

Prepoznavanje lica koristeći duboke neuronske mreže

Benček, Tena

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:255046>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-30**

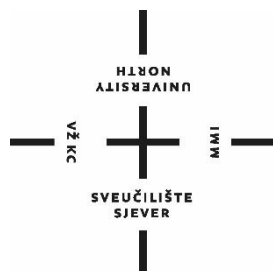


Repository / Repozitorij:

[University North Digital Repository](#)



**SVEUČILIŠTE SJEVER
SVEUČILIŠNI CENTAR VARAŽDIN**



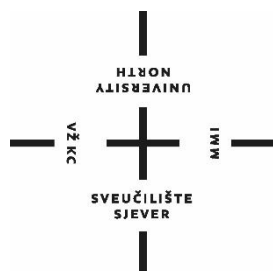
DIPLOMSKI RAD

**PREPOZNAVANJE LICA KORISTEĆI DUBOKE
NEURONSKE MREŽE**

Tena Benček

Varaždin, rujan 2022.

SVEUČILIŠTE SJEVER
SVEUČILIŠNI CENTAR VARAŽDIN
Studij Multimedija



DIPLOMSKI RAD br. 077-MMD-2022

**PREPOZNAVANJE LICA KORISTEĆI DUBOKE
NEURONSKE MREŽE**

Student:
Tena Benček, 0016134018

Mentor:
izv. prof. dr. sc. Emil Dumić

Varaždin, rujan 2022.

Prijava diplomskog rada

Definiranje teme diplomskog rada i povjerenstva

| | | | |
|-----------------------------|---|---------|-----------------|
| ODJEL | Odjel za multimediju | | |
| STUDIJ | diplomski sveučilišni studij Multimedija | | |
| PRISTUPNIK | Benček Tena | JMBAG | 0016134018 |
| DATUM | 14.9.2022. | KOLEGIJ | Računalni vid |
| NASLOV RADA | Prepoznavanje lica koristeći duboke neuronske mreže | | |
| NASLOV RADA NA ENGL. JEZIKU | Face recognition using deep neural networks | | |
| MENTOR | Emil Dumić | ZVANJE | izv.prof.dr.sc. |
| ČLANOVI POVJERENSTVA | 1. doc. art. dr. sc. Mario Periša - predsjednik | | |
| | 2. doc. art. dr. sc. Robert Geček - član | | |
| | 3. izv. prof. dr. sc. Emil Dumić - mentor | | |
| | 4. doc. dr. sc. Andrija Bernik - zamjenski član | | |
| | 5. _____ | | |

Zadatak diplomskog rada

| | |
|------|--|
| BROJ | 077-MMD-2022 |
| OPIS | <p>U ovom radu će biti napravljen sustav za detekciju lica te potom prepoznavanje lica, koristeći različite metode dubokog učenja.</p> <p>Opisat će se različiti algoritmi za detekciju, poravnavanje i prepoznavanje lica. Algoritmi za detekciju lica obuhvaćaju Haarove kaskade (Viola-Jones), histogram orijentiranih gradijenata (HOG) i metodu potpornih vektora (SVM), te metode dubokog učenja (npr. DDFD, MTCNN). Samostalno poravnavanje lica opisat će se pomoću stabala odlučivanja (ERT). Potom će se opisati raniji algoritmi za prepoznavanje lica: Fisherfaces, eigenfaces i histogrami lokalnih binarnih uzoraka (LBPH). Nakon toga će se opisati noviji algoritmi za prepoznavanje lica koristeći duboke neuronske mreže poput FaceNet, DeepFace, DeepID i VGGFace. Također će se opisati neke neuronske mreže za klasifikaciju, koje se mogu koristiti i za prepoznavanje lica: SqueezeNet i GoogLeNet. Praktični dio rada bit će ispitivanje različitih algoritama za detekciju i prepoznavanje lica, te će se usporediti točnost algoritama za prepoznavanje lica. Bit će korištena neka od postojećih baza lica ili će biti napravljena vlastita baza.</p> |

| | | | |
|----------------|------------|----------------|-------------------|
| ZADATAK URUČEN | 14.9.2022. | POTPIS MENTORA | <i>Emil Dumić</i> |
|----------------|------------|----------------|-------------------|

Sažetak

Svrha ovog rada je upoznati se s pojmom i konceptom biometrijske metode prepoznavanja lica koristeći duboke neuronske mreže. U radu je opisana biometrijska metoda prepoznavanja lica kao i njene karakteristike i povijest. Osim prepoznavanja lica pojašnjen je i pojam detekcije lica i algoritmi koji se koriste prilikom detekcije lica. Objasnjeni su stariji algoritmi za prepoznavanje lica poput Eigenface, Fisherface, histograma lokalnih binarnih uzoraka. Definiran je pojam dubokih neuronskih mreža kao i algoritmi za prepoznavanje lica koji se koriste dubokim neuronskim mrežama. Teorijski zaključci u praktičnom su dijelu podijeljeni usporedbom dva algoritma za prepoznavanje lica koji koriste duboke neuronske mreže, GoogLeNet i SqueezeNet.

Ključne riječi: biometrija; prepoznavanje lica; algoritmi za prepoznavanje lica; duboke neuronske mreže;

Abstract

The purpose of this paper is to get acquainted with the term and concept of the biometric face recognition method using deep neural networks. The paper describes the biometric face recognition method as well as its characteristics and history. In addition to face recognition, the concept of face detection and the algorithms used in face detection are also explained. Older face recognition algorithms like Eigenface, Fisherface, histogram of local binary patterns are explained. The concept of deep neural networks is defined as well as face recognition algorithms that use deep neural networks. Theoretical conclusions are divided into a practical part by comparing two face recognition algorithms that use deep neural networks, GoogLeNet and SqueezeNet.

Key words: biometrics; facial recognition; facial recognition algorithms; deep neural networks;

Sadržaj

| | |
|---|----|
| 1. Uvod..... | 1 |
| 2. Prepoznavanje lica..... | 2 |
| 2.1. Definirane prepoznavanja lica | 2 |
| 2.2. Primjena..... | 3 |
| 2.3. Izazovi | 5 |
| 3. Detekcija lica..... | 7 |
| 4. Algoritmi za detekciju lica | 8 |
| 4.1. Haar značajke (Viola Jones algoritam)..... | 8 |
| 4.2. Histogram orijentiranih algoritama | 10 |
| 4.3. Metode dubokog učenja..... | 11 |
| 4.4. Metoda potpornih vektora | 13 |
| 5. Raniji algoritmi za prepoznavanje lica..... | 14 |
| 5.1. Fisherface | 14 |
| 5.2. Eigenface | 16 |
| 5.3. Histogrami lokalnih binarnih uzoraka | 17 |
| 5.4. SIFT algoritam..... | 19 |
| 5.5. SURF algoritam..... | 20 |
| 6. Duboke neuronske mreže | 22 |
| 6.1. Definiranje dubokih neuronskih mreža | 22 |
| 6.2. Povijest | 23 |
| 6.3. Arhitektura..... | 25 |
| 7. Algoritmi za prepoznavanje lica koristeći duboke neuronske mreže | 26 |
| 7.1. DeepFace | 26 |
| 7.2. FaceNet..... | 28 |
| 7.3. DeepID | 30 |
| 7.4. GoogLeNet | 32 |
| 7.5. SqueezeNet..... | 33 |
| 7.6. VGGFace..... | 34 |
| 8. Praktični dio | 37 |
| 8.1. Priprema za testiranje | 37 |
| 8.2. SqueezeNet..... | 39 |
| 8.3. GoogLeNet | 42 |
| 8.4. Usporedba rezultata | 46 |
| 9. Zaključak..... | 49 |
| Popis literature..... | 50 |
| Popis slika | 54 |

1. Uvod

Posljednjih desetljeća razvile su se brojne biometrijske metode koje se koriste u svrhu prepoznavanja osobe. Te biometrijske metode ispituju fiziološke i bihevioralne karakteristike pojedinca kako bi se utvrdio i potvrdio njihov identitet. Jedna od takvih metoda je prepoznavanje lica.

Prepoznavanje lica aktivnost je koju ljudska vrsta obavlja svakog dana od početka vremena te je bilo neizbježno da se i tehnologija umiješa u tu aktivnost. Ova biometrijska metoda jedna je od najmanje nametljivih i najbržih biometrijskih metoda te se za nju smatra da je prirodni i prijateljski sustav za provjeru identiteta.

Cilj ovog rada je predstaviti biometrijsku metodu prepoznavanja lica s naglaskom na duboke neuronske mreže, te usporediti algoritme za prepoznavanje lica koji koriste te duboke neuronske mreže

Ovaj rad razdijeljen je na dva dijela, praktični i teorijski. Teorijski dio započinjemo s pojašnjenjem općenitih pojmova i značajki prepoznavanja i detekcije lica. Nakon toga objašnjavaju se algoritmi za detekciju lica: Viola-Jones algoritam, histogram orijentiranih gradijenata, metode dubokog učenja DDFD i MTCNN, te metoda potpunih vektora. Nakon toga prelazimo na definiranje ranijih algoritama za prepoznavanje lica, točnije priča se o Fisherface i Eigenface algoritmima, histogrami lokalnih binarnih koeficijenata, SIFT i SURF algoritmu. Konačno na red dolazi definiranje pojma i arhitekture dubokih neuronskih mreža kao i algoritama za prepoznavanje lica koji koriste duboke neuronske mreže, a to su: DeepFace, FaceNet, GoogLeNet, SqueezeNet, DeepID i VGGFace. U praktičnom dijelu bit će uspoređena dva algoritma prilagođena za prepoznavanje lica, GoogLeNet i SqueezeNet.

2. Prepoznavanje lica

2.1. Definirane prepoznavanja lica

Prepoznavanje lica jedna je od osnovnih vještina ljudske interakcije. To je zadatak koji ljudi obavljaju rutinski, gotovo mehanički u svom svakodnevnom životu već tisućama godina, te je bilo samo pitanje vremena kada će tehnologija uspješno izvršavati i taj zadatak.

Prepoznavanje lica proučava se već gotovo 50 godina zbog brojnih praktičnih primjena u području biometrije, informacijske sigurnosti i nadzornog sustava, kontrole pristupa te mnogih drugih. Kada ju uspoređujemo s drugim biometrijskim metodama, poput otiska prsta ili detekcije oka, ono je zasigurno najistaknutija metoda za autentikaciju identiteta jer je u usporedbi s drugim metodama najmanje nametljiva, najbrža te zbog svog pristupa u stvarnom vremenu i nenametljivosti ne daje ljudima dojam kao da su nužno pod nadzorom ili da se krši njihova privatnost. Zapravo može se reći da je prepoznavanje lica pasivan, nenametljiv sustav za provjeru osobnog identiteta na "prirodan" i prijateljski način [1].

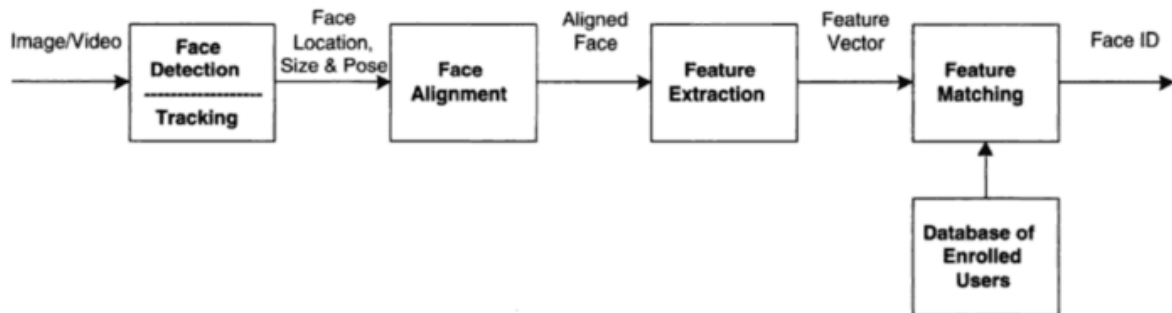
Općenito svaki biometrijski sustav za identifikaciju osobe koristi se fiziološkim karakteristikama, poput otiska prsta, lica i šarenice oke, ili obrascima ponašanja, kao što su rukopis, glas ili uzorak pritiska na tipke. Usporedno na to prepoznavanje lica definiramo kao biometrijski pristup koji koristi automatizirane metode za provjeru ili prepoznavanje identiteta žive osobe na temelju njezinih fizioloških karakteristika [2].

U svom članku [3], opširnije objašnjava biometrijski pojam prepoznavanja lica kao način identifikacije ljudskog lica primjenom modernih tehnologija koristeći biometriju kako bi se mapirale značajke lica sa slike ili video zapisa i usporedile da li postoji podudaranje sa već poznatim licima pohranjenim u bazama podataka.

Prepoznavanje lica sa slika ili videozapisa može se raditi na jedan, ili oba, od dva moguća načina: provjera lica (eng. *face verification*) ili identifikacija lica (eng. *face identification*). Provjera lica uključuje podudaranje jedan-na-jedan i ono uspoređuje sliku lica s predloškom slike lica čiji se identitet polaže, dok identifikacija lica uključuje podudaranja jedan prema više, odnosno ono uspoređuje sliku lica sa svim slikama unutar neke baze podataka ne bi li se identificirao identitet lica upita [4].

Proces prepoznavanja lica uključuje četiri osnovna koraka (Slika 2.1.):

1. Detekcija lica
2. Poravnanje lica
3. Ekstrakcija obilježja
4. Podudaranje lica



Slika 2.1. Koraci prilikom prepoznavanja lica

2.2. Primjena

Napomenuto je kako je prepoznavanje lica najpasivnija i najmanje nametljiva biometrijska metoda no to ne znači da dolazi bez kritičizma. Mnogi ju kritiziraju kao neetičko oružje kojim se špijunira javnost i narušava privatnost. Međutim, razlog za takve kritike je uglavnom nedostatak informacija. Prepoznavanje lica uglavnom se koristi proporcionalno i odgovorno. Ono ima brojne primjene koje nam svakodnevno olakšavaju život, od sigurnosti, nadzora, kontrole pristupa i drugih. Više o područjima primjene biometrijske metode prepoznavanja lica slijedi u nastavku.

Vjerojatno najkorištenije područje primjene prepoznavanja lica je sigurnost. Prepoznavanje lica igra ključnu ulogu u kaznenim istragama. Naime sustavi za prepoznavanja lica mogu identificirati lica traženih kriminalaca i osumnjičenih za kaznena djela sa slika ili videozapisa iz nadzornih sustava. Pristup u stvarnom vremenu ove tehnologije još je jedna njena prednost u polju sigurnosti. Mogućnost provjere u stvarnom vremenu da li je osoba na popisu kriminalista dodana je prednost u odnosu na proces donošenja odluka prije zaustavljanja i traženja osumnjičenih, čime se smanjuje i vjerojatnost diskriminacija. Također ono pomaže detektirati i eliminirati oružje te na kraju spriječiti kazneno djelo prije nego što se imalo priliku i dogoditi. [5]

Primjena prepoznavanja lica uobičajena je aktivnost u mnogim zračnim lukama. Sve je veća implementacija takozvanih biometrijskih putovnica koje značajno smanjuju vrijeme čekanja na kontrolu putovnica, ubrzavaju dolazak do terminala te u suštini povećavaju sigurnost u zračnim lukama jer identificiraju pojedince brže i točnije od ljudskog oka. [5]

Primjena prepoznavanja lica u području financija i bankarstva uglavnom je iskazana kroz aspekt sigurnosne prevencije i kontrole pristupa. Zaustavljanje pljačkaša banaka prvo je što padne na pamet kada se pomisli na sigurnost u bankarskoj industriji i iako je istina da sustavi za prepoznavanje lica mogu identificirati čak i osobu s maskom za lice te joj pronaći identitet putem podataka u sustavu javne sigurnosti kako bi se pribavili dokazi i uhvatili kriminalce, to nije jedina primjena sustava prepoznavanja lica u ovoj industriji [5]. Naime, ovakvi sustavi koriste se i za kontrolu pristupa određenih ljudi na razna područja unutar bankarskog objekta. Uz to banke sve više koriste prepoznavanje lica za provjeru identiteta klijenta prilikom prijave u aplikacije za internetsko i mobilno bankarstvo [6]. Dokaz tome je i podatak da od otprilike 11.000 financijskih institucija u Sjedinjenim Državama, 20% koristi biometrijski sustav prepoznavanja lica u kombinaciji s provjerom dokumenata za provjeru autentičnosti korisnika [7].

Sustav prepoznavanja lica ima i veliki potencijal za korištenje u obrazovnom sustavu. Iako još nije široko rasprostranjeno, već sada se u mnogim obrazovnim institucija ovaj sustav koristi se za detekciju prisutnosti učenika i studenata što pomaže učiteljima bolje pratiti učenike [6].

Iako vidimo da postoje mnoga područja primjene sustava prepoznavanja lica realnost je da zapravo još uvijek nismo ni zagrebli površinu kada je u pitanju istinski potencijal takvih sustava te na nama ostaje da s razvojem novih tehnologija doprinosimo i danjem razvoju potencijala ove biometrijske metode.

2.3. Izazovi

Postoji mnogo izazova koji značajno utječu na uspješnost prepoznavanje lica i podudaranje rezultata. Naime prepoznavanje lica često se susreće se s izazovima poput osvjetljenja, poze, izraza lica, šminke, plastičnih operacija, dlaka na licu, starenja i drugih.

Prilikom razvoja učinkovitog sustava za prepoznavanje lica glavni izazov predstavlja osvjetljenje radi kojeg lice može izgledati drastično drugačije. Naime, kada se slika formira na finalni rezultat utječu čimbenici poput osvjetljenja, intenziteta, spektra, ali i karakteristika kamere. Kako bi se taj izazov riješio koristi se tehnika ponovnog osvjetljenja za generiranje više frontalnih slika iste osobe s promjenjivom rasvjetom [1].

Varijacije u pozi uzrokuju značajne probleme u otkrivanju lica. Sustavi prepoznavanja lica toleriraju male varijacije u kutu lica no budući da se većina baza podataka sastoji od frontalnih slika lica prilikom prepoznavanja lica sa slika gdje su u pitanju velike varijacije u kutu rotacije za neke sustave je potrebna korekcija poze rotiranjem lica ili ravnanja s osi slike [8].

Izraz i stil lica još su jedan faktor koji uvelike utječe na prepoznavanje lica. Šminka, plastične operacije, brada i brkovi mogu promijeniti izgled i crte lica. Čak i frizura može utjecati na točnost rezultata jer može promijeniti izgled slike lica ili sakriti crte lica [1]. Neke varijacije mogu biti uzrokovane razlikom u izrazu lica. Iz tog razloga važno je prepoznati različite izraze lica za procjenu emocionalnog stanja osobe kao što su gađenje, ljutnja, sreća, straha, tuga i drugi. No postoji rješenje za to jer se izazov prepoznavanja različitih izraza lica može izračunati uz pomoć gustog optičkog toka [8].

Sljedeći čimbenik kojeg je vrijedno spomenuti je starenje. Naime starenje i bore mogu ozbiljno utjecati na uspješnost metode prepoznavanja lica. Učinak starenja u realnosti se zapravo i ne uzima previše u obzir jer zbog nepostojanja baza podataka koje sadrže slike pojedinca različite dobi ovaj učinak još nije pomno istražen [1].

Ne toliko čest, ali još uvijek vrijedan spomena, je izazov prepoznavanja lica kod sličnih osoba. Ovaj problem najistaknutiji je u slučaju blizanaca, posebice onih identičnih, jer sličnost oblika i crta lica predstavlja težak zadatak sustavu za prepoznavanje jer može uzrokovati povećanje stope lažnog prepoznavanja [8].

Posljednji izazov u prepoznavanju lica kojeg ćemo spomenuti je rezolucija slike. Naime rezolucija slike je izravno proporcionalna učinkovitosti prepoznavanja lica pa ako je rezolucija

promatrane slike dobra, tada će proces prepoznavanja biti lak i učinkovit, no u slučajevima kada imamo slike veoma niske rezolucije doći će do problema prilikom prepoznavanja [8].

Iz ovih informacija zaključujemo kako stanje slike, i izgled osobe na slici, značajno utječe na mogućnost sustava da identificira osobu sa slike i poveže ju s identitetom osobe unutar baze podataka. No već sada su se razvila rješenja za gotovo svakog od navedenih izazova te neće trebati dugo prije nego što se implementiraju rješenja i za ostatak ovih problema. Unatoč svim izazovima prepoznavanje lica još uvijek ostaje najistaknutija metoda za autentikaciju identiteta.

3. Detekcija lica

Prilikom definiranja prepoznavanja lica spomenuto je kako se ono sadrži od četiri osnovna koraka:

1. Detekcija lica
2. Poravnanje lica
3. Ekstrakcija obilježja
4. Podudaranje lica

Prvi od tih četiri koraka prilikom procesa prepoznavanja lica je detekcija lica koja je i ujedno fokus ovog poglavlja. Detekcija lica od nužne je važnosti jer lice ne može biti prepoznato ako prvo nije detektirano.

Pouzdanost detekcije lica ima veliki utjecaj na cijeli sustav prepoznavanja lica. Detektor lica trebao bi bili su stanju identificirati i locirati sva prisutna lica na nekoj slici ili videu bez obzira na njihov položaj, orijentaciju, dob, izraz ili šminku, te bi trebao biti neovisan o vanjskim uvjetima osvjetljenja i sadržaju slike, odnosno video zapisa [4].

Detekcija lica može se izvesti na temelju boje kože, pokreta, oblika lica/glave, izgleda lica ili kombinacije svih navedenih parametara. Najuspješniji algoritmi za detekciju lica bazirani su na izgledu (engl. *appearance-based methods*) bez korištenja drugih znakova [4].

Kod metoda temeljenih na izgledu detekcija lica tretira se kao problem klasificiranja skeniranog potprozora u jednu od dvije klase, lice ili ne lica. Klasiifikator lica/ne lica može se naučiti iz skupa za vježbanje koji se sastoji od primjera lica u raznim uvjetima. Izgradnja klasifikatora lica je moguća jer su pikseli na licu visoko korelirani, dok oni u pod prozoru bez lica predstavljaju mnogo manje pravilnosti. Obrada detekcija lica na temelju metoda baziranih na izgledu vrši se na sljedeći način: ulazna se slika skenira na svim mogućim mjestima i mjerilima pomoću podprozora i tada se uzorak u pod prozoru detektira kao lice ili ne lice. Međutim, velike varijacije uzrokovane promjenama u izgledu lica, osvjetljenju, izrazu i poziciji glave čine granice lica/ne lica vrlo složenim te je zato iznimno važna pouzdanost algoritma koji obavlja detekciju lica [4].

4. Algoritmi za detekciju lica

U nastavku se opisuju neki od najpopularnijih algoritama, odnosno metoda, za detekciju lica. Konkretno metode o kojima se priča su sljedeće:

1. Haarove kaskade (Viola-Jones algoritam)
2. Histogram orijentiranih gradijenata (HOG)
3. Metode dubokog učenja DDFD i MTCNN
4. Metoda potpornih vektora (SVM)

4.1. Haar značajke (Viola Jones algoritam)

Jedan od algoritama koji je doveo veliki napredak u detekciji lica je Viola-Jones algoritam kojeg su 2001. godine razvili američki znanstvenici Paul Viola i Michael Jones.

Posebnost ovog algoritma leži u tome da osim detekcija lica, što mu je primarna funkcija, on može detektirati i druge objekte na slici poput tijela čovjeka, životinja, automobila, prometnih znakova i drugo, zapravo Viola Jones algoritam može detektirati bilo što pod uvjetom da ga se tome nauči (Hrga, 2018) [9]. Ovaj algoritam posjeduje sposobnost vrlo brze detekcije lica, te radeći sa slikama od 384x288 piksela lica se prepoznaju u 15 slika u sekundi na procesoru Pentium 3, 700 MHz. Tako velika brzina kadrova postiže se jer algoritam radi isključivo s podacima koji su dostupni unutar jedne slike svjetline (engl. *Gray scale*). [10]

Viola-Jones algoritam za detekciju lica koristi strojno učenje kako bi aplikaciji dao do znanja da li slika iz baze podataka sadrži traženo lice. Zatim se kreira podprozor za pretraživanje značajki objekta dimenzije 24x24 piksela gdje se prolazi kroz cijelu sliku na način da taj podprozor pomičemo piksel po piksel preko redova i stupaca sve dok ne pokrijemo cijelu sliku i tada taj postupak ponavljamo sa slikom smanjenih dimenzija onoliko puta koliko je potrebno da se slika smanji na dimenziju manju od 24x24 piksela [9].

Kako bi bio što brži Viola-Jones algoritam prilikom detekcije lica ne koristi se pikselima već jednostavnim značajkama koje se nazivaju Haar kaskade ili Haar značajke. Haar kaskade predstavljaju sumu piksela u područjima unutar pravokutnika koji se promatra te zapravo postoje tri različite vrste ovih značajki. Prva vrsta su značajke dva pravokutnika (engl. *Two-rectangle feature*) koje predstavljaju razliku između sume piksela unutar dva pravokutnika koji

imaju istu veličinu i oblik te koji su horizontalno ili vertikalno povezni. Zatim imamo značajke tri pravokutnika (engl. *Three-rectangle feature*) koje računaju sumu unutar dva vanjska pravokutnika oduzeta od zbroja u središnjem pravokutniku. Naposljetku imamo i treću vrstu Haarovih značajki a to su značajke četiri pravokutnika (engl. *Four-rectangle feature*) koje izračunavaju razliku između para dijagonalnih pravokutnika [10].

Iako su Haar značajke daleko najpopularnije svojstvo ovog algoritam Viola-Jones algoritam zapravo obuhvaća četiri glavna svojstva tako da osim Haar značajki važno je još spomenuti i integralne slike, AdaBoost algoritam i kaskadne klasifikatore.

Integralna slika je srednji prikaz slike koji se koristi za brzo izračunavanje značajki pravokutnika. Naime, integralna slika može se izračunati u samo jednom prolazu preko originalne slike koristeći formule 4.1.1 i 4.1.2:

$$s(x, y) = s(x, y-1) + i(x, y) \quad (4.1.1)$$

$$ii(x, y) = ii(x-1, y) + s(x, y) \quad (4.1.2)$$

Unutar navedenih formula $ii(x,y)$ predstavlja osvjetljenje piksela na poziciji x,y , dok $s(x,y)$ predstavlja sumu osvjetljenja piksela u jednom retku do pozicije x , te konačno $ii(x,y)$ predstavlja sumu svih piksela lijevo i iznad pozicije x,y [10].

AdaBoost, skraćeno za *Adaptive Boosting*, je algoritam učenja koji se u početku koristio za povećanje učinkovitosti klasifikacije jednostavnog algoritma učenja na način da kombinira zbirke klasifikacijskih funkcija kako bi se formirao jači klasifikator, no sada se koristi i kako bi se odabrale značajke te istrenirali klasifikatori [10].

Unutar bilo kojeg podprozora slike ukupan broj Haarovih značajki veoma je velik, daleko veći od broja piksela (oko 160 000 za podprozor veličine 24x24 piksela). Kako bi se osigurala brza klasifikacija, proces učenja mora isključiti veliku većinu dostupnih značajki i usredotočiti se na mali skup značajki. Upravo iz tog razloga korištenjem AdaBoost algoritam razvila se metoda za konstruiranje klasifikatora odabirom malog broja važnih značajki. Taj odabir značajki postignut je jednostavnom modifikacijom AdaBoost algoritma, naime „slab klasifikator“ je ograničen tako da svaki vraćeni slabi klasifikator može ovisiti samo o jednoj osobini. Kao rezultat toga svaka faza procesa povećanja, koja bira novi slabi

klasifikator, može se promatrati kao proces odabira značajke koje kombinirano tvore jaki klasifikator [10].

Naposljetku posljednje svojstvo Viola-Jones algoritma na koje ćemo se fokusirati su kaskadni klasifikatori. Kaskadni klasifikatori postižu povećanu učinkovitost prilikom prepoznavanja lica uz radikalno smanjenje vremena izračuna te se oni sastoje od faza za trening koristeći AdaBoost algoritam. Posao svake faza je odrediti je li neki podprozor sigurno lice, da li je on možda ili ako ipak možda nije lice. Kada se podprozor klasificira da ne sadrži lice odmah se odbacuje. Suprotno tome, podprozor klasificiran kao možda lice prolazi na sljedeću fazu detekcija u kaskadi. [10]

No Viola-Jones algoritam ima više od 20 godina i iako je 2001. bio najmoderniji algoritam za detekciju lica sada on više nije jedina opcija.

4.2. Histogram orijentiranih algoritama

Histogram orijentiranih gradijenata već je star više od desetljeća no i dan danas se uvelike koristi i to s fantastičnim rezultatima. 2005. godine predstavili su da Dalal i Triggs u svom radu „*Histogram of Oriented Gradients for Human Detection*“ [11].

Histogram orijentiranih gradijenata, također poznat po svoj skraćenici kao HOG, deskriptor je značajki koji se koristi u svrhu detekcije lica ili objekata i on se temelji na ocjenjivanju normaliziranih lokalnih histograma orijentacija gradijenta slike u gustoj mreži.

Osnovna ideja iza ovog algoritma za detekciju lica je da se oblik objekta može okarakterizirati raspodjelom lokalnih gradijenta intenziteta ili smjerova rubova na male dijelove koji se nazivaju ćelije i da se za svaku ćeliju se akumulira lokalni histogram smjerova gradijenta ili orijentacije rubova preko piksela ćelije. Za bolju nepromjenjivost na osvjetljenje, sjenčanje i druge faktore koristi se kontrast prije upotrebe algoritma. To se postiže akumuliranjem lokalnog histograma na veća prostorna područja, tzv. blokove, koje tada nazivamo histogramom orijentiranog gradijenta [11].

Invarijantnost mjerila i rotacije postiže se izdvajanjem deskriptora iz ključnih točaka u prostoru mjerila slike. To se sastoji od tri koraka:

1. Detekcija ekstreme skale-prostor
2. Dodjela orijentacije

3. Ekstrakcija deskriptora

Prvi korak je namijenjen postizanju nepromjenjivosti mjerila, dok drugi korak pronalazi dominantnu orijentaciju gradijenta i naposljetku se orijentacijska brojanja vrše u odnosu na dominantni smjer [12].

Prednost histograma orijentiranih objekta je ta da oni rade s lokalnim oblicima tako da hvataju strukturu rubova s kontroliranim stupnjem nepromjenjivosti na lokalne geometrijske i fotometrijske transformacije, odnosno ako su translacije ili rotacije mnogo manje od lokalne prostorne ili orijentacijske veličine, malo se razlikuju [13].

4.3. Metode dubokog učenja

Tokom godina razvile su se brojne metode dubokog učenja za detekciju lica, no vjerojatno najpopularnija je kaskadna konvolucijska neuronska mreža s više zadataka (engl. *Multi-Task Cascaded Convolutional Neural Network*) ili skraćeno MTCNN [14].

MTCNN pristup detekcije lica zasnovan je na 3 faze kaskadnih klasifikatora. U prvoj fazi potpuno se iskorištava konvolucija mreža prijedloga zvana P-Net kako bi se dobili prozori lica kandidata i njihovih vektori regresije graničnog okvira, nakon čega se kandidati kalibriraju na temelju tih vektora. U drugoj fazi svi se kandidati dostavljaju drugoj konvolucijskoj neuronskoj mreži, u nastavku CNN, zvanom precizna mreža ili R-Net koji tada odbija veliki broj lažnih kandidata i izvodi kalibraciju s regresijom graničnog okvira. I nakon u trećoj fazi se identificiraju regije lica s više nadzora [14].

Svaki od modela unutar kaskade, P-Net, R-Net i O-Net, treniran je za tri zadatka: klasifikaciju lica, regresiju graničnog okvira i lokalizaciju orijentira lica. No unatoč tome ova tri model nisu izravno povezani već se izlaz iz prethodne faze kaskadnih klasifikatora koristi kao ulaz u sljedeću fazu što omogućuje izvođenje dodatne obrade između faza [15].

MTCNN metoda za detekciju lica prilično složena za implementaciju no to ne znači da nije moguća ili da se ne koristi. Na početku je spomenuto kako je ona vjerojatno najpopularnija metoda dubokog učenja za detekciju lica jer postoje implementacije arhitekture otvorenog koda koje se mogu trenirati na raznim bazama podataka, kao i unaprijed trenirani modeli koji se mogu izravno koristiti [15].

Iako je najpopularnija MTCNN nije jedina metoda dubokog učenja koja se koristi prilikom detekcije lica. Valja spomenuti i još jednu metodu zvanu detektor za duboko gusto lice (engl. *Deep Dense Face Detector*) ili skraćeno DDFD [16].

DDFD je metoda dubokog učenja koja se koristi prilikom detekcija lica. DDFD nastoji iskoristiti visoki kapacitet dubokih konvolucijskih mreža za klasifikaciju i izdvajanje značajki kako bi se naučio jedan klasifikator za otkrivanje lica iz više pogleda te minimizirati računsku složenost pojednostavljuvanjem arhitekture detektora. Za razliku od brojnih drugih metoda koje nastoje postići isti cilj, detekciju lica, upravo ova ima minimalnu složenost jer ona ne zahtijeva dodatne komponente za segmentaciju, regresiju graničnog okvira ili SVM klasifikatore [16].

U nastavku se opisuje proces treniranja DDFD algoritma za detekciju lica. Autori metode započeli su s finim podešavanjem AlexNeta za detekciju lica izdvajanjem primjera obuke iz skupa podataka Annotated Facial Landmarks in the Wild, skraćeno AFLW . Koristeći podešenu duboku mrežu, moguće je uzeti pristup baziran na regiji ili pristup kliznom prozoru kako bi se dobio konačni detektor lica. Klasifikator lica u ovoj metodi sastoji se od 8 slojeva gdje su prvih 5 slojeva konvolucijski dok su posljednja 3 sloja su potpuno povezana. Prvo se potpuno povezani slojevi pretvaraju u konvolucijske slojeve preoblikovanjem parametara sloja što je zauzvrat omogućilo učinkovito pokretanje CNN-a na slikama bilo koje veličine i dobivanje toplinske karte klasifikatora lica za otkrivanje lica različitih veličina. Slike moraju biti skalirane gore ili dolje i tako se dobiju nove toplinske karte. Najbolji rezultati se postižu promjenom veličine slike 3 puta po oktavi što je napredak jer mnoge druge metode zahtijevaju znatno veći broj promjene veličine po oktavi [16].

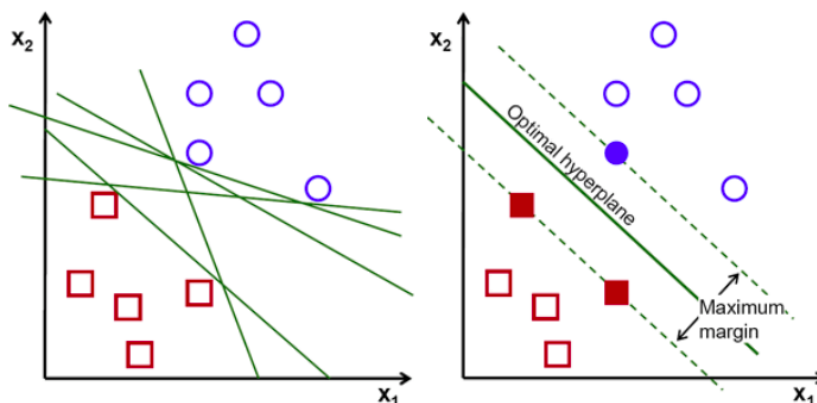
DDFD je prilično jednostavna metode, no unatoč svojoj jednostavnosti može postići vrhunske rezultate prilikom detekcije lica. Ova metoda duboko učenja može detektirati lica iz različitih kutova i donekle se nositi s okluzijom. No budući da je skup za obuku iskrivljen u odnosu na trening skup podataka s licima koja nisu poravnata, bolji se rezultati bolji se rezultati mogu postići kod uspravnih lica ili korištenjem sofisticiranijih tehnika povećanja podataka [16].

4.4. Metoda potpornih vektora

Metoda potpornih vektora (engl. *support-vector network*) ili SVM, kako će se u nastavku teksta oslovljavati, algoritam je za detekciju lica kojeg je razvio Vladimir Vapnik u AT&T Bell Laboratories.

[17] u svom radu opisuju kako SVM algoritam u suštini funkcioniра na sljedeći način: preslikava ulaz vektora u višedimenzijски prostor značajki Z kroz nelinearno preslikavanje koje je unaprijed definirano. U tom se prostoru konstruira linearna ploha odlučivanja s posebnim svojstvima koji osiguravaju visoku sposobnost generalizacije mreže.

Cilj SVM algoritma je pronaći hiperravninu u N -dimenzionalnom prostoru (Slika 4.1.) koja jasno klasificira podatkovne točke. Nastoji se pronaći ravnina koja ima maksimalnu marginu, odnosno maksimalnu udaljenost između točaka podataka obje klase (Gandhi, 2018). [18].



Slika 4.1. Moguće hiperravnine

Te spomenute hiperravnine su granice odlučivanja koje pomažu u klasifikaciji podatkovnih točaka jer one mogu pripadati različitim klasama. Dimenzija neke hiperravnine ovisi o broju značajki pa tako ako je broj ulaznih značajki 2 hiperravnina samo crta, a ako je broj ulaznih značajki 3 tada hiperravnina postaje dvodimenzionalna ravnina [18].

Metoda potpornih vektora kombinira 3 ideje:

1. Tehniku rješenja iz optimalnih hiperravnina - omogućuje proširenje vektora rješenja na vektore potpore
2. Konvolucija skalarnog umnoška – proširuje skup rješenja s linearno u nelinearno separabilne površine
3. Pojam mekih margina - omogućuje pogreške na skupu za trening [17]

5. Raniji algoritmi za prepoznavanje lica

Postoje različite vrste algoritama za prepoznavanje lica. U ovom dijelu rada fokus se stavlja na ranije algoritme koji se mogu koristiti za prepoznavanje lica, one razvijene 90-ih godina 20og stoljeća, a to su:

1. Fisherface
2. Eigenface
3. Histogrami lokalnih binarnih koeficijenata (LBPH),
4. Scale Invariant Feature Transform (SIFT)
5. Speed Up Robust Features (SURF)

Svaki od navedenih algoritama ima drugačiji pristup za izdvajanje informacija o slici i izvođenje povezivanja s ulaznom slikom. U nastavku se detaljno opisuju svaki od navedenih ranijih algoritama.

5.1. Fisherface

Jedan od popularnih algoritama koji se koristi prilikom prepoznavanja lica je Fisherface. Za razliku od drugih algoritama za prepoznavanje lica koje ćemo kasnije spomenuti, upravo se Fisherface, zbog svog nastojanja da maksimalno poveća razdvajanje između klasa u procesu treninga, smatra daleko superiornijim algoritmom.

Fisherface algoritam utemeljen je na Fisherovom linearnom diskriminatoru (engl. *Fisher's Linear Discriminant*), klasičnoj tehnici za prepoznavanje uzoraka, koji je 1936. godine razvio Robert Fisher. No kada pričamo o prepoznavanju lica tek su Chang i sur. 1991 godine predstavili metodu prepoznavanja lica koja je koristila Fisherov diskriminator. Kasnije su Baker i Nayar razvili teoriju odbijanja uzorka koja se temelji na dvoklasnom linearnom diskriminatoru [19].

Fisherov linearni diskriminator primjer je metode specifične za klasu jer on pokušava oblikovati raspršivanje kako bi ga učinio pouzdanim za klasifikaciju. Fisherova analiza ima za cilj d-dimenzionalni vektor značajki svesti na jednu dimenziju kako bi se kasnije mogao iskoristiti za klasifikaciju lica [19].

Kako zapravo Fisherface algoritam za prepoznavanje lica funkcionira [19] opisuje pomoću formula (5.1.1) i (5.1.2). Postavlja se pretpostavka da je X slučajni vektor uzeti iz c skupa

$$\mathcal{X} = \{X_1, X_2, \dots, X_c\} \quad (5.1.1)$$

$$X_i = \{X_{i1}, X_{i2}, \dots, X_{in}\} \quad (5.1.2)$$

Matrica raspršenja između klasa S_B računa se prema formuli (5.1.3.):

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (5.1.3)$$

Dok se matrica raspršenja unutar klase računa kao (5.1.4.):

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \quad (5.1.4.)$$

μ_i je srednja slika klase X_i , a N_i broj uzoraka u klasi X_i . Ako je S_W nesingularan, optimalna projekcija W_{opt} je odabrana kao matrica s ortonormalom stupaca koji maksimizira omjer determinante od matrica raspršenja između klasa projiciranih uzoraka na determinanta matrice raspršenja unutar klase projektiranih uzoraka. Jednostavnije rečeno algoritam tada traži projekciju W koja maksimizira kriterij razdvajanja klase uz pomoć formule (5.1.5.):

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} \quad (5.15)$$

Rješenje za taj optimizacijski problem dobivamo uz pomoć formule (5.1.6) i (5.1.7.):

$$S_B W_i = \lambda_i S_W W_i \quad (5.1.6)$$

Naposljetku iz toga napokon dobivamo W_{pca} (5.1.7.) i W_{fld} (5.1.8.):

$$W_{pca} = \arg \max_W |W^T S_T W| \quad (5.1.7.)$$

$$W_{fld} = \arg \max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \quad (5.1.8.)$$

Transformacijska matrica W koja projicira uzorak u $c-1$ u konačnosti je izračunata kao (5.1.9.):

$$W = W_{fld}^T W_{pca} \quad (5.1.9)$$

Iako se koristi već iznimno dugo, već gotovo 3 desetljeća, Fisherface algoritam za prepoznavanje lica još uvijek nalazi na neke probleme. Posebno se ističu dva problema, a to su stanje slike u ulazu koji će se koristiti kao testiranje slike te računanje. Računanje se razvilo u

problem jer ima vrlo kompliciran i vrlo složen proces računanja, dok problem se problem stanja slike odnosi na raznolikost svjetla, atribute lica, ekspresije lica i varijacija položaja slike samog lica [20].

5.2. Eigenface

Eigenface algoritam za prepoznavanje lica pretvara slike lica u skup karakterističnih slika lica zvanih “eigenfaces”, po kojima je algoritam i dobio ime, i one su ujedno i glavne komponente početnog skupa slika lica za trening. Prepoznavanje lica koristeći Eigenface odvija se projiciranjem nove slike u podprostor skeniran eigenfaces i zatim se lice klasificira uspoređivanjem njegovog položaja u prostoru lica s položajem poznatih pojedinaca. Ako je udaljenost položaja dovoljno mala, osoba se identificira. S druge strane, ako je udaljenost prevelika, slika se smatra onom koja pripada pojedincu za kojeg se sustav mora istrenirati [21].

Prilikom računanja Eigenfaces postavljamo pretpostavku da je slika lica $I(z,y)$ dvodimenzionalni N po N niz vrijednosti intenziteta ili vektor dimenzije N^2 . Na početku se uzima set slika lica $I_1, I_2, I_3 \dots I_M$ na kojima su lica prikazana na isti način i iste veličine. Cilj nam je izračunati prosječno lice svakog vektora r_i koristeći formulu (5.2.1.):

$$\Psi = \frac{1}{M} \sum_{i=1}^M r_i \quad (5.2.1.)$$

Svako lice se od prosječnog razlikuje po vektoru θ_i tako da se mora konstruirati blok matrica A od svih slika u vektorskom prikazu [21] uz formulu (5.2.2.):

$$\theta_i = r_i - \Psi \quad (5.2.2.)$$

Ovaj skup vrlo velikih vektora podliježe analizi glavne komponente koji traži skup M ortonormiranih vektora U_k i njihove pridružene vrijednosti X_k . Ti vektori U_k i X_k su ujedno i vektori vlastite kovarijacijske matrice C (5.2.3.) [21]:

$$C = \frac{1}{M} \sum_{n=1}^M \theta_n \theta_n^T = AA^T \quad (5.2.3.)$$

Gdje je matrica A (5.2.4.) jednaka [21]

$$A = [\theta_1 \theta_2 \dots \theta_M] \quad (N^2 \times M) \quad (5.2.4.)$$

Međutim, matrica C je $N^2 \times N^2$ i određivanje N^2 vektora i svojstvenih vrijednosti je složen zadatak za tipične veličine slike i potrebna nam je računski izvediva metoda za

pronalaženje ovih vlastitih vektora. Iz tog razloga matrica C računa se na drugi način uz formulu (5.2.5.) [21].

$$C_1 = A^T A \quad (M \times M) \quad (5.2.5.)$$

Eigenface se također može koristiti za otkrivanje i lociranje lica na pojedinačnim slikama te nam to dopušta da i prepoznamo prisutnost lica. Stvaranje vektora za sliku je ekvivalentno projiciranju slike na nižedimenzionalni prostor lica. Udaljenost E između slika i njene projekcije na prostor lica jednostavno je udaljenost između prosječnog lica (5.2.6.) [21]

$$\theta_i = r_1 - \Psi \quad (5.2.6.)$$

i njegove projekcije na prostor lica (5.2.7.):

$$\varphi = \sum_{i=1}^k w_i u_i \quad (5.2.7.)$$

Slike lica se ne mijenjaju radikalno kada se projicira u prostor lica, dok projekcija slika bez lica izgleda sasvim drugačije. Ova osnovna ideja se koristi za otkrivanje prisutnosti lica u sceni na način da se izračuna udaljenost E između lokalne podslike i prostora lica te je rezultat izračuna mapa lica $E(z, Y)$ [21].

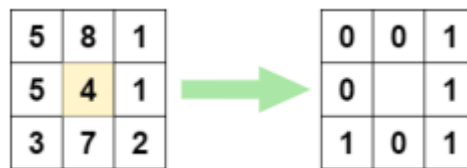
Razlozi za odabir Eigenface algoritma prilikom detekcije lica su mnogi. Naime upravo ona nudi veću stopu uspješnosti i brzine u odnosu na mnoge druge metode, jednostavna je za realizirati stvarnom vremenu i bez posebnog hardvera te je neovisna od geometrije lica. No to ne znači da je bez mane. Eigenface ima jednu izuzetno značajnu manu, a to je vrijeme izračuna. Ukoliko je baza velika proces identificiranja određene osobe može potrajati duže vrijeme [22].

5.3. Histogrami lokalnih binarnih uzoraka

Sljedeći algoritam prepoznavanja lica koji će se opisati su histogrami lokalnih binarnih uzoraka (engl. *Local Binary Patterns Histograms*) ili skraćeno LBPH. LBPH je iznimno popularna te jedan od najstarijih, no ne najstariji, algoritama za prepoznavanje lica koji je prvi put opisan 1996. godine [23].

Lokalni binarni uzorak (LBP) jednostavan je operator teksture koji označava piksele slike određujući prag susjedstva svakog piksela i smatra rezultat kao binarni broj. Kada LBP kombiniramo s histogramima orijentiranih gradijenta značajno se poboljšavaju performanse detekcije lica te slike lica možemo predstaviti jednostavnim vektorom podataka [23].

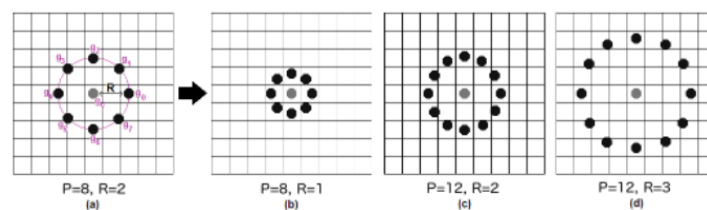
Osnovna ideja lokalnih binarnih uzoraka kod prepoznavanja lica je sažeti lokalnu strukturu na slici uspoređujući svaki piksel s njegovim susjedstvom. Neki piksel se uzima kao središte i postavlja se prag prema njegovim susjedima, odnosno prozor koji se ispituje podijeli se u ćelije (na primjer 16 x 16 piksela za svaku ćeliju) i svaki se piksel uspoređuje s 8 susjeda, sjedinjujući piksele u krug. Ako je vrijednost središnjeg piksela veća od vrijednosti susjednog tada se dodaje vrijednost 1, a ako je manja ili jednaka tada se dodaje vrijednost 0 [24] Na slici 5.1. prikazan je primjer izrade lokalnih binarnih uzoraka uzimanjem 8 susjednih piksela i njihova pretvorba u 8 binarnih brojeva.



Slika 5.1. Primjer izrade lokalnih binarnih uzoraka

LBPH algoritam koristi 4 parametra: radijus, susjedi, mreža X i mreža Y. Radijus se koristi za izgradnju kružnog lokalnog binarnog uzorka te on predstavlja polumjer oko središnjeg piksela, dok susjedi predstavljaju broj točaka uzorka za izgradnju kružnog lokalnog binarnog uzorka. Mreža X se odnosi na broj ćelija u vodoravnom smjeru, a mreža Y na broj ćelija u okomitom smjeru [23].

S obzirom na navedene parametre LBPH algoritam funkcioniра na sljedeći način: Skup podataka se stvara snimanjem slika, dodjeljivanjem jedinstvenog identifikatora i dodavanjem slika u bazu podataka. Dio skupa podataka koristi se za trening algoritma, dok se ostatak koristi za testiranje. Koristeći koncept kružnog susjedstva (Slika 5.2.), broj pojavljivanja LBP kodova na slici sastavlja se u histogram te se zatim provodi klasifikacija na način da se izračunaju osnovne sličnosti histograma koji se uspoređuju, te taj histogram sadrži opis pojedinca na tri različite razine: na razini piksela gdje se oznake kombiniraju u malom području kako bi se stvorila regionalna razina i naposljetku ti regionalni histogrami grade opći opis osobe [25].



Slika 5.2. Primjer koncept kružnog susjedstva

5.4. SIFT algoritam

Transformacija značajki neovisne o skali, odnosno SIFT, skraćeno od *Scale Invariant Feature Transform*, algoritam je za detekciju i deskripciju lokalnih značajki u slici, a može se koristiti i kod prepoznavanja lica kojeg je 1999. razvio David Lowe [26], a 2004. godine izdana je i poboljšana verzija. Najčešća područja primjene SIFT algoritma su prepoznavanje lica te trodimenzionalna rekonstrukcija različitih vrsta scena.

U suštini SIFT algoritam pretvara sliku u veliku kolekciju vektora lokalnih značajki, gdje svaki ostaje potpuno nepromjenjiv na prijevod slike, skaliranje i rotaciju, te djelomično nepromjenjiv na promjene osvjetljenja ili 3D projekcije [26].

Značajke nepromjenjive skale identificiraju se korištenjem postupnog pristupa filtriranja. Prva faza identificira ključne lokacije u prostoru mjerila tražeći mjesta koja su maksimumi ili minimumi Gaussove funkcije. Svaka točka se koristi za generiranje vektora značajki koji opisuje lokalnu regiju slike u odnosu na njen koordinatni sustav razmjera i prostora. Rezultirajući vektori značajki nazivaju se SIFT ključevi se koriste za indeksiranje i identifikaciju modela objekata kandidata. Zbirke ključeva najprije se identificiraju kroz Hash tablicu Houghove transformacije, a zatim pomoću najmanjih kvadrata koji se uklapaju u konačnu procjenu parametara modela. Kada se najmanje 3 ključa slažu oko parametara modela s malom razlikom, postoji jak dokaz za prisutnost objekta [26].

SIFT algoritam sastoji se od četiri ključna koraka:

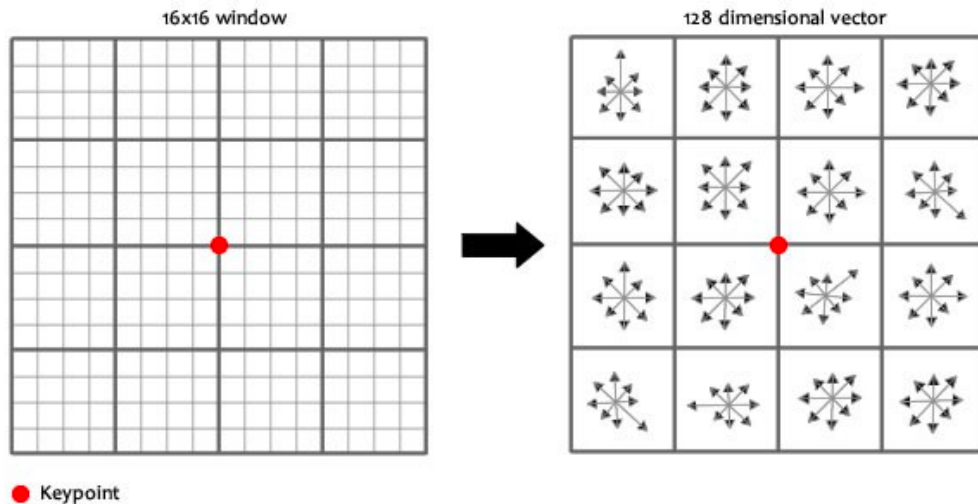
1. Detekcija skale ekstremnih razmjera (engl. *Scale-space peak selection*)
2. Lokalizacija ključnih točaka (engl. *Keypoint Localization*)
3. Dodjela orijentacije (engl. *Orientation Assignment*)
4. Generiranje opisa (engl. *Keypoint descriptor*)

Kod prvog koraka, detekcije skale ekstremnih razmjera, identificira se lokacija i razmjer ključnih točaka. Prostor mjerila slike je funkcija $L(x,y,\sigma)$ (5.4.1.) [27].:

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y) \quad (5.4.1.)$$

koja se proizvodi Gausovim zamućenjem ulazne slike. Ljestvica-prostor je podijeljen na oktave, a broj oktava i ljestvica ovisi o veličini izvorne slike. Na taj se način generiraju oktave izvorne slike, te je veličina slike svake oktave upola manja od one prethodne. Drugi korak je lokalizacija ključnih točaka i njihovo pročišćavanje odbacivanjem točaka niskog

kontrasta. Sada kad imamo legitimne ključne točke slijedi dodjela orijentacije svoj ključnoj točki kako bi ona ostala nepromjenjiva u rotaciji (Slika 5.3.). Kada svaka ključna točka ima mjesto, mjerilo, orijentaciju potrebno je generirati njen opis za lokalno područje slike o svakoj ključnoj točki na način da uzmemo prozor 16x16 oko ključne točke koji je podijeljen je na 16 podblokova veličine 4x4 te da se za svaki podblok kreira se histogram orijentacije s 8 orijentacijskih ćelija koji daju deskriptor ključne točke sa 128 elemenata te se tada konačno dobiva vektor značajki za formiranje deskriptora ključne točke [27].



Slika 5.3. Generiranje opisa ključnih točaka

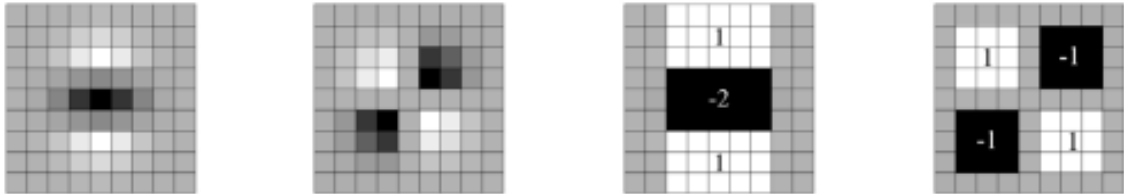
5.5. SURF algoritam

Posljednji od ranijih algoritama za za detekciju i deskripciju lokalnih značajki u slici, a može se koristiti i kod prepoznavanja lica koji će se spomenuti su ubrzane robusne značajke, odnosno SURF algoritam, što je kratica od *Speed Up Robust Features*. SURF je djelomično inspiriran prethodno opisanim SIFT algoritmom, no standardna verzija SURF algoritma nekoliko je puta brža od SIFT-a [28].

SURF se sastoji od dva koraka: ekstrakcija značajke i opis značajke. Za ekstrakciju značajki detektor koristi Hessianovu matricu zbog njenih dobrih performansi prilikom vremena i točnosti računanja, no koristi vrlo osnovnu aproksimaciju. Oslanja se na integralne slike kako bi se smanjilo vrijeme računanja i stoga ga nazivamo 'Fast-Hessian' detektorom [28]. S obzirom na točku $x = (x, y)$ Hessiova matrica $H(x, \sigma)$ u mjerilu σ definirana je sljedećom formulom (5.5.1.):

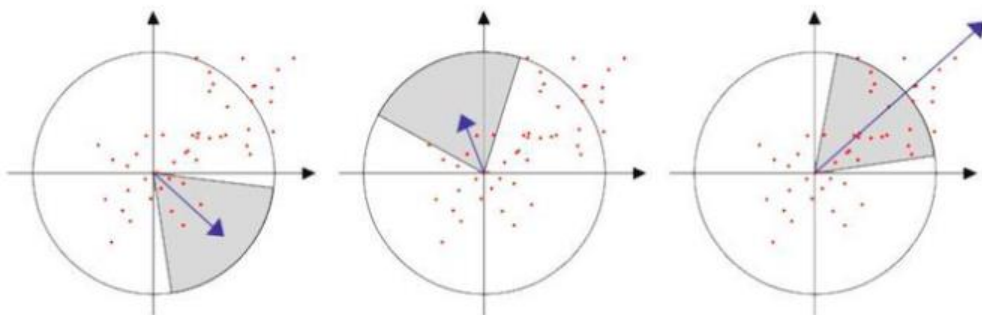
$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (5.5.1.)$$

gdje je $L_{xx}(x, \sigma)$ konvolucija derivacije drugog reda Gaussove funkcije sa slikom I u točki x , a slično za $L_{xy}(x, \sigma)$ i $L_{yy}(x, \sigma)$. Gauss je idealan za analizu razmjera i prostora, ali u praksi se moraju diskretizirati i izrezati (Slika 5.4.) što dovodi do gubitka ponovljivosti pri rotaciji slike oko neparnih višekratnika $\pi/4$ [28].



Slika 5.4. Diskretizacija i rezanje

Kreiranje SURF deskriptora značajki odvija se u dva koraka: dodjela orijentacije i komponente deskriptora. Prvi korak sastoji se od fiksiranja reproducibilne orijentacije na temelju informacija iz kružnog područja oko ključne točke računanjem Haarovog waveleta u x i y smjeru, u kružnom susjedstvu radijusa $6s$ (s odgovara σ u derivaciji Gaussove funkcije) oko ključne točke, gdje je s skala na kojoj je pronađena ključna točka (Slika 5.5). Nakon toga računa se suma rezultata vertikalnog i horizontalnog waveleta u području skeniranja te mijenjamo orijentaciju skeniranja (dodamo $\pi/3$) i ponovno računamo dok ne pronađemo orijentaciju s najvećom vrijednošću sume. U drugom koraku se konstruira kvadratna regija usklađena s odabranom orijentacijom te se iz nje računa SURF deskriptor [28].



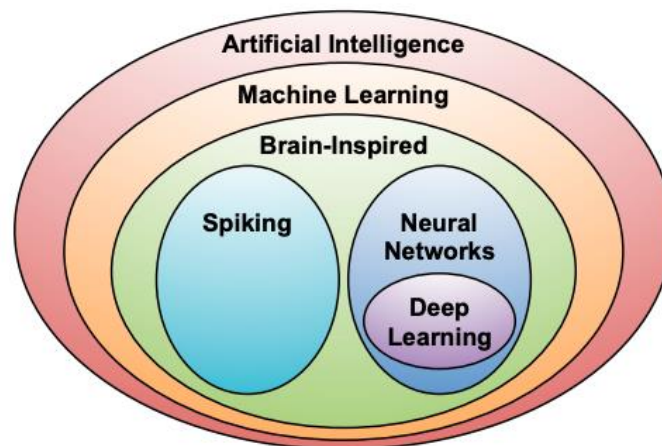
Slika 5.5. Fiksiranje reproducibilne orijentacije

6. Duboke neuronske mreže

6.1. Definiranje dubokih neuronskih mreža

Pojmovi umjetna inteligencija i duboko učenje pojmovi su koji se zadnjih desetljeća sve češće koriste u tehnološkom polju stoga nije čudo da su se pojavili i u kontekstu biometrije i prepoznavanja lica. Oblik dubokog učenja na kojem je fokus ovog poglavlja su duboke neuronske mreže koje su dio šireg polja umjetne inteligencije.

Duboko učenje produkt je razvoja umjetne neuronske mreže koja može postići aproksimaciju složene funkcije povećanjem skrivenih slojeva korištenjem dubokih neuronskih mreža za rješavanje problema strojnog učenja [29].



Slika 6.1. Duboko učenje u kontekstu umjetne inteligencije

Kako bi bolje razumjeli što su duboke neuronske mreže pogledajmo sliku 6.1. koja predoduje duboko učenje u kontekstu umjetne inteligencije. Umjetna inteligencija (engl. *artificial intelligence*), u nastavku AI, naziv je koji pridajemo znanosti koja se bavi stvaranjem inteligentnih strojeva koji imaju sposobnost postizanja ciljeva i snalaženja u novim situacijama. Unutar AI nalazi se polje strojnog učenja (engl. *machine learning*) koje računalima daje mogućnost učenja bez točnog programiranja, odnosno računalo je sposobno raditi aktivnosti izvan pojma programiranja. Unutar strojnog učenja nalazi se polje računanja inspiriranog mozgom (engl. *brain-inspired computation*) koje se dijeli na dva dijela: *spiking computing* i neuronske mreže. *Spiking computing* je polje inspirirano ljudskim mozgom i činjenicom da je komunikacija u našem mozgu zasnovana na impulsima nalik na šiljke i da se informacije koje

se prenose ne temelje samo na amplitudi šiljka već i na vremenu kada puls stigne i da proračun koji se događa u neuronu nije funkcija samo jednog vrijednost osim širine impulsa i vremenskog odnosa između različitih impulsa. Područje suprotno od spiking computinga, su neuronske mreže [30].

Duboke neuronske mreže su raznolike i dolaze u mnoštvu različitih vrsta. Stalno se razvijaju i poboljšavaju, no u svakom slučaju ulaz u DNN mora biti skup vrijednosti koje predstavljaju informacije koje će mreža ispitati [30].

Postoje dva glavna oblika mreža koje obrađuju ulazne podatke: mreže za prosljeđivanje podataka i mreže koje se ponavljaju. Mreže za prosljeđivanje podataka (engl. *Feed-forward networks*) izvode računanje kao niz operacija na izlazima prethodnog sloja, a zatim konačni skup operacija generira izlaz cijele mreže. Još jedna činjenica vrijedna pažnje o mrežama za prijenos podataka je da ova vrsta mreže nema memoriju i da je izlaz za ulaz uvijek isti bez obzira na redosljed ulaza koji su prethodno dani mreži. Međutim, mreže koje se ponavljaju (engl. *Recurrent networks*) su drugačije i one zapravo imaju internu memoriju koja omogućuje dugoročnim ovisnostima da utječu na izlaz i operacije generiraju vrijednosti koje su pohranjene unutar mreže i koriste se kao ulaz za druge operacije. Nadalje, duboke neuronske mreže mogu biti sastavljene od potpuno povezanih slojeva (engl. *fully-connected layers*) u kojima su sve izlazne aktivacije sastavljene od ponderiranog zbroja svih ulaznih aktivacija što zahtijeva značajnu količinu pohrane i računanja. Također, postoje DNN mreže s rijetko povezanim slojevima (engl. *sparsely connected layers*) u kojima su neke veze između aktivacija uklonjene postavljanjem težine na nulu bez utjecaja na točnost [30].

Oblik dubokih neuronskih mreža koji zaslužuje poseban spomen su konvolucijske neuronske mreže (engl. *Convolutional Neural Networks*) ili skraćeno CNN. Konvolucijske neuronske mreže sastoje se od višestrukih konvolucijskih slojeva koji su pak sastavljeni od višedimenzijских konvolucija, a svaki sloj generira apstrakciju više razine ulaza koja čuva ključne informacije [30].

6.2. Povijest

Priča o neuronskim mrežama započinje prije 80 godina, točnije u 1943. godine kada su matematičari Warren McCulloch i Walter Pitts u sklopu svog rada “*A Logical Calculus of Ideas Immanent in Nervous Activity*” kreirali osnovni sustav temeljen na algoritmu za oponašanje aktivnosti ljudskog mozga koji je postao prve neuronska mreža. Unapređenje u polju

neuronskih mreža došlo je 1957. godine kada je Frank Rosenblatt, značajna figura u području umjetne inteligencije i otac dubokog učenja, izumio revolucionarni algoritam za rješavanje kompliciranih zadataka prepoznavanja koji se naziva Perceptron i koji se koristi za klasifikaciju linearno odvojenih vektora [31].

Istraživanja u polju neuronskih mreže stagnirala su gotovo 20 godina, te su tad neuronske mreže postale proširene zbog zamjene funkcija praga korištenih u Perceptronu s diferencijabilnom sigmoidnom funkcijom. Taj napredak doveo je do ponovnog vala interesa u polju istraživanja neuronskih mreža, no zbog nedostataka računalnih kapaciteta potrebnih za analizu velike količina podataka taj je val ponovnog interesa usporio u četiri desetljeća koja su uslijedila [31].

Početak 2010-ih došlo je do procvata u ovom području kada su neuronske mreže i umjetna inteligencija uzele maha. Glavni razlog za to je što su u 2000-ima informatičari konačno dobili ono što im je trebalo da nastave istraživanje u ovom polju, naime procesorska snaga i brzina računala jako su porasle. Osim toga, tri su glavna čimbenika koja dovode do uspjeha dubokog učenja upravo u tom trenutku. Prvi čimbenik je količina dostupnih informacija za trening mreža jer ipak ono zahtijeva veliku količinu podataka za trening. Drugi čimbenik je količina dostupnog računalnog kapaciteta jer se sada trening dubokih neuronskih mreža može izvesti u razumnom vremenskom roku što zasigurno nije bilo realno u dvadesetom stoljeću. Konačno, treći čimbenik je evolucija algoritamskih tehnika koje su značajno poboljšale točnost rezultata i proširile potencijal korištenja dubokih neuronskih mreža u razna polja [30].

Prije 2012. godine točnost algoritama za prepoznavanje lica koji koriste duboke neuronske mreže imala je stopu pogreške od 25%. To se promijenilo 2012. godine kada su se počele koristiti grafičke procesorske jedinice (GPU) zbog njihove visoke računalne sposobnosti i smanjile su stopu pogreške na već impresivnih 10%. Od tog trenutka algoritmi dubokog učenja rezultirali su stalnim nizom poboljšanja, a stopa pogrešaka kontinuirano je padala. Do 2015. algoritmi su premašili ljudsku razinu točnosti sa stopom pogreške koja je pala ispod 5%. Od tada je stopa pogreške pala ispod 1% s algoritmom za prepoznavanje lica koji koristi duboke neuronske mreže nazvane FaceNet postižući visoku točnost od 99,63% [30].

6.3. Arhitektura

Ključni čimbenik koji razlikuje duboke neuronske mreže od drugih vrsta neuronskih mreža je da duboke neuronske mreže koriste hijerarhijsku arhitekturu, na taj način oponašajući hijerarhijski način mozga za obavljanje kognitivnih zadataka.

U dubokom učenju neuronske mreže imaju više od tri sloja, odnosno više od jednog skrivenog sloja. Prosječan broj mrežnih slojeva koji se koriste u dubokom učenju kreće se od pet do više od tisuću. Na primjer, VGG16 arhitektura neuronske mreže sastoji se od 16 slojeva, a ResNet arhitektura od 152 sloja [32].

Prvih nekoliko slojeva dubokih neuronskih mreža izvodi ekstrakciju značajki korištenjem nenadziranog učenja i isključivo težine slojeva koje se nalaze između posljednjeg skrivenog i izlaznog sloja i oni se treniraju u nadziranom načinu rada. Metoda koja se najčešće koristi za ekstrakciju značajki između ulaznog i skrivenog sloja je autoenkoder, dok je druga najčešće korištena metoda potpuno suprotna, a to je metoda Boltzmanov stroj koja se koristi za izvođenje nenadziranog učenja u obliku klasteriranja [33].

Duboke neuronske mreže sposobne su naučiti značajke visoke razine s većom kompleksnošću nego druge neuronske mreže. Svaki sloj u neuronskoj mreži sastoji se od težina (engl. *weights*) i pomaka (to je broj s kojim se zbraja općenito, $\text{izlaz} = \text{ulaz} * \text{težina} + \text{pomak}$) koji označavaju brojeve koji povećavaju ulaz i koriste se prilikom treniranja dubokih neuronskih mreža [32].

Cilj treniranja duboke neuronske mreže je pronaći prave težine koje vode do najtočnijeg rezultata. Prije nego što se neuronska mreža uvježba, težine se inicijaliziraju te neuronska mreža mijenja te težine na temelju onoga što nauči. Prilikom izgradnje neuronske mreže potrebno je odabrati matematičku funkciju koja uzima izlazne podatke iz neuronske mreže i temeljne istinite podatke te izračunava koliko je rezultat neuronske mreže bio drugačiji. Ta funkcija se naziva funkcija gubitka. Koristeći tehnike optimizacije, računalo zatim izračunava kako promijeniti težine i pomake tako da se minimizira funkcija gubitka. To nastavlja raditi dok se trenira i više vremena, težine i pomaci se prilagođavaju podacima i na kraju imamo neuronsku mrežu koja ima visoku izlaznu točnost. Većina algoritama za prepoznavanje lica koji koriste duboke neuronske mreže u današnje vrijeme imaju prosječnu točnost od oko 97 do 99% [32].

7. Algoritmi za prepoznavanje lica koristeći duboke neuronske mreže

Nakon opisivanja koncepta prepoznavanja lica i dubokih neuronskih opisat će se popularne duboke neuronske mreže koje se koriste u kontekstu algoritama za prepoznavanje lica. Točnije, u nastavu će biti opisani sljedeći algoritmi za prepoznavanje lica:

- FaceNet
- DeepFace
- DeepID
- VGGFace
- GoogLeNet
- SqueezeNet

7.1. DeepFace

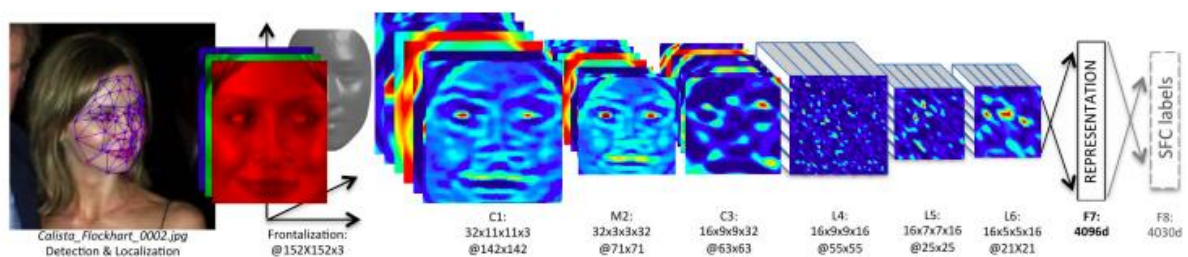
DeepFace je algoritam za prepoznavanje lica kojeg su 2014. godine Taigman, Yang, Ranzato i Wolf [34], istraživači u Facebooku, predstavili u svom radu *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*. Sa svojom točnošću od 97.35% na LFW (Labeled Face sin the Wild) skupu lica DeepFace nadmašuje većinu dosadašnjih sustava za prepoznavanje lica te je na rubu točnosti ljudske razine.

Ova duboka neuronska mreža od devet dubokih slojeva uključuje više od 120 milijuna parametara koristeći nekoliko lokalno povezanih slojeva umjesto standardnih konvolucijskih. Algoritam je treniran na velikom skupu podataka lica s oznakom identiteta od četiri milijuna slika lica koje pripadaju više od 4000 identiteta u Social Face Classification (SFC) skup lica razvijen od Facebooka, što je najveći skup podataka o licu do sada, prikupljen od populacije koja se uvelike razlikuje od one korištene za procjenu ranije navedene točnosti [34].

DeepFace prvo poravnava lice koristeći 3D model "prosječnog" lica tako da osoba na slici bude okrenuta naprijed nakon čega koristi duboko učenje kako bi pronašao numerički opis lica. Ako dvije slike imaju numerički opis DeepFace detektira da one pokazuju isto lice. Proces poravnanja lica 3D modelom ide kako slijedi: Prvo otkrivamo 6 referentnih točaka unutar izreza za otkrivanje, dvije na očima, jedna na vrhu nosa i tri točke na usnama, koje se koriste za

približno skaliranje, rotiranje i prevođenje slike na šest sidrišnih lokacija. Zatim pomoću tih otkrivenih točaka generiramo 2D sliku lica izrezanu iz izvorne slike. Na tu 2D sliku primjenjujemo mapu sa 67 referentnih točaka s njihovom odgovarajućom Delauneyevom triangulacijom kako bismo poravnali rotacije izvan ravnine. Konačna faza je frontalizacija poravnanja koja se postiže podijeljenom afinom transformacijom od izvora do cilja, usmjerena Delaunayevom triangulacijom izvedenom iz prethodnih 67 referentnih točaka [34].

DeepFace je osposobljen za višeklasno prepoznavanje lica, točnije za klasificiranje identiteta slike lica. Uzima ulaz u 3D poravnatu RGB sliku veličine 152x152 piksela. Slika zatim prolazi kroz konvolucijski sloj koji sadrži 32 filtra veličine 11x11x3 i to se dovodi do sloja za maksimalnim udruživanje (engl. *max-pooling layer*) koji zasebno za svaki kanal uzima maksimalno 3x3 prostornih susjedstava s korakom od 2. Zatim slijedi drugi konvolucijski sloj koji posjeduje 16 filtra veličine 9x9x16. Svrha ova tri sloja je izdvajanje značajki niske razine, poput jednostavnih rubova i tekstura. No maksimalno udruživanje se primjenjuje samo na prvi konvolucijski sloj, dok su sljedeća tri sloja lokalno povezani slojevi, poput konvolucijskog sloja, koji primjenjuju banku filtara, no svako mjesto na karti značajki uči drugačiji skup filtara. Korištenje lokalno povezanih slojeva može se opravdati činjenicom da na svaku izlaznu jedinicu lokalno povezanog sloja utječe vrlo veliki dio ulazne slike (ili regija ulazne slike), kod kojeg općenito nema statističke sličnosti za različite slike. Zadnja dva sloja modela su potpuno povezani slojevi i pomažu u uspostavljanju korelacije između dva udaljena dijela lica. Završna faza DeepFace algoritma je normalizacija. To znači da se značajke moraju normalizirati tako da budu između nule i jedan kako bi se smanjila osjetljivost na promjene osvjetljenja [34]. Vizualni prikaz gore opisane DeepFace arhitekture može se vidjeti na slici 7.1.



Slika 7.1. Arhitektura DeepFace algoritma

DeepFace se veoma impresivan algoritam za prepoznavanje lica i sa svojom točnošću od 97.35% pokazao se uspješnijim u od prosječnih algoritama, no već godinu dana kasnije, 2015. godine predstavljen je algoritam FaceNet koji ima točnost od 99.63% te mu je on postao najpopularniji algoritam za prepoznavanje lica.

7.2. FaceNet

FaceNet je algoritam za prepoznavanje lica koji su opisali Florian Schroff i sur. 2015 godine u svom radu *FaceNet: Unified Embedding for Face Recognition and Clustering* [35]. FaceNet se smatra vrhunskim algoritmom za prepoznavanje lica jer ima najveću točnost, čak 99.63% za LFW skup slika, te je upravo iz tog razloga i daleko najpopularniji.

FaceNet algoritam koristi duboku konvolucijsku neuronsku mrežu kako bi prepoznao lice na način da uči mapiranje iz slika u euklidski prostor gdje L2 udaljenost (L2 norma) odgovara mjeri sličnosti lica. Lica iste osobe imaju male udaljenosti, a lica različitih osoba imaju velike udaljenosti [35].

FaceNet izravno trenira svoj izlaz da bude 128-dimenzionalan, koristeći metodu trostrukog gubitka temeljenu na LMNN (Large Margin Nearest Neighbor) koja minimizira udaljenost između ulazne slike i slika koje sadrže isti identitet dok s druge strane maksimizira udaljenost između ulazne slike i slika koje sadrže različiti identitet [35].

FaceNet u svojoj arhitekturi koristi učenje od kraja do kraja (engl. *end-to-end learning*). U svrhu toga koristi se metoda trostrukog gubitka koja odražava ono što se želi postići prilikom provjere, prepoznavanja i grupiranja lica. Zapravo teži se ugrađivanju $f(x)$, iz slike x u d -dimenzionalni euklidski prostor obilježja R_d , tako da je kvadrat razmaka između svih lica istog identiteta mali, dok je kvadrat udaljenosti između para slika lica iz različitih identiteta je velika [35]. To možemo prikazati formulom (7.2.1.):

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2, \forall (x_i^a, x_i^p, x_i^n) \in T \quad (7.2.1.)$$

gdje je α margina koja se provodi između pozitivnih i negativnih parova, T je skup svih mogućih trojki u skupu za trening i ima kardinalnost N [35]. Gubitak koji se minimizira tada je prikazan formulom (7.2.2.):

$$\sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha] + \quad (7.2.2.)$$

Kako bi se osigurala brza konvergencija, ključan je odabir trojke koje krše ograničenje tripleta. To znači da s obzirom na x_i^a , moramo odabrati trojke takve da je maksimum (7.2.3.)

$$\operatorname{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2, \quad (7.2.3.)$$

a da je minimum (7.2.4.)

$$\operatorname{argmin}_{x_i^n} \|f(x_i^{\alpha}) - f(x_i^n)\|_2^2 \quad (7.2.4.)$$

Generiranje trojke na temelju cijelog skupa je u načelu skupo tako da se za to koristi jedna od sljedećih metoda:

1. Generiranje trojki na svakom koraku na temelju prethodnih kontrolnih točaka i izračunavanje minimuma i maksimuma na podskupu podataka.
2. Odabir tvrdog pozitivnog x_i^p i tvrdog negativnog x_i^n korištenjem minimuma i maksimuma na mini seriji [35].

7.3. DeepID

DeepID, skraćeno od *Deep hidden IDentity features*, je skup značajki visoke razine reprezentacije kroz duboko učenje koji se koristi za prepoznavanje lica. Prvi put su ga predstavili Sun, Wang i Tang [36] 2014. godine u svom radu “*Deep Learning Face Representation from Predicting 10,000 Classes*”, no od tada su objavili još dva rada: “*Deep Learning Face Representation by Joint Identification-Verification*” i “*DeepID3: Face Recognition with Very Deep Neural Networks*” u kojima opisuju nove verzije algoritma, DeepID2 i DeepID3.

Kako bi naučio više nego dovoljan broj (over-complete) potrebnih reprezentacija značajki visoke razine DeepID koristi duboku konvolucijsku neuronsku mrežu koja je naučena klasificirati sve slike lica dostupne za trening po njihovim identitetima, gdje je posljednja aktivacija neurona skrivenog sloja označena kao značajka. Svaka konvolucijska mreža kao ulaz uzima dio lica ili regiju lica i izvlači lokalne značajke niske razine u donjim slojevima. Brojevi značajki smanjuju se duž kaskade izdvajanja značajki, dok se u gornjim slojevima formiraju globalne značajke i značajke visoke razine. Na kraju kaskade koja sadrži podatke o identitetu i razmatra veliki broj klasa identiteta dobiva se kompaktan 160-dimenzionalni DeepID [36].

DeepID sadrži četiri konvolucijska sloja koji mogu hijerarhijski izdvojiti značajke, nakon čega slijedi DeepID sloj i softmax izlazni sloj koji ukazuje na klase identiteta. Unos je $39 \times 31 \times k$ za pravokutne regije i $31 \times 31 \times k$ za kvadratne regije, gdje je $k = 3$ za regije u boji i $k = 1$ za sive regije. Kada se ulazne veličine promjene, visina i širina mapa u slijedećim slojevima promijenit će se u skladu s tim. Dimenzija sloja DeepID je fiksna na 160, ali dimenzija izlaznog sloja varira. Brojevi značajki nastavljaju se smanjivati duž hijerarhije izdvajanja značajki sve do sloja DeepID koji predviđa veliki broj klasa identiteta sa samo nekoliko značajki. To je izraženo u formuli (7.3.1.) gdje su X_i i Y_j i-ta mapa ulaza i j-ta mapa izlaza [36]:

$$y^{j(r)} = \max(0, b^{j(r)} + \sum_i k^{ij(r)} * x^{i(r)}) \quad (7.3.1.)$$

Sloj maksimalnog udruživanja (engl. *max pooling*) računa se formulom iz nastavka u kojoj se svaki neuron u i-toj izlaznoj mapi Y_i skuplja preko $s \times s$ nepreklapajuće lokalne regije u i-toj ulaznoj mapi X_i (7.3.2.) [36].

$$y_{j,k}^i = \max_{0 < m, n < s} \{ x_{j*s+m, k*s+n}^i \} \quad (7.3.2.)$$

Nakon sloj maksimalnog udruživanja posljednji skriveni sloj DeepID-a povezan je s trećim i četvrtim konvolucijskim slojem. Povezivanje trećeg konvolucijskog i posljednjeg skrivenog sloja smanjuje gubitak informacija u četvrtom sloju. Posljednji skriveni sloj opisan je sljedećom formulom (7.3.3.) u kojoj X_1 , W_1 , X_2 i W_2 označavaju neurone i težinu u trećem i četvrtom sloju [36].

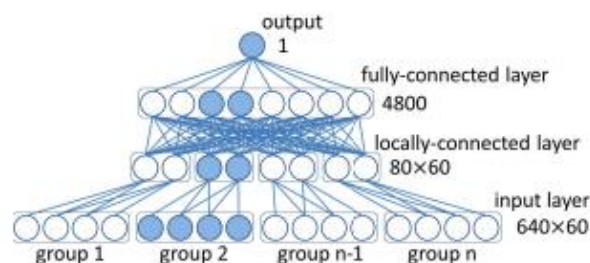
$$y_j = \max(0, \sum_i x_i^1 * w_{i,j}^1 + \sum_i x_i^2 * w_{i,j}^2 + b_j) \quad (7.3.3.)$$

Konvolucijska mreža uči se minimiziranjem $-\log Y_t$ s t-tom ciljnom klasom (7.3.4.) [36].

$$y_i = \frac{\exp(y_i^f)}{\sum_j^n \exp(y_j^f)} \quad (7.3.4.)$$

Kao i u DeepFace metodi, DeepID, metodom detekcije točaka lica, također detektira pet točaka na licu, dvije na očima, jednu na vrhu nosa i dvije u kuta usta. Značajke se zatim izdvajaju iz 60 podregija lica s deset regija, tri skale i RGB ili sivim kanalima [36].

Zbog svoje visoke uspješnosti, za verifikaciju lica koristi se Joint Bayesian tehnika. Neuronska mreža sadrži jedan ulazni sloj koji uzima DeepID, jedan lokalno povezani sloj, jedan potpuno povezani sloj i jedan izlazni neuron koji pokazuje sličnosti lica. Ulazne značajke podijeljene su u 60 grupa, od kojih svaka sadrži 640 značajki iz određenog para. Neuroni u lokalno povezanom sloju povezuju se samo s jednom grupom značajki kako bi naučili njihove odnose i smanjili dimenziju značajke, dok je drugi skriveni sloj potpuno povezan s prvim skrivenim slojem kako bi se naučili globalni odnosi. Jedan izlazni neuron potpuno je povezan s drugim skrivenim slojem. Skriveni neuroni su ReLU, a izlazni neuron je sigmoidni. Vizualni prikaz ovoga može se vidjeti na slici 7.3. Točnost ove mreže je 97.45% na skupu za validaciju od LFW skupa lica [36].



Slika 7.3. Struktura neuronske mreže

DeepID2 ima tri konvolucijska sloja nakon kojih slijedi *max pooling* i jedan lokalno povezani sloj te jedan potpuno povezani sloj. Zajednički nadzorni signali identifikacije i

verifikacije dodaju se posljednjem potpuno povezanom sloju i u nekoliko povezanih slojeva. U fazi treninga značajke DeepID2 izdvajaju se u gornjem skrivenom sloju i zatim mapiraju na jedan od velikog broja identiteta putem druge funkcije. U fazitestiranja značajke se mogu generalizirati na druge zadatke i nove identitete koji se ne vide u podacima o treningu. Ova verzija algoritma DeepID nadmašila je prvu s točnošću od 99,15% no ciljalo se na više pa je tako razvijen i DeepID3 [37].

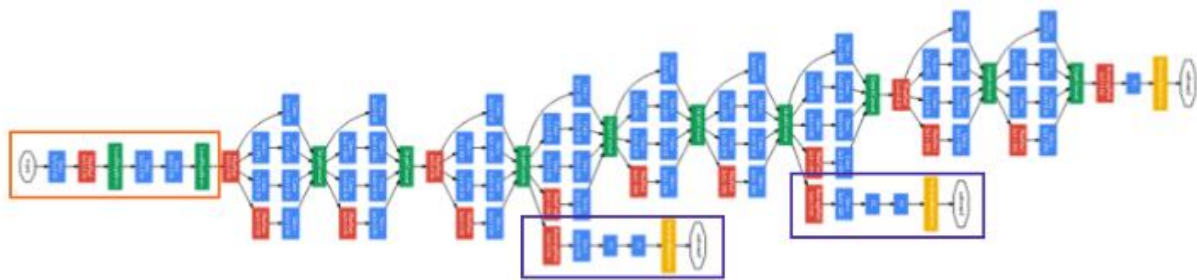
DeepID3 predložen je 2015. godine i on je opisan kao arhitektura od dvije neuronske mreže koje su znatno dublje od prethodnog DeepID2 u smislu da DeepID3 ima 10 do 15 slojeva za ekstrakciju nelinearnih značajki, a DeepID2 ima samo pet. Dubina mreže rezultat je slaganja više slojeva prije svakog sloja za udruživanje. U fazi treniranja završnom sloju ekstrakcije značajki dodaju se zajednički nadzorni signali za identifikaciju lica.. Iako je istreniran na istom skupu podataka kao i DeepID2 on ima točnost verifikacije od 99,53% na LFW skupu lica [38].

U konačnici možemo zaključiti kako DeepID metoda ima točnost prepoznavanja lica od 97,45% na *Labeled Faces in the Wild* (LFW) skupu podatka i to koristeći samo slabo poravnata lica, što je gotovo jednako dobro kao ljudsko oko koje mjeri 97,53%, no manje je od prethodno spomenutih DeepFace i FaceNet algoritama. No verzija DeepID2 ima točnost od 99.15%, dok najnovija verzija DeepID3 broji 99.53%. Time je prestignuta točnost DeepFace algoritma za prepoznavanje lica no ne i FaceNet-a koji još uvijek zauzima prvo mjesto s 99.63% točnosti.

7.4. GoogLeNet

GoogLeNet je vrsta duboke neuronske mreže koju su predložili istraživači iz Googlea 2014. godine u istraživačkom radu pod naslovom “Going Deeper with Convolutions”.

Arhitektura GoogLeNet (Slika 7.4.) sastoji se od 22 sloja ako računamo samo slojeve s parametrima ili od 27 slojeva ako računamo slojeve udruživanja. Ukupni broj neovisnih građevnih blokova korištenih za izgradnju mreže je 100, no taj broj ovisi o sustavu strojnog učenja koji se koristi. Ideja koja stoji iza ove arhitekture je računalna učinkovitost kako bi se mogla izvoditi na pojedinačnim uređajima čak i s malim računalnim resursima. Također sadrži dva pomoćna sloja klasifikatora povezana s izlazom Inception slojeva. Arhitektura GoogLeNet također se sastoji od devet početnih modula. Postoje dva sloja za maksimalno udruživanje (engl. *Max pooling*) između početnih modula čija je svrha smanjivanje uzorkovanja ulaza dok se prosljeđuje kroz mrežu, te se to postiže smanjenjem visine i širine ulaznih podataka [39].



Slika 7.4. GoogLeNet arhitektura

Novost u GoogLeNetu su pomoćni klasifikatori (engl. *auxiliary classifier*) koji se dodaju međuslojevima arhitekture. Njihova je svrha izvršiti klasifikaciju na temelju ulaza unutar središnjeg dijela mreže i dodati gubitke izračunate tijekom treninga natrag na ukupne gubitke mreže. Detalji pomoćnih klasifikatora su: prosječni objedinjeni sloj veličine filtra 5×5 i pomakom od 3, konvolucija 1×1 sa 128 filtara za smanjenje dimenzija i ispravljenom linearnu aktivaciju, potpuno povezani sloj s 1024 jedinice i ispravljenom linearnom aktivacijom, regularizacija ispadanja s omjerom odbacivanja od 70% i softmax klasifikator s 1000 klasa [39].

Zanimljivo je da je GoogLeNet bio pobjednik na izazovu klasifikacije slika ILSVRC 2014. godine gdje je bio prvi s greškom od 5 najvećih od 6,67%, bez ikakvog treninga s vanjskim podacima što je gotovo isto kao čovjek [39].

7.5. SqueezeNet

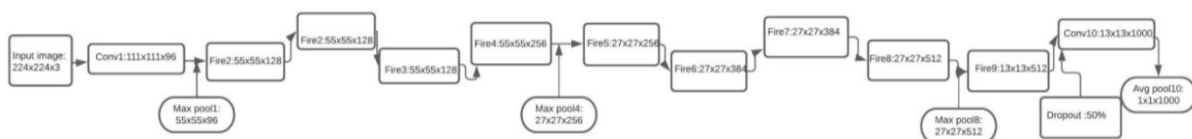
SqueezeNet je duboka neuronska mreža koju su razvili istraživači Forrest Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, Bill Dally i Kurt Keutzer [40] iz DeepScalea, Sveučilišta Kalifornija, Berkeley i Sveučilišta Stanford, a objavljena je 2016. godine.

Cilj SqueezeNet-a bio je razviti manju neuronsku mrežu s manje parametara koja se može lakše smjestiti u memoriju računala i lakše prenositi. Postoje tri glavna razloga za želju za manjim CNN-om: zahtijevaju manje komunikacije između poslužitelja tijekom treninga i manju propusnost za izvoz novog modela iz oblaka te su izvediviji za implementaciju na hardveru s ograničenom memorijom [40].

SqueezeNet model se sastoji od konvolucijskog sloja (engl. *Convolution layer*), „fine modula“ i slojeva za udruživanje (engl. *Max Pooling*). Konvolucijski sloj i sloj za maksimalno udruživanje (engl. *max pooling*) isti su kao u mnogim drugim modelima; Konvolucijski sloj

transformira ulaznu sliku kako bi iz nje izvukao značajke, dok Max Pooling sloj izvodi operaciju skupljanja koja izračunava maksimalnu vrijednost u svakoj regiji svake mape značajki. Fire modul novi je građevni blok za izgradnju CNN arhitekture i sastoji se od sloja *squeeze convolution* koji se unosi u sloj za proširenje koji ima kombinaciju 1x1 i 3x3 konvolucijskih filtara [40].

SqueezeNet je konvolucijska neuronska mreža koja ima 18 slojeva (Slika 7.5.). SqueezeNet počinje samostalnim slojem konvolucije (conv1), nakon čega slijedi 8 Fire modula (fire2-9), a završava završnim slojem konvolucije (conv10). Broj filtara postupno se povećava po fire modulu od početka do kraja mreže. SqueezeNet izvodi max-pooling s pomakom 2 nakon slojeva conv1, fire4, fire8 i conv10. Odbacivanje s omjerom od 50% primjenjuje se nakon modula fire9. Kada treniramo SqueezeNet, počinjemo sa stopom učenja od 0,04 i linearno smanjujemo stopu učenja tijekom treninga [40].



Slika 7.5. 18 slojeva SqueezeNet modela

Glavni cilj razvoja SqueezeNet-a bio je identificirati CNN arhitekture koje imaju malo parametara uz zadržavanje konkurentne točnosti. Da bi to postigli, autori su osmislili sljedeće tri strategije: zamijeniti 3x3 filtre s 1x1 filtrima, smanjiti broj ulaznih kanala 3x3 filtre korištenjem slojeva za smanjivanje dimenzionalnosti i na kraju smanjiti poduzorkovati kasno u mreži tako da slojevi konvolucije imaju velike aktivacijske mape [40].

7.6. VGGFace

VGGFace odnosi se na niz modela razvijenih za prepoznavanje lica od strane članova Visual Geometry Group (VGG) sa Sveučilišta u Oxfordu. Postoje dva glavna VGG modela za prepoznavanje lica VGGFace i VGGFace2. VGG su opisali *Parkhi, Vedaldi i Zisserman* [41] u svom radu “Deep Face Recognition” 2015. godine, a VGGFace2 opisan je 2018. godine od strane [42] u radu “*VGGFace2: A dataset for recognising faces across pose and age*”. U nastavku teksta pobliže se opisuje svaki od njih.

Cilj VGGFace-a je stvoriti veliki skup podataka o licima koji zahtijeva samo ograničenu količinu procesorske snage za označavanje prikupljanjem podataka o licima korištenjem izvora znanja dostupnih na webu. Ovaj skup podataka se koristi kao osnova za razvoj dubokih konvolucijskih neuronskih mreža za identifikaciju i provjeru lica. Ti se modeli treniraju na velikom skupu podataka, te se zatim procjenjuju na referentnim skupovima za prepoznavanje lica [41].

Arhitektura VGGFace-a sastoji se od 11 blokova, od kojih svaki sadrži linearni operator iza kojeg slijedi jedna ili više nelinearnosti. Prvih osam blokova su konvolucijski dok su posljednja tri potpuno povezana, oni su isti kao konvolucijski sloj ali veličina filtra odgovara veličini ulaznih podataka. Nakon svih konvolucijskih slojeva slijedi rektifikacijski sloj (ReLU). Ulaz u sve mreže je slika lica veličine 224×224 s oduzetom prosječnom slikom lica [41].

Model VGGFace treniran je pomoću tri opadajuće stope učenja. Težine filtra u konvolucijskoj neuronskoj mreži su inicijalizirane nasumično iz Gaussove distribucije, dok su slike za trening promijenjene tako da je manja dimenzija od širine ili visine jednaka 256. Tijekom treninga, mreža se hrani nasumičnim regijama od 224×224 piksela izrežanim iz slika. Podaci se dodatno povećavaju okretanjem slike lijevo desno s 50% vjerojatnosti. Za učenje ugrađivanja korištenjem trostrukog gubitka, mreža je zamrznuta osim posljednjeg potpuno spojenog sloja koji implementira diskriminativnu projekciju. Autori prvo opisuju proces treninga klasifikatora lica koji koristi softmax aktivacijsku funkciju u izlaznom sloju za klasificiranje lica kao ljudi. Ovaj se sloj zatim uklanja tako da je izlaz mreže vektorski prikaz lica, što se naziva ugrađivanje lica. Model se zatim dodatno uvježbava uz pomoć finog podešavanja kako bi se euklidska udaljenost između vektora generiranih za isti identitet smanjila, a vektori generirani za različite identitete povećali [41]. To se postiže korištenjem trostruke funkcije gubitka a prikazuje se formulom (7.6.1.):

$$E(W^l) = \sum_{(a,p,n) \in T} \max\{0, \alpha - \|X_a - X_n\|_2^2 + \|X_a - X_p\|_2^2\}, \quad X_i = W^l \frac{\theta(\varphi_i)}{\|\theta(\varphi_i)\|_2} \quad (7.6.1.)$$

VGGFace2 je mnogo veći skup lica od VGGFace jer sadrži 3,31 milijuna slika od 9131 osobe, od kojih je 59,3% muškaraca i 40,7% žena, varirajući između 80 i 843 slike za svaki identitet s prosjekom od 362,6 slika za svaku osobu. Slike su preuzete s Googlea i imaju velike varijacije u dobi, dobi, osvjetljenju, pozici i etničkoj pripadnosti [42].

Proces prikupljanja skupa podataka sastoji se od 6 faza. Prva faza je dobivanje i odabir popisa imena pronalaženjem što je moguće više subjekata koji imaju dovoljno jasan skup

dostupnih slika i zatim uklanjanjem identiteta s popisa kandidata koji nemaju dovoljno jasnih slika. Druga faza je dobivanje slika za svaki identitet tako da na kraju imamo 1400 slika za svaki identitet. Zatim konačno dolazi faza detekcije lica koristeći isti model kao u prethodno spomenutom MTCNN algoritmu. Nakon otkrivanja lica dolazi automatsko filtriranje prema klasifikaciji kako bi se uklonila neuobičajena lica za svaki identitet. U fazi 6 dolazi do uklanjanja duplikata klasteriranjem VLAD (Vector of locally aggregated descriptors) deskriptora za sve slike koje ostaju u fazi 4 i zadržavanjem samo jedne slike po grupi. Konačno, posljednja faza je završno automatsko i ručno filtriranje kako bi se uklonile sve greške [42].

Sve mreže VGGFace2 osposobljene su za klasifikaciju pomoću funkcije gubitka softmax. Tijekom treninga prošireni granični okvir lica mijenja se tako da kraća stranica iznosi 256 piksela, a zatim se područje od 224×224 piksela izrezuje iz svakog uzorka. Modeli se treniraju na skupu podataka, posebno na ResNet-50 i posebno za SENet koji se zatim ocjenjuju na standardnim skupovima podataka za prepoznavanje lica pokazujući izvrsnu izvedbu. Usporedbom između ResNet-50 i SENet-a pokazano je da SENet ima mnogo bolje performanse kod verifikacije i kod identifikacije te da se njegove performanse mogu dodatno poboljšati treningom na dva skupa podataka VGGFace2 i MS1M [42].

8. Praktični dio

Prepoznavanje lica biometrijska je metoda sposobna usporediti ljudsko lice sa slike s bazom podataka lica korištenjem specifičnih algoritama za precizno određivanje karakterističnih crta lica s dane slike. Te karakteristike lica, poput udaljenosti između očiju, oblika nosa ili brade, pretvaraju se u matematički prikaz i uspoređuju s podacima o drugim licima koji se nalaze u bazi podataka.

Razvoj algoritama koje se koriste prilikom prepoznavanja lica započeo je još šezdesetih godina dvadesetog stoljeća no još dan danas oni se razvijaju, uče i testiraju kako bi bili što efikasniji i točniji. Prije desetak godina većina algoritama za prepoznavanje lica koji koriste duboke neuronske mreže imala je stopu pogreške od 25%. Uvođenjem grafičke procesorske jedinice ta se stopa smanjila za 10% i od tada nastavila kontinuirano padati i do 2015. godine stopa pogreška je pala ispod 5%, a nakon toga čak i na ispod 1%.

U prethodnom dijelu rada opisano je nekoliko algoritama za prepoznavanje lica koji su razvijeni uz pomoć dubokih neuronskih mreža, a to su FaceNet, DeepFace, DeepID, GoogLeNet i SqueezeNet. FaceNet, DeepFace i DeepID trenutno se smatraju najpopularnijim i najtočnijim algoritama s visokom stopom točnosti. Ljudsko oko mjeri točnost prepoznavanja lica s 97.53% što je manje od ovih algoritama. Naime, FaceNet ima točnost od 99.63%, DeepFace od 97.35% i DeepID od 97,45%, no uzmemo li u obzir novije verzije DeepID2 broji točnost od 99.15%, a DeepID3 od 99.53%.

U nastavku rada istražiti će se točnost GoogLeNet i SqueezeNet algoritama koji su manje korišteni za prepoznavanje lica od ostalih te usporediti njihovi rezultati.

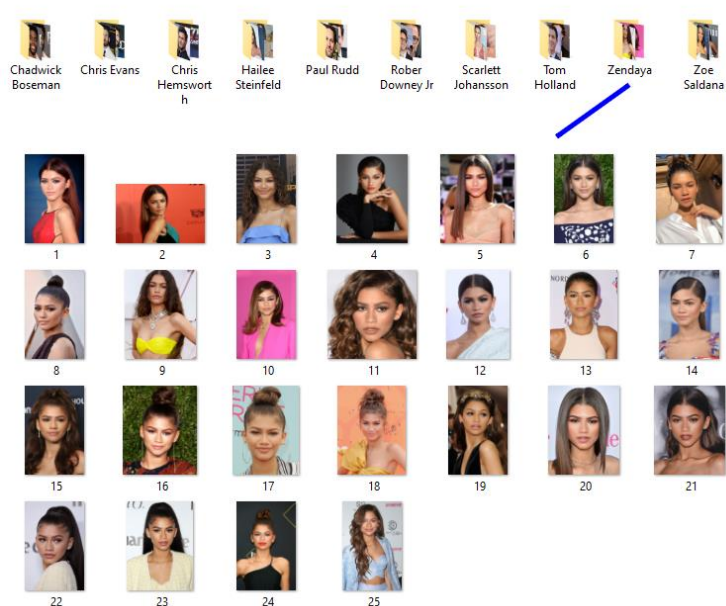
8.1. Priprema za testiranje

Kako bih usporedili GoogLeNet i SqueezeNet algoritme za prepoznavanje lica uz pomoć dubokih neuronskih mreža prvo ih moramo istrenirati i testirati nad istom bazom podataka.

Osnovna stavka koja nam je potrebna za realizaciju je okruženje. Ovom prilikom koristi se Matlab. Matlab je programska i numerička računalna platforma i programski jezik visoke razine razvijen od strane The MathWorks Inc. On integrira programiranje, vizualizaciju i računanje u okruženje koje je poprilično jednostavno za korištenje. Matlab je sposoban istražiti,

modelirati, vizualizirati i analizirati podatke, kreirati skripte, funkcije, klase te desktop i mobilne aplikacije, izvršiti velike proračune i paralelne simulacije pomoću višejezgrenih stolnih računala, GPU-ova, klastera i oblaka, te koristiti vanjska jezična sučelja i raditi u oblaku. Prva verzija Matlaba nastala je 1970. godine u suradnji sveučilišta Stanford u Kaliforniji i sveučilišta Novi Meksiko. Prva verzija, MATLAB 1.0, razvijena je 1979 godine kada je on zapravo bio jednostavni matrični kalkulator sa 71 unaprijed ugrađenom funkcijom te je tek 1984. godine reprogramiran u C jeziku, stvoren MATLAB programski jezik i pušten za komercijalno korištenje. Tokom desetljeća Matlab se kontinuirano razvijao, poboljšavao i širio moć svoje upotrebe. Danas se on koristi u gotovo svim aspektima informacije tehnologije, od strojnog i dubokog učenja, preko kontrolnih sustava, procesiranja signala i bežičnih komunikacija, te sve do obrade slike, računalnog vida i robotike (MathWorks, bez dat). Verzija Matlaba koja se koristila prilikom ovog istraživanja je R2022A Update 4 (9.12.0.2009381) od 64-bitna (win64) koja je aktivna od 7. lipnja 2022. godine.

Kako bismo mogli usporediti rezultate točnosti GoogLeNet i SqueezeNet algoritama potrebno ih je obučiti i testirati nad istom bazom lica. Prije nego što ih možemo obučiti prvo moramo prikupiti skup podataka o licima. Proces kreiranja baze lica u praksi se naziva upis prepoznavanja lica. To se smatra upisom jer upisujemo i registriramo korisnika kao primjer osobe u našem skupu podataka i aplikaciji. Za potrebe ovog rada izrađena je vlastita baza lica (Slika 8.1.) koja se sastoji od ukupno 250 fotografija lica, podijeljenih tako da imamo 10 osoba i za svaku osobu 25 različitih fotografija lica iz različitih kutova.



Slika 8.1. Korištena baza lica

8.2. SqueezeNet

Nakon kreirane baze podataka slijedi implementacija rješenja uz SqueezeNet algoritam. Kako bi mogli raditi s SqueezeNetom potreban nam je Matlab dodatak *Deep Learning Toolbox* koji pruža okruženje (engl. *framework*) za implementaciju plitkih i dubokih neuronskim mreža s algoritmima, unaprijed treniranim modelima i aplikacijama.

Kako bi dobili točnost prepoznavanja i detekcije lica potrebno je algoritam istrenirati i testirati nad bazom podataka. Iz tog razloga kreirane su dvije skripte: `training.m` i `test_network.m`.

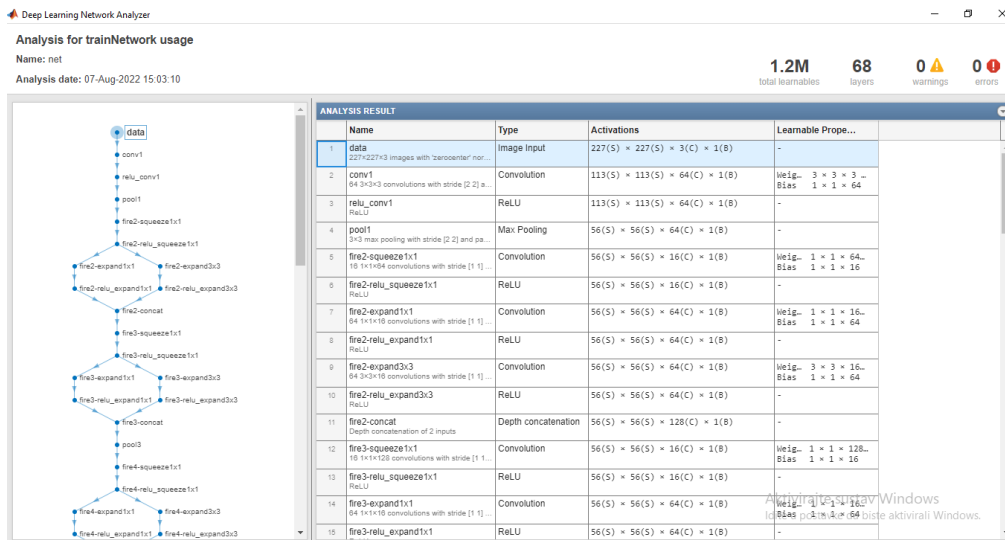
Skripta `training.m` služi za treniranje algoritma na prethodno kreiranom bazom podataka koja je nazvana `Dataset`. Baza se povezuje uz funkciju `imageDatastore`. Podaci unutar baze su organizirani su u različite mape pa moramo u funkciju uključiti i podmape na način da postavimo `'IncludeSubfolders'` na `'true'`. Imena mapa su oznake kategorija skupa podataka pa je zato argument funkcije `'LabelSource'`, `'foldernames'`.

```
Dataset = imageDatastore('Dataset', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
```

Nakon organizacije baze ona se mora podijeliti u skup za obuku i skup za validaciju uz pomoć funkcije `'splitEachLabel()'`. Ova funkcija zahtijeva dva argumenta, skup podataka koji trebamo podijeliti i omjer dijeljenja koji je u ovom slučaju 8/10, što znači da se 80% podataka koristi za treniranje i 20% za validaciju.

```
[Training_Dataset, Validation_Dataset] = splitEachLabel(Dataset,0.8);
```

Nakon što je povezana baza potrebno je i povezati algoritam s kojom želimo raditi, u ovom slučaju to je SqueezeNet. SqueezeNet je povezan na način da je uzeta varijabla `net = squeezeNet`. Nakon povezivanje mreže pozvana je funkcija `analyzeNetwork(net)` kako bi se dobio uvid u arhitekturu mreže. Nakon toga kod je pokrenut i učita se Deep Learning Network Analyzer koji pokazuje kompletnu arhitekturu SqueezeNet koji se sastoji od 68 sloja (Slika 8.2.1.).



Slika 8.2. Arhitektura SqueezeNet

Iz prvog sloja vidljivo je kako slike moraju biti rezizane te je to korak koji slijedi.

```
Input_Layer_Size = net.Layers(1).InputSize(1:2);
```

```
Resized_Training_Data = augmentedImageDatastore(Input_Layer_Size, Training_Data);
```

```
Resized_Validation_Data = augmentedImageDatastore(Input_Layer_Size, Validation_Data);
```

Kako bi se mreža istrenirala nad prethodno izrađenoj bazi podataka konvolucijski sloj i output sloj se modificiraju. Konvolucijki sloj modificira se uz varijablu `New_Convolutional_Layer` koja poziva funkciju `convolution2dLayer`.

```
New_Convolutional_Layer = convolution2dLayer([1, 1], Number_of_Classes, ...
```

Arhitektura mreže mijenja se jer se mijenjao predefimirani konvolucijki sloj s novim, modificiranim slojem. To se je postignuto na slijedeći način:

```
New_Network = replaceLayer(Network_Architecture, 'conv10', New_Convolutional_Layer);
```

```
New_Network = replaceLayer(New_Network, 'ClassificationLayer_predictions',  
New_Classification_Layer);
```

S ovim korakom definirana je mreža i sada slijedi specifikacija opcija treniranja na način da uzmemo varijablu `Training_Options` koja poziva funkciju `trainingOptions` u kojoj onda specificiramo željene opcije.

```
Training_Options = trainingOptions('sgdm', ...
```

```
'MiniBatchSize', 4, ...
```

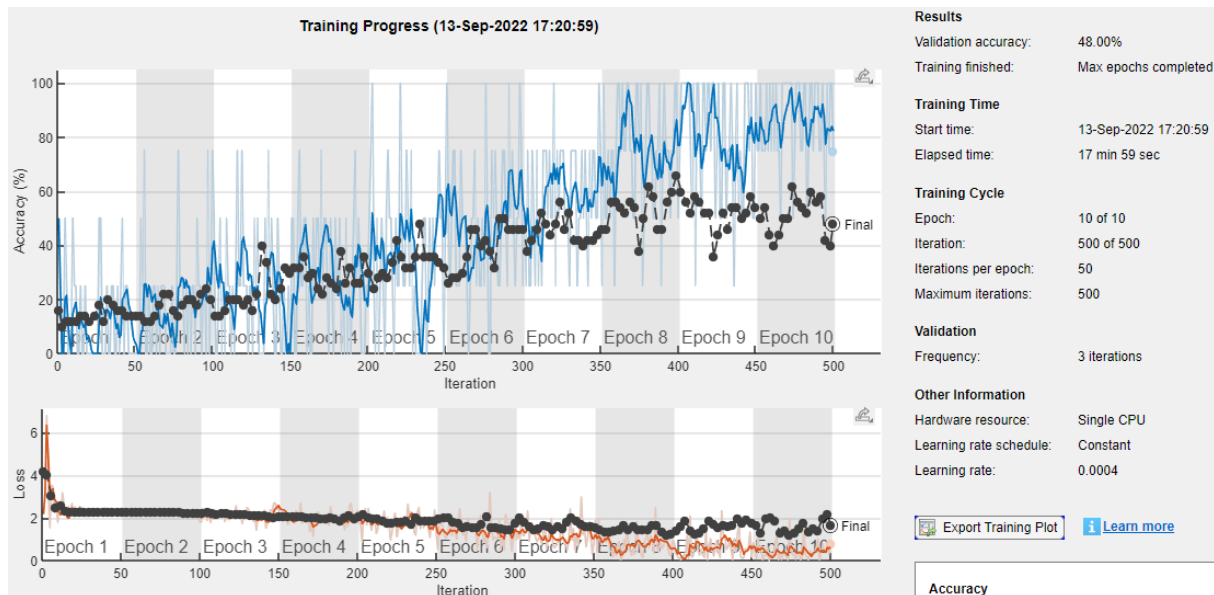
```
'MaxEpochs', 10, ...
```

```

'InitialLearnRate', 4e-4, ...
'Shuffle', 'every-epoch', ...
'ValidationData', Resized_Validation_Data, ...
'ValidationFrequency', 3, ...
'Verbose', false, ...
'Plots', 'training-progress');

```

Uz varijablu `Trained_Network` pozivamo funkciju `trainNetwork` i konačno se pokreće kod i slijedi treniranje. Slika 8.2.2. prikazuje proces treniranja mreže. Vrijeme potrebno za treniranje mreže nad bazom od 250 lica bilo je 17 minuta i 59 sekundi, s 10 epoha i 50 iteracija po epohi. Točnost provjere valjanosti (engl. *vallidation accuracy*) je 48.00%



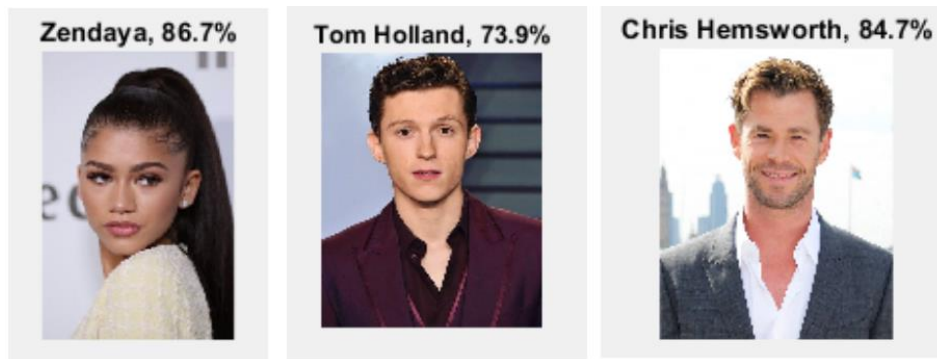
Slika 8.3. Treniranje SqueezeNeta

Nakon uspješnog treniranja mreže istu je potrebno testirati. Za svrhu testiranja kreirana je skrivena `test_network.m`. Na početku se naravno mora učitati mreža što se čini ponovnim korištenjem varijable `net` i pozvati funkcija `test_network` uz koju povežemo sliku osobe čiju točnost želimo provjeriti. Na primjer `>>test_network('Image1.jpg', net)`. Kako bi se pokazalo ime detektirane osobe, koje je pozvano iz korištene baze lica, kao i maksimalni postotak vjerojatnosti predviđanja (engl. *probability prediction*), tj. da je to lice od te osobe, korišten je sljedeći dio koda.

```
label = Label(index(i));
```

```
title(string(label) + ", " + num2str(100*max(Probability(index(i), :)), 3) + "%");
```

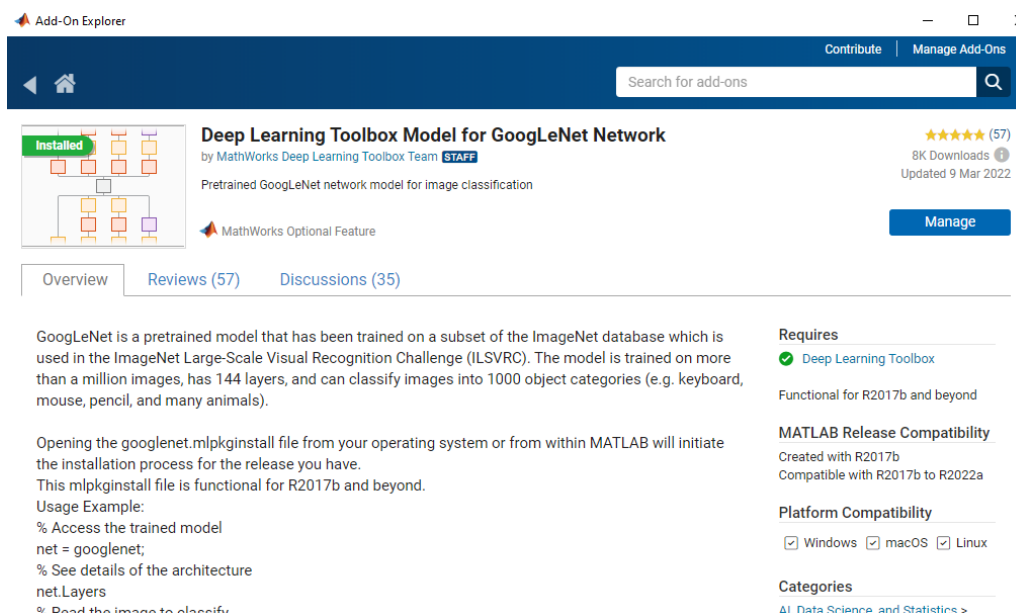
Pokretanjem funkcije `test_network` dolazimo do rezultata. Slika 8.2.3. prikazuje rezultate prepoznavanja lica za 3 poznate osobe, glumaca Tom Hollanda i Chrisa Hemswortha te glumice Zendaye. Iz priložene slike vidljivo je kako je lice Zendaye prepoznato uz postotak vjerojatnosti predviđanja od 86,7%, lice Toma Hollanda s 73,9% i Chrisa Hemswortha od 84,7%.



Slika 8.4. Rezultati testiranja

8.3. GoogLeNet

Kako bi se moglo krenuti raditi s GoogLeNetom prvo je potrebno unutar Matlaba instalirati dodatak Deep Learning Toolbox Model for GoogLeNet Network (Slika 8.3.1.).



Slika 8.5. Add-On za instalaciju GoogLeNeta

Tek nakon uspješne instalacije može se početi raditi sa GoogLeNetom. Kako i kod SqueezeNeta i ovdje će se kreirati dvije skripte: `training.m` i `test_network.m`, gdje `training.m` služi za treniranje algoritma a `test_network` za testiranje. Treniranje i testiranje odvija se na

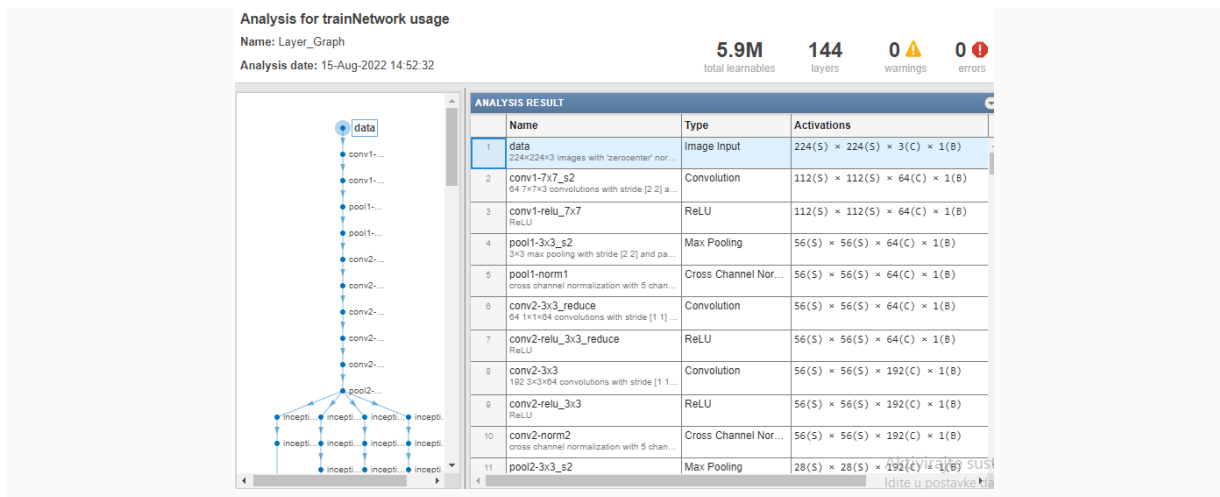
prethodno kreiranoj bazi podataka pod imenom Dataset koja se sastoji od 250 fotografija lica od sveukupno 10 poznatih osoba. I ovdje je prvi korak povezivanje baze lica. Baza se povezuje uz varijablu dana set na koju je povezana funkcija imageDatastore.

```
Dataset = imageDatastore('Dataset','IncludeSubfolders',true,'LabelSource','foldernames');
```

Nakon povezivanja baze ona se dijeli u skup za obuku i skup za validaciju pomoću funkcije 'splitEachLabel()' koja zahtijeva dva argumenta, skup podataka koji trebamo podijeliti i omjer dijeljenja. Ovdje je omjer dijeljenja postavljen na 8/10 što znači da će podaci biti podijeljeni u dvije grupe, gdje grupa za treniranje ima 80% podataka, a grupa za validaciju 20%.

```
[Training_Dataset,Validation_Dataset] = splitEachLabel(Dataset,0.8);
```

Zatim je učitani GoogLeNet i pohranjen u varijablu pod nazivom 'net', nakon čega se poziva varijabla analyzeNetwork(net) koja pokazuje grafičku strukturu mreže (Slika 8.3.2.)



Slika 8.6. Arhitektura GoogLeNeta

GoogLeNet se sastoji od 144 sloja. Slojevi su unaprijed uvježbani pa ih je potrebno zamijeniti, točnije slojevi koje treba zamijeniti su 142, 'loss3-classifier' i sloj 144 koji je odgovoran za klasificiranje objekata. Nakon što su zamijenjeni, vrijeme je da definiramo vlastite slojeve. Prije svega, definirat ćemo potpuno povezani sloj, koji je pohranjen u varijabli 'New_Featrue_Learner'. Koristimo funkciju 'fullyConnectedLayer()' za definiranje potpuno povezanog sloja i funkciju 'classificationLayer()' za definiranje sloja klasifikacije.

```
New_Feature_Learner = fullyConnectedLayer(Number_of_Classes, ...
```

```
'Name','Facial Feature Learner', ...
```

```
'WeightLearnRateFactor', 10, ...
```

```
'BiasLearnRateFactor', 10);
```

```
New_Classifier_Layer = classificationLayer('Name', 'Face Classifier');
```

Zatim se postojeći slojevi zamjenjuju redefiniranim slojevima pomoću funkcije 'replaceLayer()'. Funkcija 'replaceLayer()' zahtijeva graf sloja, naziv postojećeg sloja i novi sloj kao argumente. Ova funkcija vraća podatke grafikona slojeva o novom sloju koji se zatim pohranjuju u varijablu 'Layer_Graph'.

```
Layer_Graph = replaceLayer(Layer_Graph, Feature_Learner.Name,  
New_Feature_Learner);
```

```
Layer_Graph = replaceLayer(Layer_Graph, Output_Classifier.Name,  
New_Classifier_Layer);
```

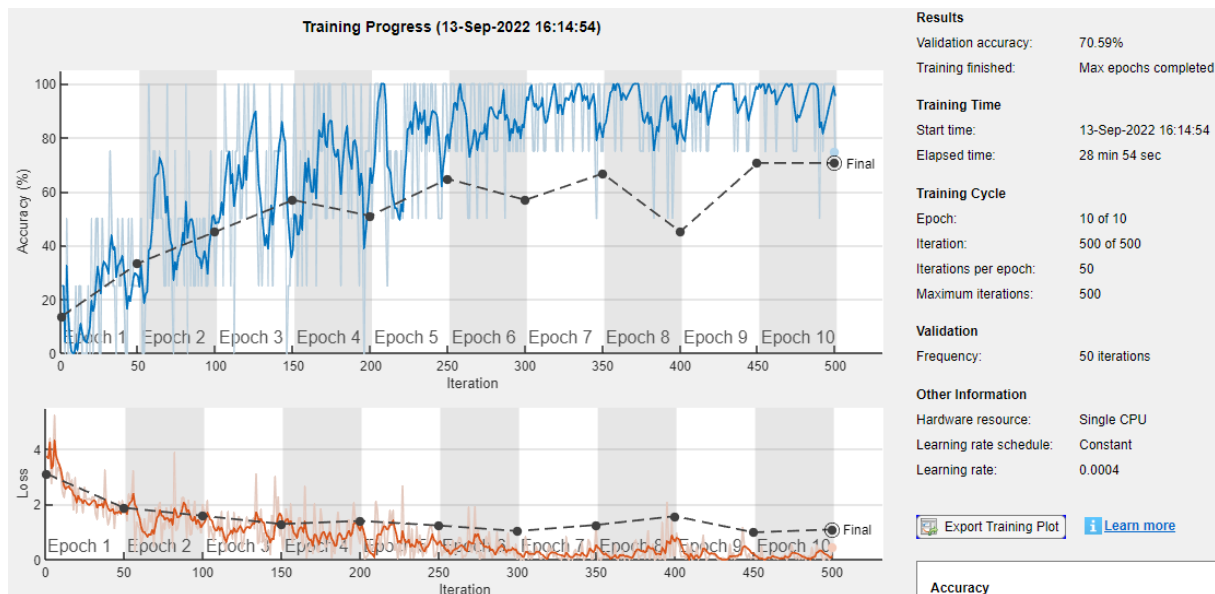
Nakon što su slojevi zamijenjeni vrijeme je za treniranje. No prije toga mora se definirati vrijednost za varijablu *ValidationData* bez koje treniranje nije moguće i to se radi na način:

```
Augmented_Validation_Image =  
augmentedImageDatastore(Input_Layer_Size(1:2),Validation_Dataset);
```

Treniranje se odvija uz pomoć funkcije `trainingOptions()` unutar koje i specificiramo željene opcije treniranja.

```
Training_Options = trainingOptions( 'sgdm', ...  
'MiniBatchSize', Size_of_Minibatch, ...  
'MaxEpochs', 10,...  
'InitialLearnRate', 4e-4, ...  
'Shuffle','every-epoch', ...  
'ValidationData', Augmented_Validation_Image, ...  
'ValidationFrequency',Validation_Frequency, ...  
'Verbose', false, ...  
'Plots','training-progress');
```

Slika 8.3.3. prikazuje proces treniranja koji je trajao 28 minuta i 54 sekundi s 10 epoha i 50 iteracija po epohi. Točnost provjere valjanosti (engl. *validation accuracy*) je 70.59%

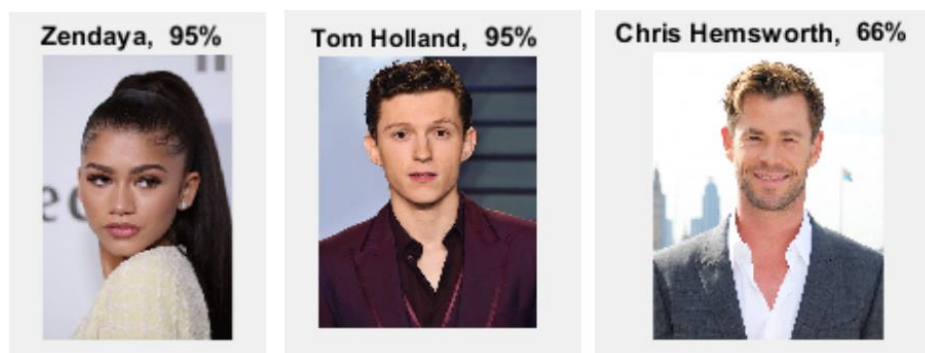


Slika 8.7. Treniranje GoogLeNeta

Nakon izvršenog treniranja kreira se skripta `test_network.m` koja će služiti za testiranje mreže. Testiranje se odvija uz pomoć funkcije `test_network` koja poziva GoogLeNet i sliku osobe čije lice želimo prepoznati.

function test_network(net, image)

Pokretanjem funkcije `test_network` dolazimo do rezultata. Slika 8.3.4. prikazuje rezultate prepoznavanja istih lica kao za Squeezenet, glumaca Tom Hollanda i Chrisa Hemswortha te glumice Zendaye. Vidimo kako su lica Zendaye i Toma Hollanda prepoznata s postotkom vjerojatnosti predviđanja od 95%, dok lice Chrisa Hemswortha ima nešto manji postotak od 66%.



Slika 8.8. Rezultati testiranja

8.4. Usporedba rezultata

U ovom će se dijelu usporediti dobiveni rezultati prepoznavanja lica SqueezeNet i GoogLeNet algoritmima. Na oba algoritma korištena je ista baza lica te su uspoređena lica sitih osoba.

Kod prepoznavanja jednog odabranog lica SqueezeNetom najveći postotak vjerojatnosti predviđanja postignut je kod Zendaya, za koju je algoritam utvrdio da je to 86.7% ta osoba. Jedno odabrano lice glumca Toma Hollanda prepoznato je s postotkom vjerojatnosti predviđanja od 73,9%. Postotak vjerojatnosti predviđanja jednog odabranog lica prepoznavanja lica kod glumca Chrisa Hemswortha je 84.7%. Rezultati prepoznavanja istih odabranih lica kao za SqueezeNeta, korištenjem GoogLeNet algoritma su sljedeći: lice Zendaye prepoznato je s postotkom vjerojatnosti predviđanja od 95%, Toma Hollanda također s 95% i naposljetku Chris Hewsworth ima najmanji postotak vjerojatnosti predviđanja od 66%

Tablica 8.4.1. prikazuje postotak vjerojatnosti predviđanja prepoznavanja lica za svaku od navedenih osoba kao i postotak razlike između algoritama pa je tako razlika između rezultata za Zendayu 8,3%, Toma Hollanda 21,1% i Chrisa Hewswortha 18,7%.

Tablica 8.4.1. Usporedba rezultata

| | SqueezeNet | GoogLeNet | Razlika |
|------------------------|-------------------|------------------|----------------|
| Zendaya | 86.7% | 95% | 8,3% |
| Tom Holland | 73,9%. | 95% | 21,1% |
| Chris Hemsworth | 84,7% | 66% | 18,7% |

Izvor: Vlastita izrada

Sada ćemo usporedit SqueezeNet i GoogLeNet algoritam sa prosječnom točnošću preostalih algoritama za prepoznavanje lica koji koriste duboke neuronske mreže koji su spomenuti u ovom radu, no važno je uzeti u obzir kako su prilikom testiranja tih drugih algoritama korištene druge baze lica, primjerice *Labeled Face in the Wild* baza. Prvo mjesto kao najtočniji algoritama današnjice zauzima FaceNet s točnosti od 99.63% [35], dok je drugi najtočniji DeepID verzija 3 čija je točnost nešto manja s 99.53% [36], zatim slijedi algoritam je VGGFace s 99.13% [42], DeepFace s 97.35% točaka [34] dok SqueezeNet i GoogLeNet prema rezultatima ovog istraživanja imaju nešto manju točnosti od ostalih. Slika 8.9. prikazuje matricu zabune za SqueezeNet. Uz pomoć nje možemo izračunati ukupnu točnost SqueezeNet algoritma na način da sumu svih lica na dijagonali podijelimo sa sumom svih lica. Iz toga se dolazi do zaključka da je točnost prepoznavanja lica uz SqueezeNet 48%.

| | | Confusion Matrix | | | | | | | | | | |
|--------------------|------------------|------------------|-------------|-----------------|------------------|-----------|-----------------|--------------------|-------------|---------|-------------|-------|
| Output Class | | Target Class | | | | | | | | | | |
| | | Chadwick Boseman | Chris Evans | Chris Hemsworth | Hailee Steinfeld | Paul Rudd | Rober Downey Jr | Scarlett Johansson | Tom Holland | Zendaya | Zoe Saldana | |
| Output Class | Chadwick Boseman | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100% |
| | | 6.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| | Chris Evans | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 56.7% |
| | | 2.0% | 4.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 33.3% |
| | Chris Hemsworth | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100% |
| | | 0.0% | 0.0% | 6.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| | Hailee Steinfeld | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 75.0% |
| | | 0.0% | 0.0% | 0.0% | 6.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 2.0% | 25.0% |
| | Paul Rudd | 1 | 2 | 2 | 0 | 3 | 4 | 0 | 2 | 0 | 0 | 21.4% |
| | | 2.0% | 4.0% | 4.0% | 0.0% | 6.0% | 8.0% | 0.0% | 4.0% | 0.0% | 0.0% | 78.6% |
| Rober Downey Jr | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 33.3% | |
| | 0.0% | 0.0% | 0.0% | 0.0% | 4.0% | 2.0% | 0.0% | 0.0% | 0.0% | 0.0% | 56.7% | |
| Scarlett Johansson | 0 | 1 | 0 | 2 | 0 | 0 | 5 | 0 | 3 | 2 | 38.5% | |
| | 0.0% | 2.0% | 0.0% | 4.0% | 0.0% | 0.0% | 10.0% | 0.0% | 6.0% | 4.0% | 51.5% | |
| Tom Holland | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 100% | |
| | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 4.0% | 0.0% | 0.0% | 0.0% | 0.0% | |
| Zendaya | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 40.0% | |
| | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 2.0% | 4.0% | 4.0% | 50.0% | |
| Zoe Saldana | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | NaN% | |
| | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | NaN% | |
| | | 50.0% | 40.0% | 50.0% | 50.0% | 50.0% | 20.0% | 100% | 40.0% | 40.0% | 0.0% | 48.0% |
| | | 40.0% | 50.0% | 40.0% | 40.0% | 40.0% | 30.0% | 0.0% | 50.0% | 50.0% | 100% | 52.0% |

Slika 8.9. Matrica zabune za Squeezenet

Slika 8.10. prikazuje matricu zabune za GoogLeNet uz pomoć koje je, na isti način kao i za Squeezenet izračunata točnost prepoznavanja lica uz GoogLeNet koja u ovom slučaju iznosi 70.6%.

| | | Confusion Matrix | | | | | | | | | | |
|--------------------|------------------|------------------|-------------|-----------------|------------------|-----------|-----------------|--------------------|-------------|---------|-------------|-------|
| Output Class | | Target Class | | | | | | | | | | |
| | | Chadwick Boseman | Chris Evans | Chris Hemsworth | Hailee Steinfeld | Paul Rudd | Rober Downey Jr | Scarlett Johansson | Tom Holland | Zendaya | Zoe Saldana | |
| Output Class | Chadwick Boseman | 5 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 71.4% |
| | | 9.8% | 0.0% | 0.0% | 0.0% | 2.0% | 2.0% | 0.0% | 0.0% | 0.0% | 0.0% | 28.6% |
| | Chris Evans | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 56.7% |
| | | 0.0% | 3.9% | 0.0% | 0.0% | 2.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 33.3% |
| | Chris Hemsworth | 0 | 2 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52.5% |
| | | 0.0% | 3.9% | 9.8% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 37.5% |
| | Hailee Steinfeld | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | NaN% |
| | | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | NaN% |
| | Paul Rudd | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 100% |
| | | 0.0% | 0.0% | 0.0% | 0.0% | 3.9% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Rober Downey Jr | 0 | 1 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 30.0% | |
| | 0.0% | 2.0% | 0.0% | 0.0% | 0.0% | 7.8% | 0.0% | 0.0% | 0.0% | 0.0% | 20.0% | |
| Scarlett Johansson | 0 | 0 | 0 | 3 | 0 | 0 | 5 | 0 | 1 | 0 | 55.6% | |
| | 0.0% | 0.0% | 0.0% | 5.9% | 0.0% | 0.0% | 9.8% | 0.0% | 2.0% | 0.0% | 34.4% | |
| Tom Holland | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 5 | 0 | 0 | 33.3% | |
| | 0.0% | 0.0% | 0.0% | 0.0% | 2.0% | 0.0% | 0.0% | 9.8% | 0.0% | 0.0% | 16.7% | |
| Zendaya | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 3 | 0 | 50.0% | |
| | 0.0% | 0.0% | 0.0% | 3.9% | 0.0% | 0.0% | 0.0% | 0.0% | 5.9% | 0.0% | 40.0% | |
| Zoe Saldana | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 5 | 33.3% | |
| | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 2.0% | 9.8% | 16.7% | |
| | | 100% | 0.0% | 100% | 0.0% | 0.0% | 100% | 100% | 50.0% | 100% | 70.6% | |
| | | 0.0% | 50.0% | 0.0% | 100% | 50.0% | 20.0% | 0.0% | 40.0% | 40.0% | 29.4% | |

Slika 8.10. Matrica zabune za GoogLeNet

Tablica 8.4.2. prikazuje algoritme za prepoznavanje lica koristeći duboke neuronske mreže rangirane prema njihovoj točnosti, od najtočnijeg do onog s najmanjom točnošću.

Tablica 8.4.2. Točnost ostalih algoritama za prepoznavanje lica

| Algoritam | Prosječna točnost |
|------------|-------------------|
| FaceNet | 99.63% |
| DeepID | 99.53% |
| VGGFace | 99.13% |
| Deepface | 97.35% |
| GoogLeNet | 70.6% |
| SqueezeNet | 48% |

Izvor: Vlastita izrada

9. Zaključak

Prepoznavanje lica bio je ljudske svakodnevice i smatra se jednim od najprirodnijih ljudskih vještina u društvenoj interakciji koji kod ne zahtijeva gotovo nikakav napor. Međutim, taj zadatak je za računala mnogo kompliciraniji. Dok ljudi golom okom prepoznaju o kojoj je osobi riječ računala moja proći zahtjevan proces treniranja i učenja. Međutim, tokom godina razvijeni su mnogi algoritmi za prepoznavanje lica od kojih neki daju impresivnu točnost, gotovo jednaku kao i ljudsko oko.

U ovom radu opisan je sam pojam prepoznavanje lica koje se definira kao oblik identifikacije lica mapiranjem značajka lica sa slike i usporedbe s licima pohranjenim u bazama lica. Proces prepoznavanja lica uključuje četiri osnovna koraka: detekcije lica, poravnanja lica, ekstenzije obilježja i podudaranja lica. Ono ima brojna područja primjene koje nam svakodnevno olakšavaju život, od sigurnosti pa sve do obrazovanja i financije.

Algoritmi za prepoznavanje lica koji su razvijeni devedesetih godina dvadesetog stoljeća sada se će smatrati „starijim“ algoritmima za prepoznavanje lica, te su neki od takvih algoritama Eigenface, Fisherface, Histogrami lokalnih binarnih koeficijenata te SIFT i SURF. Iako su već ti stariji algoritmi davali impresivne rezultate s daljim razvojem informacijski tehnologija i napredcima u dubokom učenju i umjetnoj inteligenciji ipak je došlo do razvoja točnijih algoritama.

Novija generacija algoritama za prepoznavanje lica koristi se dubokim neuronskim mrežama. Duboke neuronske mreže raznolike su te se konstantno razvijaju, no nužno je da ulaz u duboku neuronsku mrežu bude skup vrijednosti koje predstavljaju informacije koje će mreža ispitati, u ovom slučaju lica osoba. Neki od najpoznatijih i najkorištenijih algoritama za prepoznavanje lica koji koriste duboke neuronske mreže su: FaceNet, DeepFace, DeepID i VGGFace, no naravno da osim njih postoji još ogroman broj takvih algoritama.

Vjerujem kako nas u budućnosti očekuje širenje primjene prepoznavanja lica na mnoge druge funkcije i industrije kao i danje unaprjeđenje postojećih algoritama za prepoznavanje lica te naravno i razvitak novih te da će kroz sljedećih nekoliko godina točnost gotovo svih algoritama za prepoznavanje lica, posebice onih koji koriste duboke neuronske mreže, naveliko preći točnost ljudskog oka.

Popis literature

- [1]M. Hassaballah ,A. Saleh, „Face recognition: challenges, achievements and future directions“, IET Computer Vision, 2014, ISSN 1751-9632
- [2]A.S. Tolba, A.H. El-Baz, A.A. El-Harby „Face Recognition: A Literature Review“, International Journal of Signal Processing 2;2 2006
- [3]S. Symanovich, „What is facial recognition? How facial recognition works. Izvor: <https://us.norton.com/internetsecurity-iot-how-facial-recognition-software-works.html>, Zadnji pristup 20.5.2021
- [4]S. Z. Li i A. K. Jain, „Handbook of Face Recognition“ Springer Science & Business Media, 2005
- [5]V. Relly, P. Virgil, „Face Recognition as a Biometric Application“, Journal of Mechatronics and Robotics 2019, ARoTMM-IFTToMM, Bucharest Polytechnic University, Bucharest (CE), Romania
- [6]S. Wang, „The Application of Face Recognition System“, Proceedings of the 2021 International Conference on Social Development and Media Communication (SDMC 2021)
- [7]G. Jensen Grant. „How Banks and Financial Institutions Use Face Recognition to Protect People, Property, and Assets“ Izvor: <https://www.briefcam.com/resources/blog/how-banks-and-financial-institutions-use-face-recognition-to-protect-people-property-and-assets/>, Zadnji pristup 20.05.2022
- [8]M. Solomun, M. Meena, J. Kaur J, „Challenges in face recognition systems“,International journal of research and analytical reviews, vol. 6, svibanj-lipanj 2019
- [9]M. Hrga. 2018. Računalni vid. Veleučilište u Šibeniku, Stručni rad.
- [10] P. Viola, M.J. Jones, "Robust Real-Time Face Detection", International Journal of Computer Vision, vol. 57, str. 137–154, svibanj 2004., doi: 10.1023/B:VISI.0000013087.49260.fb
- [11]N. Dalal, B. Triggs, „Histograms of oriented gradients for human detection“, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, SAD, lipanj 2005., str. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

- [12]O. Deniz, G. Bueno, J. Salido i F. De la Tore, „Face recognition with histograms of oriented gradients“. VISAPP 2010 - International Conference on Computer Vision Theory and Applications, 2010. 10.5220/0002820503390344
- [13]L. Cerna, G. Cámara-Chávez, D. Menotti, „Face Detection: Histogram of Oriented Gradients and Bag of Feature Method“, 2013. Izvor: https://www.researchgate.net/publication/342886529_Face_Detection_Histogram_of_Oriented_Gradients_and_Bag_of_Feature_Method, Posljedni pristup 10.06.2022.
- [14]K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.
- [15]J. Brownlee, „How to Perform Face Detection with Deep Learning“ izvor <https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/> , zadnji pristup 11.06.2022.
- [16]S. S. Farfade, M. J. Saberain i L.J.Li, „Multi-view Face Detection Using Deep Convolutional Neural Networks“, ICMR '15: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Lipanj 2015, str 643–650
- [17]V. Vapnik, C. Cortes C. „Support-Vector Networks“, *Mach Learn* **20**, 273–297 (1995). <https://doi.org/10.1007/BF00994018>
- [18]R. Ghandi, „Support Vector Machine — Introduction to Machine Learning Algorithms“ izvor <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>, zadnji pristup 16.06.2022.
- [19]P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720, July 1997, doi: 10.1109/34.598228.
- [20]M. Anggo, L.Arapu, „Face Recognition Using Fisherface Method“, Journal of Physics: Conference Series 1028 012119, 2018
- [21]M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991, pp. 586-591, doi: 10.1109/CVPR.1991.139758.
- [22]M. Çarıkçı, F. Özen. “A Face Recognition System Based on Eigenfaces Method“, Procedia Technology 1:118–123, 2012. doi: 10.1016/j.protcy.2012.02.023

- [23]K. Salton do Prado K. „Face Recognition: Understanding LBPH Algorithm“ izvor <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>, zadnji pristup 17.06.2022.
- [24]OpenCV. „Face Recognition with OpenCV“ izvor [s https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms), zadnji pristup 17.06.2022.
- [25]R. Mare, „Understanding Facial Recognition Using Local Binary Pattern Histogram (LBPH) Algorithm“ izvor <https://www.section.io/engineering-education/understanding-facial-recognition-using-local-binary-pattern-histogram-algorithm/>, zadni pristup 17.06.2022.
- [26]D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, pp. 1150-1157 vol.2, doi: 10.1109/ICCV.1999.790410.
- [27]D. Tyagi, „Introduction to SIFT(Scale Invariant Feature Transform)“, izvor <https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>, zadnji pristup 17.06.2022.
- [28]H. Bay, T. Tuytelaars, L. Van Gool. „SURF: Speeded Up Robust Features“ Computer Vision and Image Understanding, vol.110, lipanj 2008, str. 346-359 doi: 10.1016/j.cviu.2007.09.014
- [29]K. T. Teoh, R. C., S.Z.M. Naziri, R. Hussin, M.N.M. Isa i M.S.S.M. Basir. „Face Recognition and Identification using Deep Learning Approach“, Journal of Physics: Conference Series, 2006. doi: 10.1088/1742-6596/1755/1/012006
- [30]V. Sze, Y.Chen, T, Yang, J. Emer. “Efficient Processing of Deep Neural Networks: A Tutorial and Survey.” Proceedings of the IEEE 105, no. 12 (2017): 2295–2329. doi:10.1109/JPROC.2017.2761740.
- [31]A. Thakur,. “Fundamentals of Neural Networks.” International Journal for Research in Applied Science and Engineering Technology 9, no. VIII (2021): 407–26. doi:10.22214/IJRASET.2021.37362.
- [32]T. Noor, „Deep Learning For Beginners“ izvor: <https://towardsdatascience.com/intro-to-deep-learning-d5caceedcf85#.p61qh0vz8>, zadnji pristup 14.07.2022.
- [33]K.J. Cios. „Deep Neural Networks - A Brief History“, Studies in Computational Intelligence, 2018. doi: 10.1007/978-3-319-67946-4_7

- [34]Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.
- [35]F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.
- [36]Y. Sun, X. Wang and X. Tang, "Deep Learning Face Representation from Predicting 10,000 Classes," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891-1898, doi: 10.1109/CVPR.2014.244.
- [37]Y.Sun, L. Wang, X. Tang. "Deep Learning Face Representation by Joint Identification-Verification", Proceedings of the 27th International Conference on Neural Information Processing Systems – vol. 2, 2014 doi: 10.48550/arXiv.1406.4773
- [38]Y.Sun, D. Liang, X. Wang, X. Tang, "DeepID3: Face Recognition with Very Deep Neural Networks." *ArXiv* abs/1502.00873 (2015): n. pag.
- [39] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.
- [40]F. Iandola, S.Han, M. Moskewicz, K. Ashraf, W. Dally, K. Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size." *ArXiv* abs/1602.07360 (2016): n. pag.
- [41]O. M. Parkhi, A. Vedaldi, A. Zisserman. "Deep Face Recognition", Proceedings of the British Machine Vision Conference (BMVC), pages 41.1-41.12. BMVA Press, 2015. doi: 10.5244/C.29.41
- [42]Q. Cao, L. Shen, W. Xie, M. Omka Parkhi, A. Zisserman. "VGGFace2: A Dataset for Recognising Faces across Pose and Age", 13th IEEE International Conference on Automatic Face & Gesture Recognition, 2018. doi: 10.48550/arXiv.1710.08092

Popis slika

| | |
|--|----|
| Slika 2.1. Koraci prilikom prepoznavanja lica. Izvor: https://www.academia.edu/40549627/Handbook_of_Face_Recognition , Pristup: 16.06.2022. | 3 |
| Slika 4.1. Moguće hiperravnine. izvor https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 , zadnji pristup 16.06.2022. | 13 |
| Slika 5.1. Primjer izrade lokalnih binarnih uzoraka. izvor https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b , zadnji pristup 17.06.2022. | 18 |
| Slika 5.2. Primjer koncept kružnog susjedstva. izvor https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b , zadnji pristup 17.06.2022 | 18 |
| Slika 5.3. Generiranje opisa ključnih točaka. izvor https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40 , zadnji pristup 17.06.2022. | 20 |
| Slika 5.4. Diskretizacija i rezanje. izvor https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40 , zadnji pristup 17.06.2022. | 21 |
| Slika 5.5. Fiksiranje reproducibilne orijentacije. izvor https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40 , zadnji pristup 17.06.2022. | 21 |
| Slika 6.1. Duboko učenje u kontekstu umjetne inteligencije. Izvor: https://iopscience.iop.org/article/10.1088/1742-6596/1755/1/012006/meta , zadnji pristup 22.06.2022. | 22 |
| Slika 7.1. Arhitektura DeepFace algoritma. Izvor: https://ieeexplore.ieee.org/document/6909616 , zadnji pristup 22.06.2022. | 27 |
| Slika 7.3. Struktura neuronske mreže. Izvor: https://ieeexplore.ieee.org/document/6909640 , zadnji pristup 22.06.2022 | 31 |
| Slika 7.4. GoogLeNet arhitektura. Izvor: https://ieeexplore.ieee.org/document/7298594 , zadnji pristup 23.06.2022. | 33 |
| Slika 7.5. 18 slojeva SqueezeNet modela. Izvor: https://www.semanticscholar.org/paper/Deep-Face-Recognition-Parkhi-Vedaldi/162ea969d1929ed180cc6de9f0bf116993ff6e06 , zadnji pristup 23.06.2022. | 34 |
| Slika 8.1. Korištena baza lica. Izvor: vlastita snimka zaslona | 38 |
| Slika 8.2. Arhitektura SqueezeNet. Izvor: Matlab | 40 |
| Slika 8.3. Treniranje SqueezeNeta. Izvor: vlastita snimka zaslona | 41 |
| Slika 8.4. Rezultati testiranja. Izvor: vlastita snimka zaslona | 42 |
| Slika 8.5. Add-On za instalaciju GoogLeNeta. Izvor: Matlab | 42 |
| Slika 8.6. Arhitektura GoogLeNeta. Izvor: Matlab | 43 |
| Slika 8.7. Treniranje GoogLeNeta. Izvor: vlastita snimka zaslona | 45 |
| Slika 8.8. Rezultati testiranja. Izvor: vlastita snimka zaslona | 45 |

| | |
|--|----|
| Slika 8.9. Matrica zabune za Squeezenet. Izvor: vlastita snimka zaslona..... | 48 |
| Slika 8.10. Matrica zabune za GoogLeNet. Izvor: vlastita snimka zaslona..... | 48 |



IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Tena Benček (*ime i prezime*) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Prepoznavanje lica koristeći duboke neuronske mreže (*upisati naslov*) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(*upisati ime i prezime*)

Tena Benček
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Tena Benček (*ime i prezime*) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Prepoznavanje lica koristeći duboke neuronske mreže (*upisati naslov*) čiji sam autor/ica.

Student/ica:
(*upisati ime i prezime*)

Tena Benček
(vlastoručni potpis)