. 15. 14.]
 ...
 [180. 170. 170.]
 [181. 171. 172.]
 [181. 172. 172.]]]

[[[ 13. 15. 14.]
 [ 13. 15. 14.]
 [ 13. 15. 14.]
 ...
 [180. 170. 171.]
 [181. 171. 172.]
 [181. 172. 173.]]]]]

```

Slika 9: Niz dobiven pretvorbom slike

Program zatim prosljeđuje taj niz neuronskoj mreži koja pokušava otkriti tko je osoba na slici, te nakon toga dekodira predviđanja. Zatim ispisuje top 5 previđanja i postotak sigurnosti za svako.

```

171 [-----]
b' Bob_Iger': 99.827%
b' Ronan_Farrow': 0.047%
b' Gerry_Scotti': 0.034%
b' Marc_Pircher': 0.008%
b' Corrado_Passera': 0.003%

```

Slika 10: Rezultati pogađanja

Vidimo da je program prepoznao lice kao osobu Bob Iger sa 99% sigurnosti, što je točan odgovor.

### 6.3. Testiranje neuronske mreže

Da bi se mreža mogla testirati, potrebne su određene modifikacije. Naime, pošto skup podataka na kojem je mreža trenirani nije više dostupan. Zbog toga je bilo potrebno podesiti neuronsku mrežu na skup podataka koji želimo testirati, u ovom slučaju LFW.

Neuronske mreže iz vggface modula su trenirane na skupu podataka VGGFace2, koji nažalost više nije dostupan za preuzimanje. Zbog nemogućnosti pristupa i pregleda podataka u tome skupu podataka, morao sam prilagoditi način testiranja modela. Testiranje je rađeno na skupu podataka LFW.

LFW skup podataka je baza slika lica sa preko 13 000 slika lica od 5749 osoba, gdje 1680 osoba imaju više od jedne slike. Svaka osoba ima imenovanu mapu sa imenovanom slikom. Skup se smatra mjerilom za verifikaciju i prepoznavanje zbog velikog skupa lica. Unatoč tome, bez obzira na performanse algoritma na skupu, ne može se zaključiti da je algoritam spreman za komercijalno korištenje. [34]

Iako ima velik broj lica i osoba, skup ima određene nedostatke, poput nedostatka manjina, žena, starijih i mlađih osoba. Time može doći do određenih sklonosti netočnog prepoznavanja ili diskriminacija, pogotovo ako se koristi za treniranje sustava. Još jedan problem LFW skupa podataka je ne-balansiran broj slika osoba. Neke osobe imaju preko 100 slika, dok većina osoba u skupu imaju po 1 sliku. [34]

Zbog toga je bilo potrebno podesiti neuronsku mrežu na skup podataka LFW da bi mogli testirati neuronsku mrežu. To se radi tako da otključamo potrebne slojeve u neuronskoj mreži za trening, te uzmemo dio LFW skupa podataka za trening. Zbog problema ne-balansiranost skupa, ja sam odlučio samo uzeti osobe koje imaju barem 20 slika čime bi omogućio neuronskoj mreži dovoljno slika za treniranje. Time sam dobio 62 osobe koje pripadaju tome kriteriju.

Ovdje je mogući velik broj varijacije sa pripremom podataka koje želimo testirati, te se može uzeti osobe sa više ili manje slika. U slučaju da želimo testirati na cijeloj bazi, potrebno bi bilo uvesti varijacije slika kod osoba sa jednom slikom, poput zrcaljenja ili dodavanjem buke (tj. uvođenjem degradacije slike). Time se stvaraju dodatne slike osobe na kojima sustav može trenirati i kasnije biti testiran.

Nakon toga se može odabrati koliko slika po osobi će se uzeti za treniranje i testiranje. Ja sam odlučio koristiti do 100 slika po osobi, tj. ako osoba ima više od 100 slika onda ću uzeti samo 100 slika. Ako osoba ima manje onda će se uzeti koliko ih ima. Ovdje se isto može balansirati skup podataka uzimanjem broja slika od odabranih osoba s najmanjim brojem slika.

Zatim se podijele odabrane slike i osobe u dio za trening i dio za testiranje. U mojem slučaju to je bilo 80% slika za trening i 20% za testiranje. Ovdje se isto može eksperimentirati sa podjelom podataka.

Pošto sam koristio već istreniranu neuronsku mrežu ResNet50, potrebno je bilo samo podesiti mrežu na skup LFW. To sam uradio tako što sam dodao klasifikacijski sloj i trenirao ga na pripremljenim podacima. Dodao sam i sloj za promjenu veličine slike na potrebnu

veličinu. Ovdje isto postoji velik broj mogućih kombinacija kako želimo složiti i podesiti neuronsku mrežu.

Nakon toga sam krenuo sa treniranjem koristeći već opisane podatke za treniranje. Odabrao sam 15 epoha za treniranje, iz najvećeg razloga da što je više epoha time se duže trenira mreža. Radi uštedi na vremenu ja sam ih ograničio na 15.

Neuronska mreža koju sam odabrao je ResNet50, treniran na skupu VGGFace2. Rezultati koje je mreža postigla nakon 15 epoha su 91.25% točnosti. Vidimo da čak i kraće vrijeme treniranja omogućuje postizanje dobrih rezultata za podešavanja već treniranih mreža. Rezultati isto uvelike ovise i o faktorima koje sam spomenuo koje možemo mijenjati, te ako bi uzeli više osoba, točnost bi bila niža, dok treniranjem mreže kroz više epoha bi mogli dobiti i veću točnost.



## 7. Zaključak

Kao i svaka nova tehnologija do sada, prepoznavanje lica ima veliki potencijal za olakšavanje svakodnevnog života, isto tako nosi veliki potencijal za napad na privatnost. Važno je pristupiti implementaciji takve tehnologije sa oprežnošću i pažnjom da se zaštite krajnji korisnici. U ovom radu sam prošao kroz osnove prepoznavanja lica, od kako rade do njihovih prednosti i nedostataka. Isto tako je opisano kako se mogu koristiti neuronske mreže za obavljanje takvog zadatka, te koji su neki od mogućih načina implementacije takve tehnologije za svakodnevnog čovjeka.

Implementacijom neuronske mreže sam shvatio jednostavnost korištenja i pristupa toj tehnologiji. U današnje doba se vrlo brzo može naći vodič za implementaciju i korištenje te tehnologije, što može predstavljati veliki rizik zbog nedostatka kontrola tko ima pristup tehnologiji. S druge strane, tehnologija još sadrži dosta nedostataka prije nego smo došli do točke bez povratka kod gubitka privatnosti. Neuronske mreže još uvijek treba trenirati na dovoljno velikom skupu podataka te sustav može predviđati samo podatke koje je već sreo.

## Popis literature

- [1] Electronic IDentification (28.7.2022.) *Facial Recognition: how it works and its safety*. Preuzeto dana 12.7.2023. s <https://www.electronicid.eu/en/blog/post/face-recognition/en>
- [2] Amazon (bez dat.) *What Is Facial Recognition?* Preuzeto dana 12.7.2023. s <https://aws.amazon.com/what-is/facial-recognition/>
- [3] NEC New Zealand (23.1.2022.) *5 most common uses of facial recognition* preuzeto dana 12.7.2023. s <https://www.nec.co.nz/market-leadership/publications-media/5-most-common-uses-of-facial-recognition/>
- [4] Introna, L., & Nissenbaum, H. (2010). Facial recognition technology a survey of policy and implementation issues.
- [5] David Gargaro (2023). *The pros and cons of facial recognition technology* preuzeto dana 13.7.2023. s <https://www.itpro.com/security/privacy/356882/the-pros-and-cons-of-facial-recognition-technology>
- [6] Amazon (bez dat.) *What Is A Neural Network?* preuzeto dana 15.7.2023. s <https://aws.amazon.com/what-is/neural-network/>
- [7] Great Learning Team (2022.) *Types of Neural Networks and Definition of Neural Network* preuzeto dana 15.7.2023. s <https://www.mygreatlearning.com/blog/types-of-neural-networks>
- [8] Jason Brownlee (2019.) *A Gentle Introduction to Deep Learning for Face Recognition* preuzeto dana 16.7.2023. s <https://machinelearningmastery.com/introduction-to-deep-learning-for-face-recognition/>
- [9] Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1701-1708).
- [10] Shahsavarani, S., Analoui, M., & Ghiass, R. S. (2020). M<sup>2</sup> Deep-ID: A Novel Model for Multi-View Face Identification Using Convolutional Deep Neural Networks. *arXiv preprint arXiv:2001.07871*.
- [11] Wong, S. Y., Yap, K. S., Zhai, Q., & Li, X. (2019). Realization of a hybrid locally connected extreme learning machine with DeepID for face verification. *IEEE Access*, 7, 70447-70460.
- [12] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

- [13] Luka Dulčić (2020.) *Face Recognition with FaceNet and MTCNN* preuzeto dana 17.7.2023. s <https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/>
- [14] Kasar, M. M., Bhattacharyya, D., & Kim, T. H. (2016). Face recognition using neural network: a review. *International Journal of Security and Its Applications*, 10(3), 81-100.
- [15] Alblushi, A. (2021). Face recognition based on artificial neural network: A review. *Artificial Intelligence & Robotics Development Journal*, 116-131.
- [16] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- [17] Almabdy, S., & Elrefaei, L. (2019). Deep convolutional neural network-based approaches for face recognition. *Applied Sciences*, 9(20), 4397.
- [18] Korkmaz, M., & Yilmaz, N. (2016). Face recognition by using back propagation artificial neural network and windowing method. *Journal of Image and Graphics*, 4(1), 15-19.
- [19] Bhattacharjee, D., Basu, D. K., Nasipuri, M., & Kundu, M. (2010). Human face recognition using fuzzy multilayer perceptron. *Soft Computing*, 14, 559-570.
- [20] Yan, K., Huang, S., Song, Y., Liu, W., & Fan, N. (2017, July). Face recognition based on convolution neural network. In *2017 36th Chinese Control Conference (CCC)* (pp. 4077-4081). IEEE.
- [21] Keras (bez dat.) *About Keras* preuzeto dana 30.7.2023 s <https://keras.io/about/>
- [22] Karthik Shiraly (2022.) *How to Implement Tensorflow Facial Recognition From Scratch* preuzeto dana 30.7.2023. s <https://www.width.ai/post/tensorflow-facial-recognition>
- [23] TutorialsPoint (bez dat.) *Real Time Prediction using ResNet Model* preuzeto dana 30.7.2023. s [https://www.tutorialspoint.com/keras/keras\\_real\\_time\\_prediction\\_using\\_resnet\\_model.htm](https://www.tutorialspoint.com/keras/keras_real_time_prediction_using_resnet_model.htm)
- [24] Jason Brownlee (2019) *A Gentle Introduction to Pooling Layers for Convolutional Neural Networks* preuzeto dana 6.8.2023. s <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>
- [25] Sagar Sharma (2017) *Action Functions in Neural Networks* preuzeto dana 15.8.2023. s <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- [26] Anwarul, S., & Dahiya, S. (2020). A comprehensive review on face recognition methods and factors affecting facial recognition accuracy. *Proceedings of ICRIC 2019: Recent Innovations in Computing*, 495-514.

- [27] Cyberlink(2023) *Top 7 Use Cases for Facial Recognition in 2023* preuzeto dana 20.8.2023 s <https://www.cyberlink.com/faceme/insights/articles/228/how-to-use-facial-recognition>
- [28] Opendgenus(bez dat.) *Understanding ResNet50 architecture* preuzeto dana 21.8.2023 s <https://iq.opengenius.org/resnet50-architecture/>
- [29] Aditi Rastogi(2022) *ResNet50* preuzeto dana 21.8.2023 s <https://blog.devgenius.io/resnet50-6b42934db431>
- [30] Tech Hive(2020) *Review of ResNet Family: From ResNet to ResNeSt* preuzeto dana 21.8.2023. s <https://sheng-fang.github.io/2020-05-20-review-resnet-family/>
- [31] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132-7141).
- [32] Divyansh Dwivedi (2018) *Face Detection For Beginners* preuzeto dana 27.8.2023 s <https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9>
- [33] Rachel Metz (2021) *Facebook is shutting down its facial recognition software* preuzeto dana 27.8.2023 s <https://edition.cnn.com/2021/11/02/tech/facebook-shuts-down-facial-recognition/index.html>
- [34] University of Massachusetts (bez dat.) *Labeled Faces in the Wild* preuzeto dana 28.8.2023. s <http://vis-www.cs.umass.edu/lfw/>
- [35] Divyansh Dwivedi (2018) *Face recognition for beginners* preuzeto dana 6.9.2023. s <https://towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2>

## Popis slika

Slika 1: Primjer snapchat filtera .....	6
Slika 2: Duboka neuronska mreža s više slojeva .....	12
Slika 3: Konvolucijska neuronska mreža .....	13
Slika 4: Rezultati CNN mreže AlexNet s potpornim vektorskim strojem za klasifikaciju .....	17
Slika 5: Pregled strukture ResNet-50 i SE-ResNet-50 .....	23
Slika 6: Usporedba običnog višeslojnog perceptrona i neizrazitog perceptrona .....	25
Slika 7: Ispis programa kod detekcije lica .....	30
Slika 8: Slika lica nakon obrade .....	30
Slika 9: Niz dobiven pretvorbom slike .....	31
Slika 10: Rezultati pogađanja .....	31

## Popis tablica

Tablica 1: Rezultati BPNN neuronske mreže i metode prozora kroz iteracije .....	24
---	----