Algoritmi za prepoznavanje lica

Baraba, Martina

Sveučilište u Rijeci
Fakultet za matematiku

Diplomski studij Diskretna matematika i primjene

Martina Baraba

Algoritmi za prepoznavanje lica

Diplomski rad
Rijeka, rujan 2024.

Sveučilište u Rijeci
Fakultet za matematiku

Diplomski studij Diskretna matematika i primjene

Martina Baraba

Algoritmi za prepoznavanje lica

Mentor: doc. dr. sc. Sanda Bujačić Babić

Diplomski rad
Rijeka, rujan 2024.

Sažetak

Prepoznavanje lica je metoda koja se koristi za identifikaciju ili autentifikaciju osobe na temelju analize karakteristika lica, a sustavi za prepoznavanje lica se primjenjuju za prepoznavanje osoba na fotografijama, videozapisima, sigurnosnim kamerama i slično ili uživo. Početak razvoja algoritama za prepoznavanje lica bio je sredinom 60-ih godina 20. stoljeća, ali tek sa snažnijim razvojem tehnologije i algoritama strojnog učenja problem prepoznavanja lica se uspješno rješava. Osnovna podjela algoritama za prepoznavanje lica je na algoritme za prepoznavanje $2D$ i $3D$ lica. Algoritmi koji koriste $2D$ lica zapravo koriste $2D$ digitalne slike za prepoznavanje uspoređujući sliku lica s bazom podataka prethodno snimljenih lica, dok se algoritmi koji koriste $3D$ model lica oslanjaju na $3D$ geometriju lica i u obzir uzimaju značajke poput dubine ili zakrivljenosti crta lica što nije moguće obuhvatiti na $2D$ slici. Matematičkim modelima izdvajaju se bitne značajke na licu i uspoređuju se s značajkama poznatih lica u bazi. Najpoznatiji algoritmi za prepoznavanje $2D$ lica su Eigenface algoritam, Fisherfaces, metoda potpornih vektora i skriveni Markovljev model. Ograničenja algoritama koji koriste $2D$ model lica su promjene u fizičkom izgledu, orijentaciji glave i promjene u osvjetljenju. Te nedostatke pokušavaju nadomjestiti algoritmi za prepoznavanje koji koriste $3D$ model lica, a neki od najčešće korištenih su: prepoznavanje $3D$ lica bez rekonstrukcije lica, morfabilni problem i konvolucijske neuronske mreže. Jedan od značajnijih nedostataka algoritama za prepoznavanje $3D$ lica je potreba za puno većom bazom podataka za treniranje modela i kvalitetnija oprema prilikom prikupljanja fotografija što je zahtjevno za realizaciju. U diplomskom radu opisana je povijest razvoja algoritama za prepoznavanje lica, navedene su osnovne značajke svih algoritama za prepoznavanje lica, problemi s kojima se algoritmi susreću i njihova primjena. Opisana je matematička pozadina svakog od navedenih algoritama, njihove prednosti i nedostaci. Na kraju rada, provedena je analiza glavnih komponenata na skupu slika te je nad istim skupom provedena programska realizacija metode potpornih vektora i konvolucijskih neuronskih mreža te je uspoređena dobivena točnost predikcija.

Ključne riječi: detekcija lica, analiza lica, algoritmi za prepoznavanje $2D$ i $3D$ lica, nadzirano učenje, analiza glavnih komponenti, Eigenface algoritam, metoda potpornih vektora, $3D$ prepoznavanje lica bez rekonstrukcije lica, morfabilni model, neuronske mreže, konvolucijske neuronske mreže.

Sadržaj

1 Uvod

Prepoznavanje lica je način identifikacije ili potvrde identiteta pojedinca s pomoću njegovog lica. Sustavi za prepoznavanje lica koriste se za identifikaciju ljudi na fotografijama, videozapisima i u stvarnom vremenu te se sve više koriste i razvijaju novi algoritmi u svrhu povećanja efikasnosti i sigurnosti. Iako se već desetljećima pokušavalo automatski detektirati i prepoznati lice, tek sa snažnim razvojem tehnologije i algoritama strojnog učenja taj se problem uspješnije rješava. Prepoznavanje lica je biometrijska[1] metoda koja uz pomoć tehnologije (najčešće algoritmima umjetne inteligencije i strojnog učenja) identificira ljudsko lice tako da ga locira, odnosno pronalazi, te ga skenira s fotografije, videozapisa ili u realnom vremenu nakon čega uspoređuje neke ključne crte lica s bazom lica kako bi pronašao podudaranje. Prepoznavanje lica pomaže u potvrdi identiteta osobe, iako se još uvijek ne koristi kao standardni oblik autentifikacije, zbog sve snažnijeg razvoja preciznih algoritama, postaje dio postupka multifaktorske autentifikacije[2] prije lozinki ili skeniranja otiska prsta. Popularnost sustava prepoznavanja lica jest u tome što je brži i praktičniji način provjere u odnosu na druge biometrijske tehnologije, sigurniji je način identifikacije od korištenja e-maila i lozinke te je danas dovoljno rasprostranjen da ga je relativno jednostavno integrirati u većinu sigurnosnih softvera. No, treba istaknuti da, iako su današnji algoritmi za prepoznavanje lica precizni, nisu u potpunosti točni te su svi oslonjeni na pripadnu bazu podataka poznatih lica pa time ovise o veličini baze i brzini pretraživanja baze. Također, sve prisutniji problem korištenja ovakve tehnologije je zaštita privatnosti pojedinca s obzirom na to da se algoritam može provesti bez znanja i pristanka osobe da se njeno lice skenira [13, 14, 44].

[1] Biometrija - skup automatiziranih metoda za jedinstveno prepoznavanje ljudi temeljenih na jednoj ili više fizičkih (otisak prsta, rožnica oka, prepoznavanje lica i sl.) i ponašajnih karakteristika (rukopis, hod i sl.).

[2] Multifaktorska (višefaktorska) autentifikacija, MFA - sigurnosni proces koji zahtjeva dva ili više faktora autentifikacije kako bi se potvrdio identitet korisnika. Faktori mogu uključivati poznavanje lozinke, korištenje tokena ili biometriju, a cilj uvođenja više faktora je povećanje sigurnosti.

2 Algoritmi za prepoznavanje lica

Algoritam za prepoznavanje lica je biometrijski alat koji na temelju lica može prepoznati osobu. Kao svi biometrijski procesi, algoritmi za prepoznavanje lica mogu izvršiti dvije različite funkcije: identifikaciju i autentifikaciju. Cilj identifikacije je identificirati osobu unutar skupine pojedinaca, unutar određenog područja, slike ili baze podataka. Sustav obrađuje svako snimljeno lice kako bi se generirao predložak i zatim provjerava odgovara li ono osobi koja je poznata sustavu. Autentifikacija je proces utvrđivanja je li osoba zaista ona za koju se predstavlja. Sustav uspoređuje unaprijed snimljenu sliku lica osobe s licem osobe za koju se predstavlja. Postupak se naziva i 1-na-1 verifikacija. Identifikacija se odvija prilikom postavljanja računa novog korisnika, klijenta ili zaposlenika u kojem ta osoba daje svoje osobne podatke kako bi se identificirala i zatim potvrđuje svoj identitet. Provjera identiteta tada može uključivati skeniranje lica ili neke druge korake, a odvija se svaki put kad korisnik pristupi računu ili usluzi. Autentifikacija je drugi korak procesa kako bi se korisnik povezao s prethodno unesenim informacijama i provodi se kako bi se osiguralo da je on doista osoba za koju se predstavlja [21].

Svaki algoritam za prepoznavanje lica sastoji se od slike lica osobe koja se želi prepoznati i baze podataka koju čine sve slike lica s kojima se lice osobe uspoređuje. Baze podataka mogu biti vrlo raznolike - od baza podataka koje se sastoje od digitaliziranih fotografija s osobnih iskaznica do baza podataka koje čine fotografije preuzete s društvenih mreža. Iako se algoritmi međusobno razlikuju, sve ih karakteriziraju sljedeći koraci:

1. *detekcija lica* - lociranje i izdvajanje lica sa slike ili videozapisa unutar sustava,

2. *analiza lica* - očitavanje geometrije lica te određivanje bitnih značajki za prepoznavanje lica (utvrđivanje udaljenosti između očiju, detekcija jagodičnih kostiju, utvrđivanje udaljenosti od čela do brade, detekcija konture usana,...). Glavne značajke lica nazivaju se orijentiri i oni su ključni za razlikovanje lica. Orijentiri se pretvaraju u skup digitalnih informacija koji se naziva digitalni otisak lica. Otisak lica je jedinstven za svaku osobu,

3. *prepoznavanje lica* - digitalni otisak lica se uspoređuje s relevantnom bazom poznatih digitalnih otisaka lica. U ovoj fazi slika prolazi kroz nekoliko slojeva obrade kako bi se osigurala točnost, a algoritmi moraju uzeti u obzir razlike u osvjetljenju, izrazu lica i kutovima slikanja. Ukoliko se otisak lica podudara s otiskom lica neke od pohranjenih slika u bazi, s obzirom na definiranu razinu preciznosti, tada algoritam donosi odluku o podudaranju [14].

Prepoznavanje lica je za računalo zahtjevan problem jer, dok ljudsko oko lako razaznaje lice na slici, računalo to ne može učiniti instinktivno. S tim ciljem je napisan veliki broj računalnih algoritama koji su u mogućnosti "prepoznati" ljudsko lice i to na temelju detaljne analize piksela na slici. Naime, ova vrsta algoritama traži specifične uzorke i karakteristike u rasporedu piksela na slici, kao što su razmak između očiju, oblik nosa i usta, kako bi se identificiralo lice. Procesi ovakvog tipa zahtijevaju složene algoritme koji se mogu realizirati u što kraćem vremenskom roku iako je njihova zadaća nerijetko vremenski

vrlo zahtjevna. Također, različite dimenzije slika, korištenje boja i uvjeti osvjetljenja dodatno kompliciraju problem detekcije lica jer računalo mora znati raspoznati te varijacije samo na temelju analize piksela. Računalu je čak i početni problem, odnosno utvrđivanje nalazi li se lice na slici ili ne, vrlo zahtijevan zadatak, a to se najčešće postiže analizom obrazaca i uprosječivanjem karakteristika koje su specifične za lica [8].

Korištene tehnike prepoznavanja lica temelje se na procijenjenom podudaranju između slike lica i slika pohranjenih u bazi pa se algoritmi za prepoznavanje lica smatraju vjerojatnosnim. Kako bi algoritmi što bolje funkcionirali ključne značajke koje trebaju biti zadovoljene su:

- odabir pogodne baze podataka - točnost algoritma ovisi o odabiru baze na kojoj je treniran model. Baza mora biti statistički reprezentativna te također mora sadržavati slike s varijacijama u osvjetljenju, kutovima slikanja i izrazu lica te moraju biti uvedene slike različite rezolucije. Statistički reprezentativna baza podataka točno odražava karakteristike i varijacije populacije iz koje je promatrani uzorak. Podaci u bazi su raznoliki i obuhvaćaju različite segmente populacije kako bi se rezultati mogli generalizirati na cjelokupnu populaciju. Reprezentativnost se postiže pažljivim dizajnom uzorka i metodama prikupljanja podataka koje minimiziraju pristranost i osiguravaju da svi dijelovi populacije imaju proporcionalan udio u bazi podataka. Iz navedenih razloga kvalitetnu bazu podataka je teško izraditi te je potrebno vrijeme i prikladna tehnologija. Stoga, sam odabir i izrada baze podataka je prvi problem na koji istraživači nailaze prilikom razvoja algoritama za prepoznavanje lica,

- sigurnost i privatnost - korisnički podaci (digitalni otisci lica) moraju biti šifrirani kako bi se zaštitili u slučaju krađe podataka. Algoritmi za prepoznavanje lica u bazi podataka imaju pohranjene slike lica s pridruženim identitetom osobe na slici koji se šifrira kako u slučaju hakerskih napada i neovlaštenih preuzimanja baze podataka podaci ne bi bili zloupotrebljeni za krađu identiteta, praćenje, identificiranje i slično,

- točnost algoritma - korišteni algoritmi moraju imati dovoljnu razinu točnosti i preciznosti, a posebno se treba obratiti pozornost na stope lažnog prihvaćanja (dvije slike se lažno podudaraju kao jednake) i stope lažnog odbijanja (algoritam ne prepoznaje poznato lice, već ga klasificira kao nepoznato). U praksi, želi se postići da je stopa lažnog prihvaćanja niska, a lažnog odbacivanja prihvatljivo visoka jer su posljedice lažnog prihvaćanja puno ozbiljnije nego posljedice lažnog odbijanja. Naime, želi se postići niska stopa lažnog prihvaćanja jer je tada malo vjerojatno da će neovlaštena osoba biti pogrešno identificirana kao ovlaštena i imati pristup sustavu, što je ključna mjera za sprječavanje sigurnosnih propusta i zaštitu povjerljivih informacija. Visoko sigurnosni sustavi puno su usmjereniji na sprječavanje neovlaštenih pristupa nego na osiguravanje da svaki ovlašteni pristup bude uspješan, stoga su više stope lažnog odbijanja prihvatljive. Dok lažno odbijanje može biti naporno za korisnike i zahtijevati dodatne provjere verifikacije, lažno prihvaćanje može dovesti do velikih sigurnosnih propusta, stoga je prioritet smanjiti stopu lažnog prihvaćanja,

- transparentnost i etičnost - algoritmi ne koriste neetičke postupke koji uključuju

stvaranje baze podataka neovlaštenim preuzimanjem slika s društvenih mreža radi treniranja algoritma i ne obuhvaćaju druge postupke kršenja privatnosti [44].

2.1 Poznati problemi u detekciji lica

Mnogobrojni su problemi koji se javljaju prilikom detekcije lica. Najveći problemi vezani su uz točnost korištenih algoritama i sigurnost te zaštitu privatnosti. Veliki problem predstavljaju prevarantski napadi, kao što su korištenje slika i videozapisa kako bi se obmanuli algoritmi i kako bi se neovlašteno pristupilo sigurnosnim sustavima. Facebook i Google suočili su se s ozbiljnim tužbama zbog narušavanja privatnosti povezanih s njihovim tehnologijama za prepoznavanje lica. 2020. godine savezna država Texas, SAD je tužila Facebook jer je kompanija kršila zakon o privatnosti biometrijskih podataka time što je automatski označavala lica korisnika na slikama bez njihovog pristanka. Kako bi se oslobodio optužbi, Facebook je pristao platiti kaznu od 650 milijuna dolara [36]. Google se borio s optužbama radi neovlaštenog korištenja vlastite tehnologije za prepoznavanje lica unutar usluge Google Photos za analiziranje fotografija, kreiranje i pohranjivanje predložaka korisničkih lica bez ikakvog informiranja korisnika. Google je platio kaznu oko 100 milijuna dolara čime je svaki korisnik koji je sudjelovao u grupnoj tužbi dobio oko 96$ isplate [7]. Sve veća zabrinutost javnosti javlja se zbog korištenja tehnologije prepoznavanja lica za nadzor, dok još uvijek ne postoje jasno definirane granice koliko daleko u nadziranju građana se tehnologija smije koristiti (je li to nadziranje građana samo na javnim površinama ili i unutar privatnih posjeda kao što je parking neke privatne tvrtke, prostor marine, eksterijeri hotela, itd.). Postavlja se pitanje koliko je tehnologija točna i precizna ako bi se koristila u sudskim procesima prilikom osuđivanja kriminalaca. Jedan od problema je i u tome što je dokazano da je tehnologija puno manje efikasna u identificiranju drugih rasa. Naime, pokazalo se da su algoritmi precizniji kod identifikacije bijelih muškaraca, nego drugih rasa ili općenito kod osoba ženskog spola [13].

2.2 Povijesni razvoj algoritama za prepoznavanje lica

Početak istraživanja i razvoja algoritama za prepoznavanje lica započeo je između 1964. i 1966. godine kada se je W.W. Bledsoe[3] sa svojim timom u svojem radu usredotočio na razvoj programa kojim bi računalo moglo prepoznati ljudsko lice. Razvili su sustav mjerenja za fotografiju lica u kojem bi čovjek koristeći grafički RAND tablet[4] odredio koordinate crta lica i nekih temeljnih karakteristika kao što su središte zjenica oka, unutarnji i vanjski kutevi očiju, položaj nosa i vrh linija kose (Slika 1, [35]), a zatim su te koordinate korištene za izračunavanje 20 pojedinačnih udaljenosti uključujući širinu usta i očiju. Zapis koji povezuje ime s izmjerenim brojčanim vrijednostima bio je pohranjen u bazu podataka.

[3]Woodrow Wilson Bledsoe - američki matematičar, informatičar i pedagog poznat po svom doprinosu u prepoznavanju uzoraka, lica i jedan od začetnika umjetne inteligencije.

[4]RAND tablet - računalni uređaj, preteča današnjeg tableta, osmišljen 1960-ih godina koji detektiranjem pokreta olovke po plohi tableta, tekst ili crtež se pretvara u digitalni zapis i prikazuje na računalu. Uređaj je bio površine oko 10 inča.

Slika 1: Podjela lica na temeljne karakteristike i njihovo označavanje

Sljedeća faza procesa uključuje određivanje udaljenosti ključnih točaka na nepoznatom licu nakon čega bi računalo usporedilo dobivene udaljenosti s udaljenostima zapisa u bazi podataka, izračunalo bi razliku i kao rezultat dohvatilo onu fotografiju iz baze za koju su razlike bile najmanje. S obzirom da je čovjek prvo trebao odrediti koordinate crta lica kako bi računalo moglo izračunati razlike u udaljenostima između fotografija, projekt je dobio naziv *"čovjek stroj"* (eng. *"man-machine"*). Takav sustav prepoznavanja lica nije bio brz za današnje standarde - čovjek je mogao odrediti koordinate crta lica za otprilike 40 fotografija unutar jednog sata te je sam Bledsoe uvidio da mnogi čimbenici uvelike utječu na sposobnost računala da točno prepozna jednu osobu na različitim fotografijama. Naime, ako se osoba fotografirala pod vrlo različitim kutevima, u različitoj životnoj dobi, s različitim izrazom lica ili različitim osvjetljenjem, to je uvelike utjecalo na rezultat i algoritam je mogao dati netočne rezultate budući koristi pristup prepoznavanja lica temeljen na udaljenostima za klasificiranje dva lica kao jednaka. Unatoč tome razvoj algoritma za prepoznavanje lice je privukao pozornost, posebno sigurnosnih agencija i jedna od neimenovanih obavještajnih agencija nastavila je financirati Bledsoeovo kontinuirano istraživanje, no ono nikad nije objavljeno [35].

Nadovezujući se na Bledsoeov rad, A. J. Goldstein, L.D. Harmon i A.B. Lesk 1971. godine u svom radu [3] proširuju broj karakteristika lica koje se promatraju prilikom prepoznavanja uvodeći 21 specifičan, subjektivan marker kao što su boja kose i debljina usnica kako bi automatizirali prepoznavanje. Iako se time točnost poboljšala, stvarna biometrija lica se još uvijek morala računati ručno što je bilo iznimno zahtjevno. Godine 1977. T. Kanade je izradio potpuno funkcionalnu aplikaciju za prepoznavanje lica [13]. Aplikacija bi locirala anatomske značajke lica kao što je brada i izračunao bi se omjer udaljenosti između crta lica bez ljudske intervencije za razliku od ranijih projekata. Iako se kasnije otkrilo da sustav nije uvijek mogao pouzdano identificirati crte lica, interes za razvojem algoritama je rastao i od tada se algoritmi za prepoznavanje $2D$ lica intenzivno proučavaju [13].

1988. godine L. Sirovich i M. Kirby počeli su primjenjivati linearnu algebru za rješavanje problema prepoznavanja lica. Njihov cilj bio je sliku lica prikazati niskodimenzionalnim prikazom. Pokazali su da se analizom značajki na zbirci slika lica može formirati skup

osnovnih značajki za promatranje te da je potrebno manje od stotinu vrijednosti kako bi se točno kodirala normalizirana slika lica [37]. Ovakav pristup kasnije se razvio u Eigenface algoritam za prepoznavanje lica. Tri godine kasnije M. Turk i A. Pentland proširuju Eigenface pristup otkrivši kako detektirati i identificirati ljudsko lice unutar slike koja sadrži i druge objekte u gotovo stvarnom vremenu (do tada su se najčešće kao baza podataka koristili fotografski portreti ljudskog lica). Lociranje lica unutar slike postalo je popularno s razvojem Analize glavnih komponenti (eng. *Principal Component Analysis, PCA*). Algoritam bi usporedio karakteristike lica sa slike s karakteristikama lica pohranjenih u bazi. Njihov je pristup tipični predstavnik algoritama koji prepoznaju lica na osnovu dvodimenzionalnog predloška jer su u svojem radu koristili činjenicu da su lica normalno uspravna na slici i stoga se mogu opisati kao mali skup 2D karakterističnih prikaza. Nedostatak svih istraživanja bio je što su korištene male baze podataka te su postojeće baze bile usklađene s istraživanjem. Također, većina procjena razvijenih algoritama nije slijedila standardni protokol testiranja koji je uključivao zasebne skupove slika za treniranje algoritama i testiranje [32].

Agencija za napredne obrambene istraživačke projekte (DARPA[5]) i američki Nacionalni institut za standarde i tehnologiju (NIST[6]) 1993. godine pokrenuli su program tehnologije prepoznavanja lica (eng. *Facial Recognition Tehnology, FERET*). Cilj projekta bio je stvoriti veliki, automatski sustav za prepoznavanje lica koji bi se koristio u obavještajne, sigurnosne i policijske svrhe, ali i sustav koji bi potaknuo razvoj komercijalnog tržišta za prepoznavanje lica. Želja je bila unaprijediti područje tehnologije prepoznavanja lica stvaranjem zajedničke baze podataka sa slikama lica koje bi istraživači mogli koristiti u svome radu i prilikom testiranja algoritama. Prije početka programa, većina istraživača prikupila je i koristila vlastitu bazu podataka koja je bila usklađena s njihovim istraživanjem (način i uvjeti u kojima su slikane slike prilagođen je istraživanju) pa su nerijetko te baze podataka bile male (oko 50 slika) i algoritmi su isključivo na njima bili dosta precizni. Tri baze su se izdvajale po broju prikupljenih slika:

- A. Pentland s Massachusetts Intitute of Tehnology (MIT) stvorio je bazu podataka sa 7500 slika koje je prikupio u visoko kontroliranom okruženju - s kontroliranim osvjetljenjem, s fiksiranim očima na istoj lokaciji i sve slike su bile slikane kao frontalni pogledi,
- J. Wilder sa Sveučilišta Rutgers u New Jerseyu sastavio je bazu podataka s 250 slika,
- C. Malsburg sa Sveučilišta Južna Kalifornije (USC) koristio je bazu od 100 slika koje su bile unaprijed definirane i standardizirane veličine i osvjetljenja i za razliku od prethodnih baza, uključivale su rotaciju glave [12].

Nepostojanje zajedničke baze podataka onemogućavalo je usporedbu rezultata istraživanja prepoznavanja lica i usporedbu između različitih algoritama jer u radovima nisu korištene iste slike, niti su one bile slikane u jednakim uvjetima i s jednakim prikazima lica. Projekt prepoznavanja lica se odvio u tri faze od kojih je svaka trajala oko godinu dana.

[5]eng. Defense Advanced Research Projects Agency
[6]eng. National Institute of Standards and Technology

Prva faza projekta bila je usmjerena na prikupljanje slika lica i kreiranje baze podataka. Slike su bile slikane 35-mm kamerom, a slike svakog sudionika u projektu su uključivale frontalni pogled, lijevi i desni profil, poluprofil i četvrt profil (ukupno je od 5 do 11 slika svakog sudionika pohranjeno u bazu) što prikazuje Slika 2, [46]. Slikanje se odvijalo u istom fizičkom okruženju prilikom svake sesije kako bi sve slike bile standardizirane, s eventualno malim odstupanjima između slika koje su prikupljene na različite datume zbog postavljanja opreme. Pet organizacija je dobilo tako kreiranu bazu podataka na testiranje i uspoređivanje putem svojih algoritama za prepoznavanje lica, a kao najbolji algoritmi istaknuli su se algoritmi razvijeni na MIT-u i USC-u. Druga faza projekta je bila usmjerena na nadopunjavanje postojeće baze s novim kompletima slika, ali naglasak kod druge faze je bio na prikupljanju duplih skupova slika (slika osoba koje su sudjelovale i u prvoj fazi kako bi se mogle usporediti razlike i performanse algoritama s obzirom na vremenski odmak). Tijekom treće faze dodan je još određeni broj kompleta slika u bazu podataka te je cilj ove faze bio izmjeriti napredak algoritma od procjene u prvoj fazi te evaluacija svih algoritama. Cilj je bio istaknuti prednosti i slabosti svakog algoritma i odrediti buduće ciljeve istraživanja. Uz stvaranje javno dostupne standardne baze podataka, evaluacija algoritama smatra se najvećom vrijednosti ovog projekta [12].

Slika 2: Uzorak slika iz FERET baze podataka

Tijekom sve tri faze projekta prikupljeno je oko 1564 kompleta slika 1199 osoba, od čega je 365 duplih kompleta slika, što je činilo bazu s ukupno 14126 slika. Implementirana su dva algoritma za prepoznavanje lica - PCA i korelacijski algoritam koji za cilj ima usporediti karakteristične točke lica s poznatim licima u bazi podataka koristeći statističku metodu korelacije između slika, odnosno uspoređujući njihov intenzitet piksela. 2003. godine DARPA je objavila 24-bitnu verziju slika u boji visoke rezolucije u FERET bazi podataka. Snažnim razvojem strojnog učenja i neuronskih mreža, FERET baza podataka pokazala se kao korisna baza za treniranje algoritama za prepoznavanje lica [12].

U drugoj polovici 90-ih godina 20.-tog stoljeća uredi za motorna vozila Zapadne Virginije i Novog Meksika postali su prvi uredi koji su koristili automatizirani sustav za prepoznavanje lica kako bi spriječili izdavanje vozačke dozvole koristeći lažni identitet. Njihova baza podataka koristila je postojeću bazu podataka fotografija za digitalne osobne iskaznice. To je bilo jedno od prvih velikih tržišta u kojima je prepoznavanje lica građanima

predstavljeno kao standardna metoda identifikacije [13]. C. Malsburg i njegov istraživački tim sa Sveučilišta Bochum u Njemačkoj razvili su Bochumov sustav 1997. godine, koji je koristio Gaborov filter[7]. Gaborov filter snimio bi crte lica i odredio mrežnu strukturu lica koja se sastojala od međusobno povezanih točaka koje mapiraju značajne značajke na licu [15]. Uspostavljanjem mreže točaka, sustav je mogao učinkovito povezati različite značajke lica na način koji bi očuvao prostorne odnose između njih. Metoda detekcije lica koju je razvio Malsburg nadmašila je većinu drugih sustava za detekciju lica na tadašnjem tržištu. Softver je bio dovoljno robustan da izvrši identifikaciju iz nesavršenih prikaza lica, a bio je dobro prilagođen i preprekama identifikacije kao što su brkovi ili brada, naočale ili promjena frizure [13]. Prepoznavanje lica u stvarnom vremenu na video snimkama postalo je moguće 2001. godine. Jedan od prvih javnih događaja na kojima je bila testirana tehnologija prepoznavanja lica bio je SuperBowl 2002. godine. Testiranje se odvilo bez znanja građana, i iako se otkrilo nekoliko sitnih kriminalaca, test se smatrao neuspješnim zbog lažno pozitivnih identifikacija i oštre kritike javnosti zbog narušavanja privatnosti [39]. Tehnološko ograničenje koje je došlo do izražaja je to što prepoznavanje lica nije još uvijek dobro funkcioniralo u velikim gužvama i prilikom kretanja osoba koje ne gledaju direktno u kameru. Također, javila se potreba za velikim bazama podataka koje više nije bilo moguće prikupljati fotografiranjem pojedinaca, stoga su sredinom 2000-ih godina istraživači počeli raditi baze podataka putem web pretraživanja bez brige o pristanku pojedinca. Veliki napredak u prihvaćanju algoritama za prepoznavanje lica u javnosti dogodio se 2010. godine kada je Facebook implementirao funkciju prepoznavanja lica koja je korisniku pomogla identificirati osobe čije se lice nalazilo na fotografiji koje korisnik objavljuje. Usprkos kritikama medija zbog narušavanja privatnosti, korisnicima Facebooka nova značajka nije smetala i nije negativno utjecala na popularnost web stranice [1].

Uvođenjem većih baza podataka 1990-ih godina preciznost algoritama se povećala, no istraživači su primijetili da na točnost algoritama utječu promjene poput orijentacije glave, različitih izraza lica, šminke ili promjene u osvjetljenju što su bile mane algoritama koji koriste $2D$ modele lica. Te prepreke pokušali su premostiti uvođenjem algoritama koji koriste $3D$ modele lica za prepoznavanje u kojima se koristi trodimenzionalna geometrija ljudskog lica. G. Gordon je u svom radu 1992. godine pokazao da se može poboljšati točnost prepoznavanja lica ako se kombinira frontalni pogled i poluprofil, što se smatra početkom istraživanja algoritama koji su se oslanjali na $3D$ modele prikaza lica. Glavno ograničenje ovih algoritama je generiranje 3D slike i još uvijek je aktivno polje istraživanja [16].

Razvoj strojnog učenja i umjetne inteligencije kroz zadnjih nekoliko godina uvelike je utjecao na razvoj novih i poboljšanje starih algoritama strojnog učenja. Metoda potpornih vektora i metode dubokog učenja (neuronske mreže) algoritmi su koji se često koriste za razlikovanje lica od "ne-lica" i detekciju različitih značajki lica. Facebook se 2014. godine poslužio fotografijama svojih korisnika kako bi trenirao model dubokog učenja DeepFace [11]. Tvrtka nikad nije objavila skup podataka, ali projekt je duboko učenje

[7]Gaborov filter je linearni filter koji se koristi za analizu teksture modelirajući način na koji ljudski vid percipira teksture i rubove jer može analizirati frekvencijske i orjentacijske karakteristike slike. Koristi se u obradi slika i prepoznavanju uzoraka za ekstrakciju značajki sa slika.

promovirao u najbolji algoritam za prepoznavanje lica i u potpuni zaborav bacio ručne provjere i označavanje jer su skupovi za testiranje postali milijunski. Apple je 2017. godine novim iPhone-om X tržištu predstavio uređaj koji koristi prepoznavanje lica kao jednu od svojih primarnih novih značajki sigurnosti [39]. Uređaj je vrlo brzo rasprodan što je dokazalo da potrošači prihvaćaju prepoznavanje lica kao sigurnosnu metodu provjere. Iako se algoritmi za prepoznavanje lica danas smatraju preciznima i brzo napreduju, nošenje maski kao zaštitna mjera tijekom pandemije virusa Covid-19 imalo je neželjenu posljedicu povećanja poteškoća u prepoznavanju lica i istaknule su se slabosti postojećih algoritama u situacijama kada je većina lica sakrivena pa se na taj način istraživačima ukazalo na čemu moraju dodatno poraditi. No, u standardnim okolnostima, današnji algoritmi za prepoznavanje lica jesu precizni. Zbog brzine razvoja algoritama, sustavi za prepoznavanje lica postaju sve dostupniji i korišteniji te su jedan od standarda za sigurnost i zaštitu [1].

2.3 Podjela na algoritme za prepoznavanje $2D$ i $3D$ modela lica

Algoritmi za prepoznavanje lica dijele se na algoritme za prepoznavanje $2D$ i $3D$ modela lica. Kroz povijest puno više su se proučavali i primjenjivali algoritmi za prepoznavanje $2D$ modela lica koji koriste 2D digitalne slike i prepoznaju lice uspoređujući ih s bazom podataka prethodno snimljenih slika. Ovakav pristup u prepoznavanju lica konstruiran je matematičkim modelom kojim se izdvajaju informacije iz digitalne slike kako bi se prepoznale glavne značajke lica koje nazivamo orijentiri i zatim se oni uspoređuju s postojećim slikama u bazi podataka. Klasični pristupi u prepoznavanju lica na osnovu $2D$ modela lica uključuju sljedeće algoritme:

- Eigenface algoritam,
- Fisherfaces ili linearna diskriminantna analiza (eng. *Linear Discriminant Analysis, LDA*),
- nezavisna analiza komponenti (eng. *Independent Component Analysis, ICA*),
- metoda potpornih vektora (eng. *Support Vector Machine, SVM*) i konvolucijske neuronske mreže (eng. *Convolutional Neural Network, CNN*),
- skriveni Markovljev model (eng. *Hidden Markov Model, HMM*).

Svaki od navedenih algoritama ima svoje karakteristike, prednosti i mane. Međutim, ograničenja algoritama ovog tipa su promjene u fizičkom izgledu (starenje, drugačiji izrazi lica, puštanje brade ili brkova, nošenje naočala ili šminke), orijentacija glave na slici i promjena osvjetljenja. Pokazalo se da se ti problemi mogu prevladati korištenjem trodimenzionalnih geometrijskih karakteristika ljudskog lica, odnosno implementiranjem algoritama koji prepoznaju lica na osnovu $3D$ modela. Snimanje lica vrši se koristeći sustav s više kamera (stereoskopijom), koristeći daljinske kamere ili 3D skenerom. Proces započinje $3D$ snimanjem lica i analizom značajki poput dubine i zakrivljenosti crta lica. Nije bitno samo prepoznati položaj bitnih značajki lica, npr. očiju ili usana, već uvidjeti dubinu očnih duplji ili položaj usana na licu. Potrebno je prikupiti više slika iz različitih kutova i kombinirati

ih u cjelovite $3D$ podatke, što se naziva $3D$ rekonstrukcija. Prednosti prepoznavanja $3D$ prikaza lica su očite jer zahtijevaju naprednije algoritme za pretprocesiranje slike, modele akvizicije lica i modele izdvajanja značajki temeljene na prepoznavanju $2D$ lica. Tehnologija bilježi crte lica iz različitih kuteva što povećava točnost i smanjuje mogućnost pojavljivanja lažno pozitivnih ili lažno negativnih prepoznavanja zbog različitog osvjetljenja, sjena ili nekog drugog čimbenika okoline. Također, slika lica je sveobuhvatnija pa je teže zavarati sustav jer se dodatno mogu detektirati neke sitnije karakteristike lica kao što su bore ili ožiljci. Neki od algoritama koji koriste $3D$ modele lica za prepoznavanje su:

- 3D prepoznavanje lica bez rekonstrukcije lica (eng. *3D Face Recognition without Facial Surface Reconstruction*),
- morfabilni model (eng. *morphable model*),
- 3D konvolucijske neuronske mreže (eng. *Convolutional neural network,* $3CNN$).

Sustavi za prepoznavanje $3D$ modela lica jedna su od najnaprednijih tehnologija za prepoznavanje lica i najbrže se razvijaju u današnje vrijeme. Neki od glavnih izazova su točnost i preciznost tehnologije. Iako $3D$ snimanje omogućuje veću percepciju i dubinu od tradicionalnog $2D$ snimanja, još uvijek je teško precizno zabilježiti određene crte lica. Osim toga, uvjeti osvjetljenja i kutevi kamere također mogu utjecati na kvalitetu snimljene slike. Jedna od poteškoća je i količina podataka potrebna za treniranje algoritama koji koriste $3D$ modele lica. Za razliku od algoritama koji koriste $2D$ modele za prepoznavanje lica i koji zahtijevaju samo jednu sliku za identifikaciju osobe, algoritmi koji koriste $3D$ modele za prepoznavanje lica zahtijevaju više slika iz različitih kuteva kako bi slika lica bila sveobuhvatna. Stoga je potrebno prikupiti, obraditi i pohraniti veću količinu podataka, a to zahtjeva bolju i skuplju tehnologiju za pripremu i obradu podataka, treniranje i konkretan proces prepoznavanja lica.

3 Algoritmi za prepoznavanje $2D$ modela lica

Istraživanja prepoznavanja lica provedena $2D$ pristupom mogu se klasificirati u tri kategorije: analitičke, globalne i hibridne. Analitičkim pristupima cilj je prepoznati lice uspoređujući komponente lica, na način da se mjeri udaljenost određenih točaka i kutovi između njih, dok globalni pristup pokušava prepoznati lice samo na temelju podataka izvedenih bez izdvajanja pojedinačnih značajki. Hibridni pristup kombinira analitičke i globalne pristupe kako bi pokušao dobiti podatke koji točnije određuju lice.

3.1 Osnovne definicije i pojmovi

Definicija 3.1.1. Preslikavanje $A : \{1, \ldots, m\} \times \{1, \ldots, n\} \to \mathbb{F}$ naziva se *matrica* dimenzije (m, n) s koeficijentima iz polja $\mathbb{F}$. Elementi matrice su skalari $a_{ij} = A_{ij} = A(i, j)$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. Matrica A^T naziva se *transponirana matrica* i vrijedi $A^T = [a_{ji}]$ matrice $A = [a_{ij}]$, $i = 1, \ldots, m$, $j = 1, \ldots, n$.

Napomena 3.1.1. Skup svih matrica dimenzije $m \times n$ s elementima iz polja $\mathbb{F}$ označavamo $M_{mn}(\mathbb{F})$, odnosno vrijedi da je $A \in M_{mn}(\mathbb{F})$, ako je A matrica s m redaka i n stupaca s elementima iz polja $\mathbb{F}$. Ako je broj redaka matrice A jednak broju stupaca, tada kažemo da je A kvadratana matrica, a skup svih *kvadratnih matrica* s n redaka, odnosno stupaca nad poljem $\mathbb{F}$ označavamo $M_{nn}(\mathbb{F})$ ili kraće $M_n(\mathbb{F})$.

Definicija 3.1.2. Skalar $\lambda \in \mathbb{F}$ je *svojstvena vrijednost* matrice $A \in M_{nn}(\mathbb{F})$ ako postoji ne-nul vektor v tako da vrijedi

$$Av = \lambda v. \tag{1}$$

Svaki vektor v koji zadovoljava jednakost (1) naziva se *svojstveni vektor* matrice A pridružen svojstvenoj vrijednosti λ.

Definicija 3.1.3. *Varijanca* (mjera disperzije) je prosječno kvadratno odstupanje vrijednosti niza podataka od njegove aritmetičke sredine. Neka je $x_1, \ldots, x_n$, niz podataka i neka je $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ aritmetička sredina toga niza. Varijanca se označava sa σ^2 i definira

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

Definicija 3.1.4. *Kovarijanca* je aritmetička sredina produkata odstupanja vrijednosti dviju varijabla od njihova prosjeka. Za dane varijable X i Y, neka su $(x_i, y_i),\ i = 1, \ldots, n$, parovi njihovih vrijednosti i $\bar{x}, \bar{y}$ njihove aritmetičke sredine. Kovarijanca od X i Y je definirana:

$$\mathrm{Cov}(X, Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).$$

Napomena 3.1.2. Ukoliko želimo izračunati varijancu i kovarijancu uzorka, a ne cijele populacije onda u formulama navedenim u definicijama 3.1.3. i 3.1.4. dijelimo s $n-1$, a ne n, gdje je n veličina uzorka.

Definicija 3.1.5. *Kovarijacijska matrica* slučajnog vektora $X = (X_1, \ldots, X_n)$ se definira

$$\mathrm{Cov}(X) = \begin{bmatrix} \mathrm{Var}(X_1) & \ldots & \mathrm{Cov}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_1, X_n) & \ldots & \mathrm{Var}(X_n) \end{bmatrix}.$$

Napomena 3.1.3. Kako za dane varijable X i Y vrijedi

$$\mathrm{Cov}(X, Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) = \mathrm{Cov}(Y, X),$$

slijedi da je matrica kovarijanci simetrična matrica, odnosno da vrijedi $\mathrm{Cov}(X) = (\mathrm{Cov}(X))^T$.

Primjer 3.1.1. Neka su dana tri skupa podataka $X = \{1, 3, 5, 7, 9\}$, $Y = \{2, 4, 6, 8, 10\}$ i $Z = \{1, 4, 6, 9, 11\}$. Odredite kovarijancu između danih skupova i kovarijacijsku matricu. Vrijedi da je $|X| = |Y| = |Z| = 5$, pa su aritmetičke sredine skupova X, Y i Z jednake

$$\bar{x} = \frac{1}{5}\sum_{i=1}^{5} x_i = \frac{25}{5} = 5, \quad \bar{y} = \frac{1}{5}\sum_{i=1}^{5} y_i = \frac{30}{5} = 6, \quad \bar{z} = \frac{1}{5}\sum_{i=1}^{5} z_i = \frac{31}{5} = 6.2.$$

Kovarijance između zadanih skupova iznose:

$$\mathrm{Cov}(X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{4}\sum_{i=1}^{5}(x_i - 5)(y_i - 6) = \frac{40}{4} = 10,$$

$$\mathrm{Cov}(X, Z) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(z_i - \bar{z}) = \frac{1}{4}\sum_{i=1}^{5}(x_i - 5)(z_i - 6.2) = \frac{50}{4} = 12.5,$$

$$\mathrm{Cov}(Y, Z) = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})(z_i - \bar{z}) = \frac{1}{4}\sum_{i=1}^{5}(y_i - 6)(z_i - 6.2) = \frac{50}{4} = 12.5.$$

Varijance danih skupova podataka su:

$$\sigma_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{4}\sum_{i=1}^{5}(x_i - 5)^2 = \frac{40}{4} = 10,$$

$$\sigma_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2 = \frac{1}{4}\sum_{i=1}^{5}(y_i - 6)^2 = \frac{40}{4} = 10,$$

$$\sigma_Z^2 = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - \bar{z})^2 = \frac{1}{4}\sum_{i=1}^{5}(z_i - 6.2)^2 = \frac{157}{10} = 15.7.$$

Kovarijacijska matrica slučajnog vektora (X, Y) je:

$$\begin{bmatrix} \mathrm{Var}(X) & \mathrm{Cov}(X,Y) & \mathrm{Cov}(X,Z) \\ \mathrm{Cov}(X,Y) & \mathrm{Var}(Y) & \mathrm{Cov}(Y,Z) \\ \mathrm{Cov}(X,Z) & \mathrm{Cov}(Y,Z) & \mathrm{Var}(Z) \end{bmatrix} = \begin{bmatrix} 10 & 10 & 12.5 \\ 10 & 10 & 12.5 \\ 12.5 & 12.5 & 15.7 \end{bmatrix}.$$

Definicija 3.1.6. Za dva vektora x i y kažemo da su *ortogonalna* ako im je skalarni produkt jednak 0.

Definicija 3.1.7. Za vektor $x \in \mathbb{R}^n$ kažemo da je *normiran* ako je njegova norma jednaka 1, odnosno $||x|| = \sqrt{\sum_{i=1}^{n} x_i^2} = 1$.

Definicija 3.1.8 Neka je V vektorski prostor nad poljem K. Skup $\{v_1, \ldots, v_n\}$ se naziva *baza* vektorskog prostora ako je $\{v_1, \ldots, v_n\}$ linearno nezavisan skup koji generira cijeli prostor.

Napomena 3.1.4. Neka je $B = \{v_1, v_2, \ldots, v_n\}$ baza vektorskog prostora V nad poljem K. Tada skup B generira cijeli vektorski prostor V što znači da za svaki vektor

$x \in V$ postoje jedinstveni skalari $\lambda_1, \ldots, \lambda_n \in K$ tako da vrijedi

$$x = \sum_{i=1}^{n} \lambda_i v_i.$$

Baza vektorskog prostora nije jedinstvena, ali svake dvije baze vektorskog prostora sadrže jednak broj vektora i kardinalnost baze jednaka je dimenziji vektorskog prostora[8].

Definicija 3.1.9. *Euklidska udaljenost* vektora $x, y \in \mathbb{R}^n$ je definirana kao

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

3.1.1 Nadzirano učenje

Nadzirano učenje je vrsta strojnog učenja koja koristi označene skupove podataka za učenje, odnosno treniranje algoritama što znači da su trening podaci podijeljeni na ulazne i izlazne podatke, odnosno na nezavisne i zavisne varijable. Cilj je da model pronađe vezu između njih tijekom procesa treniranja kako bi mogao ispravno predvidjeti izlazne vrijednosti na novim, nepoznatim ulaznim podacima.

Ulazni podaci algoritama nadziranog učenja dani su nizom značajki (atributa), a izlazni podatak je vrijednost odgovarajuće zavisne varijable za određeni niz značajki. Niz značajki je formaliziran *vektorom značajki* $x = (x_1, \ldots, x_n)$, gdje je x_i vrijednost pojedinog atributa, a n ukupan broj atributa. Prostor svih primjera, ulaznih podataka označen je s $\mathcal{X}$, a svih izlaznih vrijednosti s $\mathcal{Y}$. Ukupan broj primjera je N. Dakle, vrijedi

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N} \subseteq \mathcal{X} \times \mathcal{Y}.$$

Cilj je odrediti funkciju $h : \mathcal{X} \to \mathcal{Y}$ koja se naziva *hipoteza* i koja je nepoznata. Model strojnog učenja definiramo kao skup hipoteza koje su parametrizirane s θ, odnosno

$$\mathcal{H} = \{h(x; \theta)\}_{\theta}.$$

Svaki parametar θ odgovara jednoj funkciji te je cilj pronaći "najbolju" hipotezu h, pa se opisani problem klasificira kao optimizacijski problem. Za svaku hipotezu se određuje koliko je "dobra" na skupu označenih primjera $\mathcal{D}$, a odstupanja se mjere *empirijskom greškom*, $E(h \mid \mathcal{D})$. Svaki primjer iz skupa primjera pridonosi empirijskoj grešci, a greška svakog pojedinog primjera mjeri se *funkcijom gubitka* J koja je definirana

$$J : \mathcal{Y} \times \mathcal{Y} \to R_0^+,$$

tako da je

$$J(y, h(x)) = \alpha,$$

gdje je y izlazna vrijednost za ulaz x, $h(x)$ je vrijednost hipoteze tog primjera i $\alpha \in \mathbb{R}_0^+$ vrijednost funkcije gubitka. U slučaju da želimo odrediti hipotezu za klasifikacijski primjer,

[8] dokaz tvrdnji navedenih u napomeni može se pronaći u [20].

onda je funkcija gubitka jednaka nuli u slučaju kad je klasifikacija točno određena. Empirijska greška je definirana kao očekivana vrijednost funkcije gubitka na skupu označenih primjera:

$$E(h\,|\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} J(y^{(i)}, h(x^{(i)}).$$

Cilj je pronaći hipotezu $h \in \mathcal{H}$ koja minimizira empirijsku grešku, odnosno riješiti optimizacijski problem:

$$h^* = \text{argmin}_{h\in\mathcal{H}} E(h \mid \mathcal{D}).$$

Hipoteza h^* je hipoteza za koju je greška klasifikacije trening primjera najmanja [4].

3.2 Analiza glavnih komponenata (PCA)

Analiza glavnih komponenti (eng. *Principal Component Analysis, PCA*) je metoda smanjenja dimenzionalnosti koja se često koristi za reduciranje velikih skupova podataka. Cilj metode je smanjiti broj varijabli smanjenjem dimenzionalnosti tako da se sačuvaju najvažnije karakteristike skupa, a izbrišu one manje važne. Time se smanjuje točnost metode, ali se povećava njena jednostavnost pa je skupove podataka lakše analizirati i vizualizirati. Navedeno se postiže tako da se odabire glavna komponenta varijacije u izvornom skupu podataka, a ostale komponente se određuju ortogonalno na prethodne i predstavljaju najveći sljedeći izvor varijacije. Značajke se tada prikazuju kao linearna kombinacija glavnih komponenti. Cilj je da su nove značajke (glavne komponente) različite, odnosno da je njihova kovarijanca jednaka 0 i da je zbroj varijacije novih značajki jednak zbroju varijacije izvornih obilježja. Svaka uzastopna glavna komponenta objašnjava varijancu koja je ostala nakon njene prethodne komponente, tako da odabir samo nekoliko prvih komponenti dovoljno aproksimira izvorni skup podataka bez potrebe za dodatnim značajkama. Neka je dan skup podataka X s n podataka, od kojih svaki ima k značajki:

$$X = \begin{bmatrix} x_{11} & \dots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nk} \end{bmatrix}.$$

Izračunamo aritmetičku sredinu $\bar{x}_1, \dots, \bar{x}_k$ za svaku od značajki i normaliziramo skup podataka X te dobivamo

$$X' = \begin{bmatrix} x_{11} - \bar{x}_1 & \dots & x_{1k} - \bar{x}_k \\ \vdots & \ddots & \vdots \\ x_{n1} - \bar{x}_1 & \dots & x_{nk} - \bar{x}_k \end{bmatrix}.$$

Računamo matricu kovarijance za normalizirani skup podataka X':

$$\Sigma = \frac{1}{n}(X')^T X'.$$

Odredimo svojstvene vektore i svojstvene vrijednosti matrice kovarijance, Σ. Ako želimo odrediti k glavnih komponenti, promatramo k najvećih svojstvenih vrijednosti i njima pridružene svojstvene vektore definiramo kao glavne komponente.

Primjer 3.2.1. Za zadani skup podataka X potrebno je odrediti prve dvije glavne komponente.

$$X = \begin{bmatrix} -1 & 1 & -2 \\ 2 & -1 & 1 \\ -2 & 1 & -1 \\ 1 & 2 & 1 \end{bmatrix}.$$

Za zadani skup podataka aritmetičke sredine su $\bar{x}_1 = 0, \bar{x}_2 = 0.75, \bar{x}_3 = -0.25$. Normalizirana matrica X' je jednaka

$$X' = \begin{bmatrix} -1 & 0.25 & -1.75 \\ 2 & -1.75 & 1.25 \\ -2 & 0.25 & -0.75 \\ 1 & 1.25 & 1.25 \end{bmatrix}.$$

Matrica kovarijanci je tada:

$$\Sigma = \frac{1}{4}(X')^T X' = \frac{1}{4} \begin{bmatrix} 10 & -3 & 5 \\ -3 & 5 & -3.75 \\ -5 & -3.75 & 5.75 \end{bmatrix} = \begin{bmatrix} 2.5 & -0.75 & 1.25 \\ -0.75 & 1.1875 & -0.3125 \\ 1.25 & -0.3125 & 1.6875 \end{bmatrix}.$$

Svojstvene vrijednosti matrice Σ su:

$$\lambda_1 = 0.308, \quad \lambda_2 = 0.988, \quad \lambda_3 = 3.829,$$

s pridruženim svojstvenim vektorima:

$$v_1 = [-0.29 \ \ 0.818 \ \ 1]^T, \quad v_2 = [-1.739 \ \ -1.840 \ \ 1]^T, \quad v_3 = [1.358 \ \ -0.740 \ \ 1]^T,$$

respektivno. Kako je λ_3 najveća svojstvena vrijednost, prva glavna komponenta je svojstveni vektor v_3, a zatim je sljedeća najveća svojstvena vrijednost λ_2 pa je vektor v_2 druga glavna komponenta danog skupa podataka. Dakle, vektor značajki danog skupa podataka se sastoji od svojstvenih vektora v_3 i v_2 i dan je s

$$\begin{bmatrix} -1 - 739 & 1.358 \\ 1.840 & -0.740 \\ 1 & 1 \end{bmatrix}.$$

Početan, trodimenzionalan skup podataka postao je dvodimenzionalan odbacivanjem trećeg svojstvenog vektora v_1 s najmanjom pridruženom svojstvenom vrijednošću λ_1. Ako bismo dodatno htjeli smanjiti dimenzionalnost danog skupa podataka, odbacili bismo svojstveni vektor v_2 i vektor značajki bio bi jednak svojstvenom vektoru v_3 s najvećom pridruženom svojstvenom vrijednošću λ_3. Generalno, ukoliko odlučimo zadržati k svojstvenih vektora koji su pridruženi k najvećim svojstvenim vrijednostima od ukupno n, smanjili smo dimenzionalnost početnog skupa podataka s n na k. Biramo k svojstvenih vektora tako da oni najviše pridonose varijaciji skupa podataka, a smanjena dimenzionalnost omogućuje bržu analizu i obranu prilikom rješavanja problema [25].

Algoritmi za prepoznavanje lica kao ulazni podatak primaju sliku koja je prikazana kao polje piksela dimenzija $N \times N$ ili vektor piksela duljine N^2, pri čemu je N velik. Na ulazno polje ili vektor se zatim primjenjuje opisani PCA algoritam kako bi se smanjila dimenzionalnost prostora, a očuvale bitne značajke, odnosno najbitnije varijacije u pikselima na slici. Time se sama kvaliteta slike gubi (s obzirom da se reduciraju pikseli na slici), ali čuvaju se one najbitnije informacije za raspoznavanje.

3.3 Eigenface metoda

Eigenface metodu razvili su L. Sirovich i M. Kirby 1986. godine u svom radu [37] te se ona koristi u predstavljanju i prepoznavanju lica na temelju analize glavnih komponenti. Većina prethodnih radova koji su se bavili temom automatiziranog prepoznavanja lica ignorirala je pitanje koji su aspekti lica bitni za prepoznavanje. Ideja Eigenface pristupa bila je izdvojiti relevantne informacije sa slike lica, učinkovito ih kodirati i usporediti kodirano lice s bazom podataka svih kodiranih lica. Glavni cilj je bio osigurati da odabrane značajke lica sačuvaju varijaciju između lica pohranjenih u bazi. Tehnički, potrebno je pronaći glavne komponente distribucije lica, pri čemu je svaka slika prikazana kao vektor u visokodimenzionalnom prostoru. Primjenom analize glavnih komponenti smanjuje se dimenzionalnost početnog prostora tako da se uvodi pojam *vlastitih lica* (eng. *eigenfaces*). Vlastita lica su glavne komponente koje dijele početno lice na skup *vektora značajki (svojstvenih vektora)*. Vektori značajki mogu se odrediti kao svojstveni vektori matrice kovarijanci skupa slika lica i koriste se za određivanje varijacije između više slika. Svako lice u bazi se tada može reprezentirati kao linearna kombinacija vlastitih lica. Lice se može aproksimirati korištenjem vlastitih lica koji imaju najveće svojstvene vrijednosti i koja stoga određuju najveću varijaciju. Najboljih M vlastitih lica (M lica koja su dobivena kao glavne komponente provedene PCA analize) generiraju M-dimenzionalni potprostor početnog prostora koji se naziva *prostor lica* [32].

Počevši od skupa originalnih slika lica, izračunali su koordinatni sustav čije su osi glavne komponente (svojstveni vektori) dobivene provedbom PCA analize. Koordinantne osi su na taj način usklađene sa smjerovima u kojima slike lica najviše variraju i time je koordinatni sustav učinkovitiji za predstavljanje lica od izvornog (izvorne vrijednosti piksela). Koordinate u novom koordinatnom sustavu su zapravo slike koje se nazivaju *vlastite slike* (eng. *eigenpictures*). Za svaku sliku lica se određuje udaljenost slike lica od svih vlastitih lica koordinatnog sustava i na temelju toga pohranjuju se koeficijenti, odnosno kvantificirani podaci o tome koliko svako vlastito lice sudjeluje u slici što definira težinu pripadnog lica. Tvrdili su da se slike lica mogu približno rekonstruirati pohranjivanjem težina i vlastite slike, pri čemu se težine koje opisuju svako lice dobivaju projiciranjem slike lica na vlastitu sliku. Razvila se ideja da, ako se mnoštvo slika lica može rekonstruirati korištenjem male zbirke vlastitih slika i težina, možda je učinkovit način prepoznavanja lica usporediti težine i vlastite slike za dano lice s težinama i vlastitim slikama lica u bazi.

Neka je slika lica $I(x, y)$ dvodimenzionalno $N \times N$ polje 8-bitnih vrijednosti. Slika također može biti predstavljena kao vektor duljine N^2 ili kao točka u prostoru dimenzije N^2. S obzirom na to da su slike slične u ukupnoj konfiguraciji, one neće biti nasumično raspoređene u ogromnom prostoru slika i mogu se opisati potprostorom relativno niske

dimenzije. Analizom glavnih komponenti određujemo vlastita lica i pomoću onih vlastitih lica koja imaju najveće svojstvene vrijednosti generiramo potprostor čime je dimenzija prostora znatno smanjena u odnosu na dimenziju početnog prostora. Usporedba između slika lica pohranjenih u bazi i pripadnih vlastitih lica dana je na Slici 3, [32].

Slika 3: Usporedba slika lica pohranjenih u bazi i vlastitih lica nakon provedene analize glavnih komponenata

Slika 3 prikazuje usporedbu slika lica pohranjenih u bazi podataka i dobivena vlastita lica provedbom analize glavnih komponenata. Slike pohranjene u bazi imaju veću razlučivost i oštrinu, dok se kod slika vlastitih lica razaznaju samo najbitnije značajke i sjene. To se događa zbog toga što PCA algoritam smanjuje dimenzionalnost broja piksela na slici čime se gubi na kvaliteti, ali se očuvaju najbitnije značajke i ubrzava se proces prepoznavanja.

Eigenface model možemo opisati na sljedeći način. Pretpostavimo da je $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_M\}$ trening skup slika lica. Definiramo prosječno lice trening skupa kao prosjek svih lica u trening skupu, $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$. Prosječno lice se određuje računanjem prosječne vrijednosti svakog piksela na slikama lica standardiziranih dimenzija. Svako lice se od prosječnog lica razlikuje za $\Phi_i = \Gamma_i - \Psi,\ \ i = 1, \ldots, M$. Prosječno lice za lica iz baze podataka prikazana na Slici 3 je prikazano na Slici 4, [32].

Slika 4: Prosječno lice slika lica pohranjenih u bazi koje su prikazana na Slici 3

Želimo odrediti M ortogonalnih i normiranih vektora u_i koji najbolje opisuju distribu-

ciju podataka. Stoga je cilj maksimizirati varijancu glavnih komponenti, a to se postiže određivanjem svojstvenih vektora matrice kovarijanci C s najvećim svojstvenim vrijednostima. Matrica kovarijanci C opisuje odstupanja lica iz trening skupa od prosječnog lica i definirana je s

$$C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T = AA^T, \tag{2}$$

gdje je matrica $A = [\Phi_1 \ \Phi_2 \ \dots \ \Phi_n]$ matrica dimenzija $N^2 \times M$. Svojstveni vektor u_k matrice C s pripadnom svojstvenom vrijednošću λ_k određujemo rješavajući jednadžbu

$$Cu_k = \lambda_k u_k.$$

Vektori u_i, $i = 1, \dots, M$, su glavne komponente PCA analize, a njima odgovarajuće svojstvene vrijednosti λ_i, $i = 1, \dots, M$, predstavljaju količinu varijance koja je obuhvaćena tim vektorom. Za svaki svojstveni vektor u_k mora vrijediti

$$\lambda_k = \frac{1}{M}\sum_{i=1}^{M}(u_k^T\Phi_i)^2 \tag{3}$$

te mora biti zadovoljen uvjet ortonormiranosti:

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1, & l = k \\ 0, & \text{inače} \end{cases}, \qquad l, k = 1, \dots, M,$$

gdje je δ_{lk} Kroneckerov simbol[9]. Uvjet ortonormiranosti je posljedica maksimizacije varijance jer želimo postići da je kovarijanca glavnih komponenti nula, odnosno vektora u_l i u_k za $l \neq k$ jednaka 0. Kako se svaka slika može prikazati kao vektor duljine N^2, matrica kovarijanci C je dimenzije $N^2 \times N^2$ pa je računanje svojstvenih vektora duljine N^2 računalno zahtjevno i skupo (za standardnu sliku dimenzija 256×256 je $N^2 = 256^2 = 65536$). Ako je početni trening skup slika odabran tako da vrijedi da je $M < N^2$, onda je broj svojstvenih vektora s ne-nul svojstvenim vrijednostima najviše $M - 1$. Stoga, kako bi početan problem bio računalno prihvatljiviji, određujemo svojstvene vektore matrice A^TA dimenzija $M \times M$. Na primjer, promotrimo trening skup podataka sa 100 slika gdje slike imaju 256×256 piksela. Tada je $N^2 = 65536$, dok je $M = 100$, odnosno C je matrica dimenzija 65536×65536, dok je matrica A dimenzija 100×100.

Propozicija 3.3.1. Neka je A kvadratna matrica reda n. Ako je v svojstveni vektor matrice A^TA s pridruženom svojstvenom vrijednošću $\lambda \neq 0$, onda je Av svojstveni vektor matrice AA^T sa svojstvenom vrijednošću λ.

Dokaz. Neka je v svojstveni vektor matrice A^TA sa svojstvenom vrijednošću $\lambda \neq 0$. Tada vrijedi:

$$A^TAv = \lambda v.$$

Pomnožimo li prethodnu jednakost slijeva matricom A dobivamo:

$$A(A^TAv) = A(\lambda v).$$

[9]Kroneckerov simbol δ_{ij} je definiran kao $\delta_{ij} = 1$ ako je $i = j$ i $\delta_{ij} = 0$ ako je $i \neq j$.

Iz svojstva asocijativnosti množenja matrica i komutativnosti množenja matrice sa skalarom dobivamo:

$$(AA^T)Av = \lambda(Av).$$

Slijedi da je λ svojstvena vrijednost matrice AA^T pridružena svojstvenom vektoru Av. $\square$

Svojstvene vektore v_i, $i = 1, \ldots, M$, matrice A^TA određujemo rješavanjem sustava:

$$A^TAv_i = \mu_i v_i,$$

pri čemu su μ_i, $i = 1, \ldots, M$, svojstvene vrijednosti pripadnih svojstvenih vektora. Iz prethodne propozicije i definicije matrice $C = AA^T$, slijedi da je Av_i svojstveni vektor matrice C. Stoga, definiramo $M \times M$ matricu $L = A^TA$, gdje je $L[l_{ij}] = (A^TA)_{ij} = \Phi_i^T\Phi_j$, $i, j = 1, \ldots, M$, i određujemo svojstvene vektore v_i matrice L. Pomoću generiranih M svojstvenih vektora, uz M početnih slika lica iz trening skupa podataka i prosječno lice, definira se svojstveno lice (eng. eigenface) kao sljedeća linearna kombinacija

$$u_l = \sum_{i=1}^{M} v_{li}\Phi_i, \qquad l = 1, \ldots, M. \tag{4}$$

U praksi, veličina trening skupa M bit će znatno manja od dimenzije vektora slike N^2. Smanjenjem dimenzionalnosti korištenjem PCA metode izračuni se znatno ubrzavaju i postaju računalno podnošljivi [32].

3.3.1 Prepoznavanje lica korištenje Eigenface metode

L. Sirovich i M. Kirby su testirali pojednostavljenu verziju Eigenface metode na skupu od 115 slika muškaraca bijele rase, snimljenih u kontroliranim uvjetima i otkrili su da je oko 40 vlastitih lica dovoljno za učinkovito opisivanje skupa slika lica. S obzirom na to da su se vlastita lica pokazala učinkovita za opisivanje slika lica u kontroliranim uvjetima, postavilo se pitanje koliko su učinkovita za identifikaciju lica. U praksi, za identifikaciju lica je potrebno manje svojstvenih lica jer nije potrebna rekonstrukcija slike. Stoga se može odabrati $M' < M$ vlastitih lica i identifikacija postaje problem prepoznavanja uzorka. Vlastita lica čine bazu prostora lica M'. Neka je s Γ označena nova slika lica. Proces transformacije nove slike lica u prostor lica definira se kao projiciranje slike i opisano je na sljedeći način:

$$\omega_k = u_k^T(\Gamma - \Psi), \qquad k = 1, \ldots, M'. \tag{5}$$

Nova slika centrira se s obzirom na prosječnu sliku lica, što je u (5) opisano s $\Gamma - \Psi$, a zatim se svaki piksel te slike množi s odgovarajućom vrijednošću piksela slika vlastitih lica i zbrajaju se dobivene vrijednosti. Neka je

$$\Omega^T = [\omega_1 \ \ldots \omega_{M'}]$$

vektor težina, gdje ω_k predstavlja doprinos k-tog vlastitog lica za projiciranu sliku u prostor lica. Vektor težina se koristi u algoritmu prepoznavanja uzorka za klasifikaciju

ulazne slike lica. Cilj je odrediti koja od slika iz baze najbolje odgovara novoj slici lica. Najjednostavnija metoda za klasifikaciju vlastitog lica koje je najpodudarnije s ulaznom slikom Γ jest pronaći vlastito lice k koje minimizira Euklidsku udaljenost

$$\epsilon_k = ||\Omega - \Omega_k||^2,$$

gdje je Ω_k vektor težina za k-to vlastito lice i izračunava se uprosječivanjem težina malog broja slika lica za svakog pojedinca (ponekad je to samo jedna slika po pojedincu). Ulazna slika se podudara s k-tim vlastitim licem ako je $\epsilon_k < \theta$, gdje je θ odabrani prag. Ako je $\epsilon_k > \theta$, $\forall k = 1, \ldots, M'$, onda se ulazno lice označuje kao "nepoznato". Lice klasificirano kao "nepoznato", može se koristiti za stvaranje novog vlastitog lica, proširujući skup poznatih lica. Budući da je stvaranje vektora težine ekvivalentno projiciranju izvorne slike lica na niskodimenzionalni prostor lica, mnoge slike se mogu projicirati na zadani uzorak, pa čak i one koje nisu slike lica. No, to nije problem sustava jer je udaljenost između slike i prostora lica definirana kao kvadrat udaljenosti između razlike ulazne, uprosječene slike, $\Phi = \Gamma - \Psi$ i njezine projekcije u prostor lica $\Phi_f = \sum_{i=1}^{M'} \omega_i u_i$:

$$\epsilon^2 = ||\Phi - \Phi_f||^2.$$

Ova udaljenost pomaže u određivanju koliko se dobro slika uklapa u potprostor slika lica i ako je ona veća od određene vrijednosti praga θ' (vrijednost praga za udaljenost od potprostora ne mora biti jednako definirana kao vrijednost praga za udaljenost od vlastitih lica) promatrana slika se ne smatra slikom lica.

Ukratko, Eigenface algoritam za prepoznavanje lica možemo opisati u sljedećih nekoliko koraka:

1. potrebno je izraditi i organizirati bazu slika lica tako da je u bazu uključeno više slika lica, s varijacijama u izrazima lica, položaju glave i osvjetljenju. Na primjer, promatramo po 5 slika za 10 osoba čime dobivamo bazu od 50 slika,

2. izračunati matricu L dimenzija $M \times M$, gdje je M dimenzija prostora lica, odnosno broj vlastitih lica i odrediti svojstvene vektore i svojstvene vrijednosti. Odaberemo M' svojstvenih vektora s najvećim vrijednostima svojstvenih vrijednosti,

3. generirati vlastita lica s pomoću odabranih svojstvenih vektora i odrediti prostor lica,

4. za svako vlastito lice izračunati vektor težina $\Omega_k,\ k = 1, \ldots, M'$, te definirati vrijednost praga θ te dopuštenu udaljenost θ' od prostora lica,

5. za svaku novu sliku izračunati vektor težina Ω, euklidske udaljenosti $\epsilon_k, k = 1, \ldots, M'$, i udaljenost od prostora lica. Ako postoji jedan (ili više njih) indeksa k tako da vrijedi $\epsilon_k < \theta$, i zadovoljena je udaljenost od prostora lica, ulazna slika se identificira kao k-to vlastito lice, pri čemu je indeks k ono lice za koje je dobivena udaljenost najmanja. U suprotnom, ako je udaljenost od potprostora lica zadovoljena, ali ne postoji indeks k za koji vrijedi da je $\epsilon_k < \theta$, ulazna slika lica se smatra "nepoznatim" licem. Ako udaljenost od potprostora nije zadovoljena onda se ulazna slika ne smatra slikom lica,

6. ako je nova slika klasificirana kao poznata osoba u bazi, slika se može dodati bazi te se ponovno može provesti postupak opisan u prva tri koraka čime se modificira prostor lica i omogućuje prilagodba sustava na dodana nova lica.

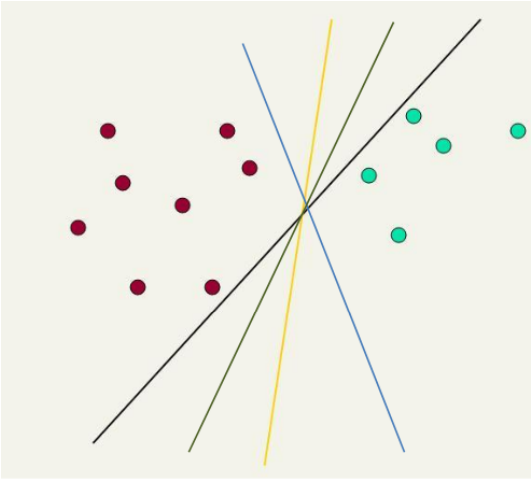
Eigenface pristup je učinkovit jer smanjuje dimenzionalnost slika lica, čineći proces prepoznavanja računalno učinkovitim uz zadržavanje bitnih značajki za razlikovanje različitih lica. Nedostatak algoritma je potreba za centriranim slikama (sve slike lica moraju biti prilagođene na način da su bitne značajke lica poput očiju, nosa, usta i druge na jednakoj poziciji na svim slikama) koje moraju biti jednakih dimenzija. Dodatno, algoritam je osjetljiv na uvjete osvjetljenja, sjenu, veličinu lica na slici te različite položaje glave [32].

3.4 SVM algoritam za prepoznavanje lica

Metoda potpornih vektora (eng. *Support Vector Machine, SVM*) je algoritam nadziranog učenja koji predviđa kojoj klasi pojedini podatak pripada.

3.4.1 Opis SVM algoritma

Neka je $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ trening skup podataka, gdje je $x^i = (x_1^{(i)}, \ldots, x_m^{(i)})$ vektor značajki i $y^{(i)}$, $i = 1, \ldots, N$, varijabla vrijednosti. Podaci su prikazani kao m-dimenzionalni vektori. Pitamo se je li moguće za dane podatke odrediti $(m-1)$-dimenzionalnu *razdvajajuću ravninu (hiperravninu)*. Neka su X i Y skupovi točaka u n-dimenzionalnom Euklidskom prostoru. X i Y su *linearno separabilni* ako postoji $n+1$ realnih brojeva $w_1, \ldots, w_n, k$ tako da za $\forall x \in X$ vrijedi $\sum_{i=}^{n} w_i x_i < k$ i $\forall y \in Y$ vrijedi $\sum_{i=1}^{n} y_i w_i > k$. Ako je problem linearno separabilan, moguće je pronaći takvu hiperravninu ili više njih i u toj situaciji želimo odabrati onu koja najbolje razdvaja podatke, odnosno onu koja je najudaljenija od oba skupa podataka kao što je prikazano na Slici 5, [41]. Stoga, cilj algoritma je maksimizirati udaljenost hiperravnine od oba skupa.



Slika 5: Primjer linearno separabilnog skupa podatka i mogućih razdvajajućih hiperravnina

Jednadžba tražene hiperravnine je

$$h(x) = w^T x + b = 0, \tag{6}$$

gdje je w vektor normale pripadne hiperravnine, a b određuje udaljenost hiperravnine od ishodišta u smjeru vektora w. Najbliži podaci (oni za koje je udaljenost od hiperravnine najmanja) iz svake klase pripadaju hiperravninama koje nazivamo *marginama*, a područje omeđeno marginama *marginalnim područjem*. Točke koje se nalaze na margini nazivaju se *potporni vektori*. Uvijek postoje barem dva potporna vektora, po jedan za svaku klasu. Udaljenost d svake točke od hiperravnine $w^T x + b = 0$ je

$$d = \frac{w^T x + b}{||w||}.$$

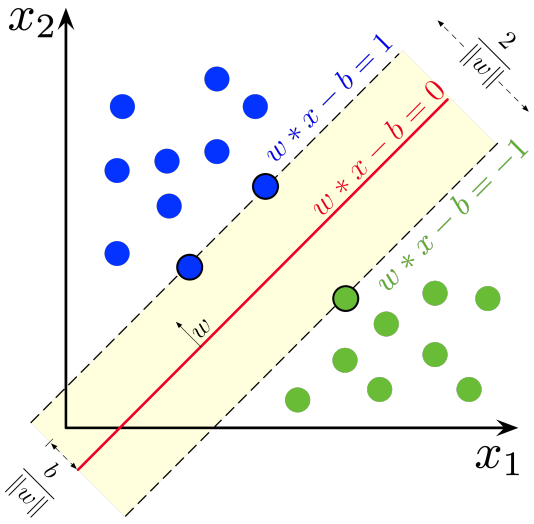
Cilj je maksimizirati marginu pa je potrebno odrediti parametre w, b i za koje će margine biti maksimalno udaljene:

$$\text{argmax}_{w,b}\Big\{\frac{1}{||w||}\min_i\{y^{(i)}(w^T x^{(i)} + b)\Big\}. \tag{7}$$

Time smo dobili optimizacijski problem. Stoga, kako bismo pojednostavili (7), možemo pretpostaviti da za primjere $x^{(i)}$ koji su najbliži margini vrijedi $y^{(i)}(w^T x^{(i)} + b) = 1$, a onda za sve ostale primjere mora vrijediti

$$y^{(i)}(w^T x^{(i)} + b) \geq 1, \;\; i = 1, \ldots, N.$$

Udaljenost između margina je $\frac{2}{||w||}$. Na Slici 6, [40] prikazana je maksimalno razdvajajuća hiperravnina za binarni problem klasifikacije.



Slika 6: Prikaz maksimalno razdvajajuće hiperravnine, margina i potpornih vektora

Tada (7) možemo zapisati kao

$$\begin{gathered}\text{argmax}_{w,b}\frac{1}{||w||}, \\ y^{(i)}(w^T x^{(i)} + b) \geq 0, \; i = 1, \ldots, N.\end{gathered} \tag{8}$$

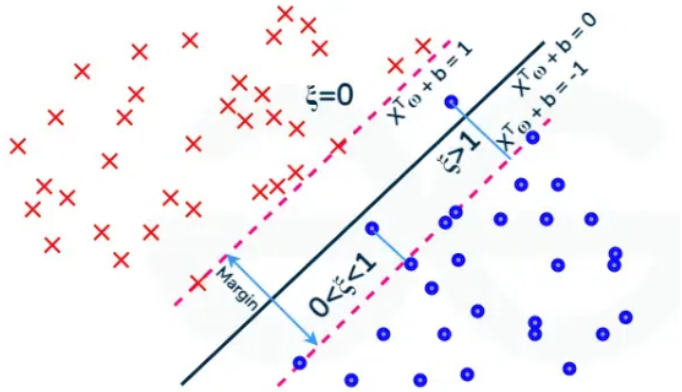
U kontekstu optimizacije, problem (8) ekvivalentno možemo zapisati

$$\begin{aligned} &\text{argmax}_{w,b}\frac{1}{2}||w||^2, \\ &y^{(i)}(w^T x^{(i)} + b) \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{9}$$

Ako problem nije linearno separabilan, odnosno početni skup podataka se ne može linearno odvojiti, tada se koristi potpuno drugačiji pristup od onoga kada je problem linearno separabilan. Jedno od mogućih rješenja takve vrste problema je proširenje prethodnog modela tako da dopustimo netočnu klasifikaciju pojedinih primjera. Takva formulacija problema zove se *meka margina*. Kod meke margine dozvoljavamo netočne klasifikacije pojedinih primjera (želimo da je takvih primjera što je manje moguće), ali svaku takvu klasifikaciju "kažnjavamo" na način što je primjer dalje od prave granice, kažnjavanje je strože. Neka su

$$\xi \geq 0, \quad i = 1, \ldots, N,$$

varijable koje određuju vrijednost kazne. Za ispravno klasificirane primjere vrijedi $\xi_i = 0$, za primjere koji su unutar marginalnog područja, ali s ispravne strane granice vrijedi $0 < \xi_i \leq 1$, a za primjere sa suprotne strane granice definiramo $\xi_i = |y^{(i)} - h(x^{(i)})|$ kao što je prikazano na Slici 7, [45].



Slika 7: Prikaz meke margine i kažnjavanja netočno klasificiranih primjera

Optimizacijska ograničenja su:

$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi, \quad i = 1, \ldots, N. \tag{10}$$

Kažnjavanjem pojedinih primjera dopuštamo da vrijednost $y^{(i)}h(x)$ bude negativna ili manja od jedan za pogrešno klasificirane primjere. Cilj je maksimizirati marginu, ali i kazniti primjere s pogrešne strane margine. Stoga, potrebno je optimizirati

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i. \tag{11}$$

Parametar $C > 0$ je parametar kojim se pokušava održati balans između veličine margine i ukupne kazne. Što je veća vrijednost parametra C, veće je kažnjavanje i problem je

složeniji. Želimo pronaći vrijednosti (w^*, b^*) za koje se postiže minimum funkcije (11) uz ograničenje (10).

Dva opisana modela su vrlo jednostavna i vrlo često se ne mogu primijeniti na nelinearne probleme klasifikacije koji se javljaju u praksi. Nelinearni problemi se najčešće javljaju kada je broj primjera u trening skupu N puno veći od dimenzije prostora vektora m jer je tada ulazni prostor gusto popunjen i mala je vjerojatnost da su podaci linearno separabilni. Ideja kod rješavanja takvih problema je preslikati početni problem u prostor veće dimenzije u kojem se može postići linearna separabilnost, umjesto pretvorbe početnog problema u nelinearnu inačicu [4, 24].

3.4.2 SVM algoritam za prepoznavanje lica

SVM algoritam se kod prepoznavanja lica koristi se i za identifikaciju i za autentifikaciju. Kod identifikacije, algoritam kao ulazni podatak prima sliku nepoznate osobe i kao rezultat vraća najsličnije pojedince unutar baze podataka. Kod autentifikacije, algoritam kao ulazni podatak prima sliku s identitetom osobe za koju se tvrdi da je osoba na slici i algoritam radi provjeru, te ili prihvaća ili odbija predloženi identitet. Također može vratiti i mjeru pouzdanosti s kojom tvrdi da je osoba na slici.

Neka je slika osobe prikazana kao p-dimenzionalni vektor (možemo uzeti da je $p = N^2$ ako je slika dimenzija $N \times N$). Autentifikacija je problem binarne klasifikacije jer će algoritam ili prihvatiti ili odbiti predloženi identitet osobe. Jednostavna metoda za konstruiranje klasifikatora za osobu X je da se jedna klasa sastoji od slika lica osobe X, a druga klasa se sastoji od slika drugih osoba. Pišemo $p_1 \sim p_2$ ako su p_1 i p_2 slike istog lica, inače pišemo $p_1 \not\sim p_2$. Ako je problem linearno separabilan, SVM algoritam poistovjećuje lice p s identitetom osobe ako je

$$w \cdot p + b \leq 0,$$

pri čemu je w vektor težina, p vektor značajki ulazne slike i b pristranost. Točnost autentifikacije mjeri se s pomoću statistika: P_V je vjerojatnost točnih autentifikacija i P_F je vjerojatnost lažnih prihvaćanja. Želi se minimizirati rizik greške na skupu za treniranje i osigurati da algoritam dobro klasificira nove, neviđene podatke. Ekstremni slučajevi nastupaju kada se sve provjere odbacuju, pa niti jednom korisniku nije potvrđen identitet, ali isto tako nije došlo do lažnog prihvaćanja, odnosno vrijedi da su $P_V = P_F = 0$. Druga ekstremna situacija je ako su sve provjere prihvaćene, pa je svakom pravom korisniku potvrđen identitet, ali isto tako svakom neovlaštenom korisniku je potvrđen identitet, odnosno vrijedi da su $P_V = P_F = 1$. Potrebno je pronaći kompromis između odabira P_V i P_F i to najčešće ovisi o tome koliko se siguran sustav želi postići (veća vjerojatnost P_F i manja P_V) ili osigurati zadovoljstvo korisnika korištenjem tehnologije (veća vjerojatnost P_V). Kako bi se vrijednosti P_V i P_F mogle podešavati uvodimo parametar Δ u model tako da definiramo razdvajajuću hiperravninu kao:

$$w \cdot z + b = \Delta,$$

i lice p je prihvaćeno kao lice osobe čiji se identitet autentificira ako je:

$$wp + b \leq \Delta.$$

Manje vrijednosti parametara Δ ukazuju na strožu autentifikaciju, odnosno manje vrijednosti mjera P_V, P_F, dok veće vrijednosti parametra Δ ukazuju da je sustav autentifikacije blaži i vrijednosti mjera P_V, P_F su veće. Ako je $\Delta = -\infty$, vrijedi $P_V = P_F = 0$, odnosno sva lica p se odbijaju, a ako je $\Delta = \infty$, vrijedi $P_V = P_F = 1$, odnosno sva lica su prihvaćena. Za sve vrijednosti $\Delta \in \mathbb{R}$ testiraju se sve vrijednosti P_V i P_F. Najčešće se prepoznavanje lica vrši u visokodimenzionalnim prostoru gdje linearna separacija uglavnom nije moguća. Stoga, metoda potpornih vektora koristi jezgrenu funkciju i jezgren trik, kako bi početni problem preveli u problem preslikan u prostor više dimenzije u okviru kojeg je linearna separacija moguća [22]. Lice p se prihvaća kao lice osobe čiji identitet želimo autentificirati ako vrijedi:

$$\sum_{i=1}^{N_s} \alpha_i y^{(i)} k(s_i, p) + b \leq \Delta,$$

gdje su s_i potporni vektori za dani trening skup podataka $\mathcal{D} = \{x^{(i)}, y^{(i)}\}$, N_S ukupan broj potpornih vektora i α_i težine potpornih vektora kojima je definiran utjecaj tog potpornog vektora prilikom klasifikacije. Kod autentifikacije dan je skup podataka $\{g_j\}$ od m poznatih osoba. Algoritam kao ulazni podatak dobiva sliku lica p i identitet osobe za koju se tvrdi da je na slici. Prvi korak autentifikacije je računanje sličnosti:

$$\delta = \sum_{i=1}^{N_S} \alpha_i y^{(i)} k(s_i, g_j - p) + b.$$

Ako je $\delta \leq \Delta$, onda se slika lica prihvaća kao slika te osobe. Za razliku od autentifikacije, kod identifikacije osobe algoritam kao ulazni podatak dobiva samo sliku lica p. Zatim se izračunava sličnost između ulazne slike p i svake slike u skupu podataka g_j kao

$$\delta_j = \sum_{i=1}^{N_S} \alpha_i y^{(i)} k(s_i, g_j - p) + b.$$

Slika p se identificira kao slika osobe g_j za koju je mjera sličnosti δ_j minimalna [33].

4 Algoritmi za prepoznavanje $3D$ modela lica

Algoritmi za prepoznavanje $3D$ modela lica koriste trodimenzionalnu geometriju ljudskog lica i time imaju veći potencijal postizanja bolje točnosti nego algoritmi koji koriste $2D$ modele lica. Uporabom trodimenzionalne geometrije nadomještaju se nedostaci algoritama koji koriste $2D$ modele lica kao što su promjena osvjetljenja i sjene, različiti izrazi lica, orijentacija glave, šminka i slično. Opisani Eigenface algoritam koristi analizu glavnih komponenata za smanjenje dimenzionalnosti slike lica i reprezentaciju lica u obliku skupa vlastitih vektora. SVM klasificira lice pomoću hiperravnine koja optimalno razdvaja različita lica u prostoru značajki lica. Navedena ograničenja tih algoritama proizlaze iz ograničenih informacija o licu sadržanih u $2D$ slici. U radu [17] je pokazano da

se točnost prepoznavanja lica može poboljšati kombinirajući frontalni i profilni pogled. Glavni problemi s kojima se svi algoritmi za prepoznavanje $3D$ lica susreću su sinteza lica, animacije lica ili opće promjene u izgledu lica koje vode do problema lociranja određenih značajki na različitim licima ili problema razlikovanja stvarnog lica od lica koje nikad ne bi moglo biti lice u stvarnom svijetu. Također, algoritmi koji koriste $3D$ modele lica za prepoznavanje zahtijevaju trodimenzionalne ulazne podatke i veće baze podataka što je zahtjevnije i skuplje za provedbu. U nastavku rada opisani su algoritmi za prepoznavanje $3D$ modela lica. Prepoznavanje $3D$ lica bez rekonstrukcije oslanja se na dubinske podatke i informacije o obliku lica kako bi poboljšalo točnost prepoznavanja, a da pritom nije potrebno rekonstruirati trodimenzionalni model lica. Morfabilni modeli kombiniraju geometrijske i teksturalne informacije kako bi stvorili statističke modele lica koji mogu generirati nove varijacije lica i omogućiti preciznije prepoznavanje i analizu. Konvolucijske neuronske mreže koriste slojevitu arhitekturu za automatsko učenje značajki lica iz velikih skupova podataka, omogućujući visoku preciznost.

4.1 Prepoznavanje $3D$ lica bez rekonstrukcije lica

Algoritam prepoznavanja $3D$ lica bez rekonstrukcije lica tretira lice kao trodimenzionalnu površinu, odnosno stvara se detaljan $3D$ model lica. Međutim, rekonstrukcija $3D$ površine može biti računalno zahtjevna i stoga, umjesto rekonstrukcije cijele površine, algoritam se fokusira na *površinske gradijente*. Površinski gradijenti opisuju koliko je površina nagnuta ili zakrivljena u svakoj točki. Korištenjem površinskih gradijenata, algoritam i dalje može dohvatiti bitne značajke lica potrebne za prepoznavanje (kao što su oblik i konture) bez stvaranja potpunog $3D$ modela. Time se uravnotežuju točnost i učinkovitost što algoritam čini primjenjivim u stvarnom svijetu.

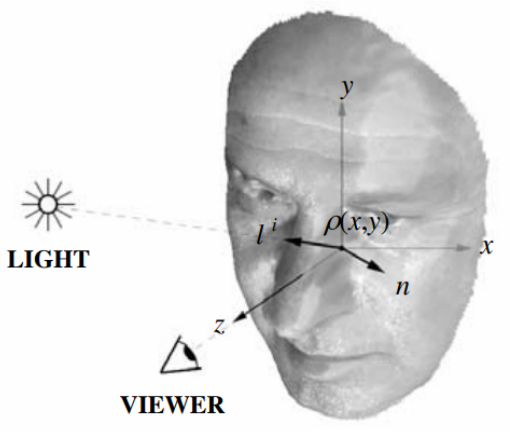
Fotometrijska stereotehnika sastoji se od dobivanja nekoliko slika istog predmeta pri različitim uvjetima osvjetljenja (smjer svjetla se promjeni kod svake slike). Model pretpostavlja da je površina lica Lambertova, odnosno ravnomjerno reflektira svjetlost u svim smjerovima. Ovo pojednostavljuje računanje jer je svjetlost na slikama izravno povezana s kutom između smjera svjetla i vektora normale površine. Površina lica predstavljena je kao funkcija koja obično pokazuje kako se z-koordinata mijenja preko $2D$ mreže razapete x i y koordinatama. Lambertov model refleksije koristi se za opisivanje načina kako se svjetlost odbija od površine koja jednoliko raspršuje svjetlost u svim smjerovima. Svjetlost ili intenzitet na slici proporcionalna je kosinusu kuta između smjera izvora svjetlosti i vektora normale površine, a također ovisi o reflektivnosti površine - albedu[10]. To je opisano jednadžbom:

$$I^i(x, y) = \rho(x, y) \cdot n(x, y) \cdot l^i, \tag{12}$$

gdje $\rho(x, y)$ mjeri koliko svjetlosti površina (lice) reflektira, $n(x, y)$ je vektor normale površine u zadanoj točki, l je vektor smjera svjetlosti koji pokazuje iz kojeg smjera svjetlost dolazi i indeks i ukazuje na $i = 1, \ldots, n$, promjena svjetlosti na fotografijama. Intenzitet opažen na slici u bilo kojoj točki je umnožak refleksije površine i toga koliko je normala

[10]mjera koja pokazuje koliko se svjetlosti reflektira s površine nekog tijela.

površine usklađena sa smjerom izvora svjetlosti. Na Slici 8, [6] je dana shema prikaza lica u koordinatnom sustavu.



Slika 8: Shema prikaza površine lica u koordinatnom sustavu

Vektor normale se računa kao

$$n(x,y) = \frac{(-z_x(x,y), -z_y(x,y), 1)}{\sqrt{1 + z_x(x,y)^2 + z_y(x,y)^2}},$$

gdje je $z(x,y)$ visina površine u točki (x,y), a z_x i z_y su gradijenti površine u x i y smjeru (z_x je parcijalna derivacija od $z(x,y)$ s obzirom na x, a $z_y(x,y)$ s obzirom na y). Gradijent u smjeru z-osi iznosi 1 jer je vektor normale okomit na površinu. Nazivnik osigurava da je vektor normale normiran. Jednadžba (12) može se izraziti i putem jednakosti:

$$I(x,y) = Lv, \tag{13}$$

pri čemu je

$$L = \begin{bmatrix} l_1^1 & l_2^1 & l_3^1 \\ \vdots & \vdots & \vdots \\ l_1^N & l_2^N & l_3^N \end{bmatrix} \quad \text{i} \quad v(x,y) = (v_1, v_2, v_3),$$

te

$$v_1 = -z_x v_3, \quad v_2 = -z_y v_3, \quad v_3 = \frac{\rho(x,y)}{\sqrt{1 + ||z||_2^2}}.$$

Potrebno je imati N, $N \geq 3$ različitih, međusobno nezavisnih vektora svjetlosti $\{l^i\}_{i=1}^N$ i popratne intenzitete slike $\{I^i\}_{i=1}^N$ kako bi se mogle rekonstruirati vrijednosti gradijenta ∇z. Gradijent ∇z pruža informacije kako je površina nagnuta u smjeru x i y osi u svakoj točki (x,y). Vrijednost gradijenta ∇z se određuje rješavajući jednadžbu

$$v = L^T(L^{-1})L^T I(x,y), \tag{14}$$

korištenjem metode najmanjih kvadrata. Cilj je odrediti vektor $v(x,y)$ koji odgovara promatranom intenzitetu $I(x,y)$ s obzirom na smjerove osvjetljenja L. Kada se odredi $\nabla z(x,y)$, može se izračunati i cijela površina $z(x,y)$.

Jedan od problema ovog modela je što lice može imati različite izraze, odnosno objekt je koji može mijenjati oblik. Kod fiksnih objekata kako bi se uskladile površine se najčešće koriste transformacije poput rotacije, translacije i skaliranja, no one nisu primjenjive kod objekata koji mijenjaju oblik poput lica. No, ipak se mogu koristiti izometrijske transformacije (ili druge transformacije koje čuvaju udaljenost), a površine koje nastaju takvim transformacijama nazivaju se *izometrijske površine*. Želi se stvoriti prikaz površine lica koji ostaje nepromjenjiv korištenjem izometrijskih transformacija, što bi značilo da bi reprezentacija lica ostala jednaka bez obzira na izraz i ekspresiju, a to bi omogućilo uporabu jednostavnijih algoritama za prepoznavanje. Pretpostavljamo da je dana poliedarska aproksimacija površine lica, S. Poliedarska aproksimacija površine lica dana je konačnim skupom točaka p_i, $i = 1, \ldots, n$, i za svake dvije točke na površini određena je udaljenost između njih

$$\delta(p_i, p_j) = \delta_{ij}.$$

Ako se vrijednosti δ_{ij} zapišu u obliku matrice, dobiva se matrica Δ svih međusobnih udaljenosti. Radi jednostavnosti, promatraju se kvadrati svih udaljenosti, odnosno vrijedi $\Delta_{ij} = \delta_{ij}^2$. Matrica Δ je invarijantna za transformacije izometrijske površine, što znači da udaljenosti ostaju jednake ako se lice deformira, ali nije jedinstvena jer ovisi o proizvoljnom odabiru točaka na licu i njihovom redoslijedu. Ako se kvadrati udaljenosti promatraju kao razlike, može se primijeniti tehnika *višedimenzionalnog skaliranja* (eng. *multidimensional scalling, MDS*) za smanjenje dimenzionalnosti. Cilj je preslikati točke iz izvornog višedimenzionalnog prostora $3D$ površine lica u nižedimenzionalni Euklidski prostor tako da udaljenosti između točaka u novom prostoru odražavaju izvorne udaljenosti na površini. To je ekvivalentno definiranju funkcije između dva metrička prostora:

$$\varphi \,:\, (S, \delta) \,\rightarrow\, (R^m, d), \qquad \varphi(p_i) = x_i.$$

Cilj je odrediti φ tako da vrijedi

$$\delta(p_i, p_j) \sim d(x_i, x_j).$$

Greške prilikom preslikavanja, ϵ_{ij}, označavaju koliko su udaljenosti u preslikanom prostoru usklađene s udaljenostima na površini lica, vrijedi

$$\epsilon_{ij} = |\delta_{ij} - d_{ij}|.$$

Ukupna greška je definirana kao $\epsilon = f(\epsilon_{ij})$, gdje je f monotona funkcija koja sumira po svim vrijednostima i, j. Jedan od najjednostavnijih primjera višedimenzionalnog skaliranja je *klasično skaliranje*. Kod klasičnog skaliranja matrica Δ se dvostruko centrira na način da se odredi matrica

$$B = -\frac{1}{2} J \Delta J, \qquad J = I - \frac{1}{n} U,$$

gdje je I jedinična $n \times n$ matrica i U je $n \times n$ matrica čiji su svi elementi jedinice. Prvih m svojstvenih vektora e_i, kojima su pridružene najveće svojstvene vrijednosti matrice B, određuju vektore u novom prostoru, odnosno vrijedi $x_i^j = e_i^j,\ i = 1, \ldots, n,\ j = 1, \ldots, m$, gdje je $x_i^j,\ j$-ta koordinata vektora x_i. Skup točaka $\{x_i\}_{i=1}^n$ naziva se *invarijantni kanonski*

oblik površine. Procjena udaljenosti na površini se određuje pomoću gradijenta ∇z koji opisuje nagib ili promjenu visine točaka na površini.

Algoritam prepoznaje lice na način da se prvo odredi gradijent ∇z koji ukazuje na sve zakrivljenosti i promjene u visini površine lica. Lice je centrirano te se izrezuju nebitni ili promjenjivi dijelovi, a zatim se izdvajaju konture lica kako bi se obradila samo površina lica. Nakon toga se određuje n točaka na površini lica i njihove udaljenosti te se provodi višedimenzionalno skaliranje kako bi se smanjila dimenzionalnost prostora, ali očuvala udaljenost odabranih točaka na površini lica. Dobiveni invarijantni kanonski oblik površine uspoređuje se s bazom podataka i lice se prihvaća kao valjano ili se odbija ako su razlike između kanonskog oblika i predložaka u bazi veće od unaprijed definiranog praga tolerancije prihvaćanja lica [6].

4.2 Morfabilni model

Morfabilni model omogućuje stvaranje proizvoljnih ljudskih lica uz kontrolu vjerojatnosti generiranih lica te izračunava korespondenciju između novih lica. Korištenjem velikog broja pohranjenih $3D$ skenova lica, model se temelji na morfiranju[11] i koristi metode klasifikacije uzorka kako bi stekao znanje o varijacijama lica. U modelu su sva lica poravnana na način da svaka točka na jednom licu odgovara sličnoj točki na drugom licu. Na početku je dan skup primjera lica koji čini zajednički predložak za stvaranje poravnanja i kada se ono uspostavi, ostaje jednako u svim kasnijim koracima algoritma. To omogućuje da kasnije kombiniranje različitih lica djeluje prirodno jer su sva lica poravnata kako bi se međusobno slagala po obliku. Druga ideja modela je odvajanje oblika i boje lica od drugih čimbenika kao što su osvjetljenje i kutevi kamere. To omogućuje manipulaciju nad oblikom i bojom što daje preciznije i realističnije promjene u licu. Morfabilni model lica je višedimenzionalna $3D$ funkcija morfiranja koja se temelji na linearnoj kombinaciji velikog broja $3D$ skenova lica. Izračunavanjem prosječnog lica i glavnih izvora varijacije za dani skup podataka, distribucija vjerojatnosti primjenjena je na funkciju morfiranja kako bi se izbjeglo generiranje nerealnih lica. To je parametarski model lica koji je sposoban generirati bilo koje lice, a problem korespondencije postaje optimizacijski problem. Novo lice, bilo da se radi o slici ili $3D$ skenu lica, usklađuje se s modelom minimiziranjem razlika između lica i njegove rekonstrukcije pomoću funkcije modela lica. Morfabilni $3D$ model lica predstavlja naprednu verziju tehnike interpolacije između geometrija lica, koja je prvi put uvedena u radu [30]. Automatskim izračunavanjem korespondencija između pojedinačnih $3D$ podataka o licu, povećava se broj značajki korištenih u prikazu lica s nekoliko stotina na desetke tisuća. Također, koristi se veći broj lica, što omogućuje interpolaciju između stotina "osnovnih" lica, umjesto samo nekoliko. Cilj ovog proširenog morfabilnog modela lica je prikazati bilo koje lice kao linearnu kombinaciju ograničenog skupa primjera lica.

Morfabilni model temelji se na skupu podataka koji se naziva skup primjera lica i koji se sastoji od $3D$ skenova lica. Morfiranje između lica zahtjeva korespondenciju između svih lica. Geometrija lica je opisana vektorom oblika

$$S = (X_1, Y_1, Z_1, X_2, Y_2, Z_2, \ldots, X_n, Y_n, Z_n) \in \mathbb{R}^{3n}$$

[11] obrada slike pomoću računalnih animacija.

koji sadrži $3D$ koordinate (X, Y, Z) odabranih n značajki koje definiraju oblik lica i vektorom teksture

$$T = (R_1, G_1, B_1, \ldots, R_n, G_n, B_n) \in \mathbb{R}^{3n}$$

koji sadrži RGB[12] vrijednosti za svaku od n značajki. Radi jednostavnosti pretpostavljamo da je svakoj od n značajki pridružen jedan vektor teksture. Morfabilni model sastoji se od m lica u bazi podataka koja čine primjere lica. Svako lice u bazi je predstavljeno vektorom oblika S_i i vektorom teksture T_i. Novo lice je opisano vektorima oblika S_m i teksture T_m. Oblik novog lica dobiva se linearnom kombinacijom vektora oblika m lica iz skupa primjera lica, a tekstura se dobiva linearnom kombinacijom vektora teksture m lica iz skupa primjera. To je moguće jer su sva lica u potpunoj korespondenciji, odnosno jednako su poravnana te svaka točka jednog lica odgovara jednakoj točki na drugom licu. Stoga, vrijedi:

$$S_m = \sum_{i=1}^{m} a_i S_i, \qquad T_m = \sum_{i=1}^{m} b_i T_i. \tag{15}$$

Koeficijenti (težine) a_i i b_i određuju koliko svako lice u bazi doprinosi obliku i teksturi novog lica, stoga zbroj težina iznosi 1, to jest

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} b_i = 1.$$

Morfabilni model se definira kao skup svih mogućih lica $(S_m(a), T_m(b))$ parametriziranih vektorima $a = (a_1, \ldots, a_n)^T$ i $b = (b_1, \ldots, b_n)^T$. Mijenjanjem vrijednosti koeficijenata a_i, b_i, $i = 1, \ldots, m$, stvaraju se proizvoljna nova lica promjenom vektora oblika i teksture iz skupa primjera lica. No, ne daju sve kombinacije koeficijenata a_i i b_i realistična lica. Kako bi se osiguralo da generirana lica izgledaju kao uvjerljiva ljudska lica, potrebno je kontrolirano odabrati koeficijente. Model procjenjuje distribuciju vjerojatnosti za koeficijente a_i i b_i i na temelju skupa primjera lica. Ova distribucija vjerojatnosti odražava koliko su vjerojatne određene kombinacije koeficijenata, na temelju karakteristika stvarnih lica u skupu podataka. Model određuje prosječno lice računajući prosječnu vrijednost vektora oblika $\bar{S}$ i vektora teksture $\bar{T}$. Za svako lice i u bazi podataka izračuna se razlika između vektora oblika i teksture i prosječnog vektora oblika i teksture za cijelu bazu kao

$$\Delta S_i = S_i - \bar{S}, \qquad \Delta T_i = T_i - \bar{T}, \qquad i = 1, \ldots, m,$$

koje ukazuju koliko svako pojedinačno lice odstupa od prosjeka. Na temelju razlika ΔS_i i ΔT_i računaju se matrice kovarijance C_S i C_T. Na temelju prosječnih vrijednosti $\bar{S}$ i $\bar{T}$ i matrica kovarijance C_S i C_T skupu podataka je prilagođena multivarijatna normalna distribucija[13] [29]. Ona daje vjerojatnosni model koji opisuje kako su značajke oblika i teksture

[12]RGB (eng. *Red, Green, Blue.*) - sustav koji predstavlja boje koje se koriste na digitalnom zaslonu.

[13]Multivarijatna normalna distribucija je generalizacija normalne (Gaussove) distribucije na više dimenzija. Opisuje distribuciju vektora slučajnih varijabli koje imaju normalnu razdiobu. Formalno, vektor $X = (X_1, \ldots, X_n)^T$ prati n-dimenzionalnu normalnu distribuciju s očekivanjem μ i kovarijacijskom matricom Σ, što pišemo $X \sim \mathcal{N}(\mu, \Sigma)$ ako mu je funkcija gustoće jednaka

$$f(x|\mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(y-\mu)\right).$$

lica raspoređene u bazi podataka i na temelju toga se određuju koeficijenti a_i i b_i kojima se generiraju nova lica. S obzirom na to da su vektori oblika i teksture $3n$-dimenzionalni, provodi se PCA analiza kako bi se smanjila dimenzionalnost problema. Vektori oblika i teksture transformiraju se u ortogonalni koordinatni sustav definiran svojstvenim vektorima s_i i t_i za oblik i teksturu matrica kovarijanci C_S i C_T, respektivno. Svojstveni vektori su poredani prema vrijednostima svojih odgovarajućih svojstvenih vrijednosti, od najveće prema najmanjoj. Tada je

$$S_m = \bar{S} + \sum_{i=1}^{m-1} \alpha_i s_i \quad \text{i} \quad T_m = \bar{T} + \sum_{i=1}^{m-1} \beta_i t_i, \quad \alpha, \beta \in \mathbb{R}^{m-1}, \tag{16}$$

gdje su α_i, β_i koeficijenti (težine) svojstvenih vektora oblika i težine, respektivno. Kako je $\sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{m} \beta_i = 1$, vrijedi da je samo prvih $m-1$ koeficijenata linearno nezavisno. Vjerojatnost da je odabran vektor koeficijenata α određena je kao:

$$p(\alpha) \sim \exp\left[-\frac{1}{2}\sum_{i=1}^{m-1}\left(\frac{\alpha_i}{\sigma_i}\right)^2\right],$$

pri čemu je σ_i^2 svojstvena vrijednost pridružena svojstvenom vektoru s_i matrice kovarijance C_S. To ukazuje da je vjerojatnije da će koeficijent α_i koji odgovara većoj svojstvenoj vrijednosti σ_i^2 poprimiti veću vrijednost, dok su koeficijenti pridruženi manjim svojstvenim vrijednostima manje vjerojatni. Vjerojatnost $p(\beta)$ određuje se analogno [5].

Generiranje $3D$ modela lica koji promatra varijacije vektora oblika lica S i vektora teksture lica T može se opisati afinim modelom:

$$\begin{aligned} S = S(\alpha, \beta) &= \bar{S} + \alpha B_{id} + \beta B_{exp} \\ T = T(\delta) &= \bar{T} + \delta B_t, \end{aligned} \tag{17}$$

gdje su $\bar{S}, \bar{T}$ prosječni vektori oblika i teksture i B_{id}, B_{exp}, B_t su baze vektora dobivene provedenom PCA analizom za početan skup podataka (identiteta), oblik (ekspresiju) i teksturu. Ove baze bilježe glavne varijacije u obliku i teksturi promatrane u skupu podataka. Koeficijenti α, β i δ određuju koliko početna slika, promjena ekspresije i teksture utječe na generirano lice. Promjenom koeficijenata dobivaju se nova lica unutar modela. Položaj p lica u $3D$ prostoru je definirana rotacijom r i translacijom $t \in \mathbb{R}^3$ koje ukazuju kako je lice orijentirano i postavljeno u prostoru. Uklapanjem $3D$ modela na ulaznu $2D$ sliku lica možemo procijeniti položaj lica na $2D$ slici jer će prilikom uklapanja biti utvrđeni parametri poze, uz ranije definirane projektivne parametre α_i za oblik lica. Projektivni parametri - žarišna daljina f kamere i translacija t te parametri položaja - kutevi rotacije oko vertikalne, ϕ, horizontalne, θ i osi kamere, γ definiraju vektor položaja lica. Sve značajke lica preslikavaju se u koordinatni sustav kamere pomoću transformacija rotacije i translacije. k-ta značajka lica $X_k = (x_k, y_k, z_k l)^T$ preslika se u točku $W_k = (w_1, w_2, w_3)^T$ transformacijom:

$$(W_1, W_2, W_3)^T = R_\gamma R_\theta R_\phi X_k + t,$$

gdje su R_ϕ i R_θ rotacije oko vertikalne i horizontalne osi, R_γ rotacija oko osi kamere, a t translacija koja translatira točku u koordinatni sustav kamere. Nakon što je k-ta značajka

lica preslikana u koordinatni sustav kamere, potrebno ju je projicirati na $2D$ ravninu slike kako bi bila vidljiva na slici. To se radi pomoću perspektivne projekcije:

$$p_1 = M_1 + f\frac{w_1}{w_3}$$
$$p_2 = M_2 + f\frac{w_2}{w_3},$$

pri čemu je f žarišna duljina fotoaparata koja kontrolira razinu zumiranja, w_1, w_2, w_3 su koordinate značajke u koordinatnom sustavu kamere. Dodatno, (M_1, M_2) je glavna točka u kojoj optička os siječe ravninu slike i obično se nalazi u središtu slike, a (p_1, p_2) su $2D$ koordinate značajke lica u ravnini.

Prednosti morfabilnog modela su te što kombiniranjem oblika i teksture stvara više predložaka lica i time generira veći broj lica za usporedbu i prepoznavanje. Model može simulirati kako lice izgleda pod različitim uvjetima osvjetljenja, čineći ga manje osjetljivim na promjene osvjetljenja, nego korištenje $2D$ modela. Također, promatra se $3D$ geometrija lica pa model može uočiti više detalja, što je bitno za razlikovanje sličnih lica. Neki od nedostataka modela su što uklapanje morfabilnog modela u $2D$ sliku ili $3D$ skeniranje lica može biti računalno zahtjevno i dugotrajno, posebno za slike i skenove visoke rezolucije. Model zahtjeva visokokvalitetne ulazne podatke za točno prilagođavanje pa slike ili skenovi niske rezolucije mogu dovesti do lošeg uklapanja modela, što rezultira netočnim prepoznavanjem. Također, izrada modela zahtijeva velik skup $3D$ skenova lica, što nije lako i brzo izvedivo. Iako je $3D$ model, morfabilni model još uvijek nije toliko dobro prilagođen prepoznavanju lica kada je ono djelomično zaklonjeno (duga kosa, brada, nošenje naočala, maske i slično) [5, 9].

4.3 Konvolucijske neuronske mreže

4.3.1 Neuronske mreže

Neuronska mreža (eng. *Neural Network, NN*) je model strojnog učenja koji se sastoji od međusobno povezanih čvorova koji se nazivaju *neuroni*. Dobila je ime jer se vezama između neurona žele oponašati sinapse neurona u mozgu. Neuroni u neuronskoj mreži su agregirani u slojeve, a slojevi su međusobno povezani. Prvi sloj se naziva *ulazni sloj*, posljednji se naziva *izlazni sloj*, a svi slojevi između njih *skriveni slojevi*. Što je veći broj slojeva i neurona u svakom sloju, to je neuronska mreža složenija. Neuronske mreže koje imaju više od dva skrivena sloja nazivaju se *duboke neuronske mreže*. Neuron u neuronskoj mreži prima podatke od neurona prethodnog sloja (jednog ili više njih), a izlaz se izračunava primjenom neke funkcije koja se naziva *aktivacijska funkcija*. Težine određuju koliki je utjecaj pojedinog neurona prethodnog sloja na neuron u novom sloju. Cilj treniranja modela putem neuronskih mreža je određivanje vrijednosti težina za sve neuronske slojeve. Ako je izbor početnih vrijednosti dobar, odnosno ako se za te vrijednosti se dobiva ispravna izlazna vrijednost, stvarni parametri se zadržavaju i prelazi se na sljedeći unos. Međutim, ako dobiveni izlaz nije jednak željenoj vrijednosti, težine se korigiraju tijekom procesa treniranja. Kako bi se odredilo kako i koju težinu modificirati, provodi se proces *širenja unatrag ili backpropagacije* (eng. *backpropagation*). Ideja backpropagacije je

vraćanje unatrag od izlaznog do početnog sloja. Uspoređuje se unos svakog sloja s greškom sloja i mijenja se vrijednost odgovarajuće težine tako da se smanji svaka pojedinačna greška na sloju [28, 42].

Neuronske mreže su model nadziranog učenja. Pretpostavimo da je za izlaznu vrijednost y i ulazne podatke x određena hipoteza $h_\theta(x)$. Neka je zadan skup označenih primjera $\mathcal{D} = \{(x^{(i)}, y^{(i)}) \ ; \ i = 1, \ldots, n\}$. Funkcija gubitka i-tog trening primjera je

$$J^{(i)}(\theta) = \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2,$$

a srednjekvadratna funkcija gubitka je definirana kao

$$J(\theta) = \frac{2}{n}\sum_{i=1}^{n} J^{(i)}(\theta).$$

Pretpostavimo da se neuronska mreža sastoji od jednog neurona. Neka je $x \in \mathbb{R}^d$ ulazni podatak, y izlazni podatak i neka je neuronska mreža parametrizirana s θ. Jedna od najjednostavnijih parametrizacija je oblika

$$h_\theta(x) = \max\{wx + b, 0\}, \ \ w \in \mathbb{R}^d, \ b \in \mathbb{R}, \ \theta = (w, b). \tag{18}$$

b se naziva *pristranost* (eng. *bias*), a vektor w je *vektor težina*. Takva neuronska mreža ima jedan sloj. Funkcija $\mathbb{R} \to \mathbb{R}$ definirana kao $\max\{t, 0\}$ naziva se ReLU (eng. *Rectified Linear Unit*) ili *funkcija jedinične rampe* i definirana je kao

$$\mathrm{ReLU}(t) = \max\{t, 0\}.$$

ReLU funkcija je primjer aktivacijske funkcije i jedna je od najčešće korištenih aktivacijskih funkcija. Stoga, (18) možemo pisati kao

$$h_\theta(x) = \mathrm{ReLU}\{wx + b, 0\}, \ \ w \in \mathbb{R}^d, \ b \in \mathbb{R}, \ \theta = (w, b). \tag{19}$$

Složenija neuronska mreža može se konstruirati tako da koristi pojedinačne neurone i slaže ih zajedno tako da jedan neuron proslijedi svoj izlaz kao ulaz u sljedeći neuron, što rezultira složenijim modelom.

Radi jednostavnosti, neka je za početak zadana potpuno povezana neuronska mreža s dva skrivena sloja, s ulaznim vektorom $x \in \mathbb{R}^d$ i m neurona u skrivenom sloju. Svakom neuronu u skrivenom sloju je pridružen težinski vektor $w_j^{[1]}$ koji određuje koliko svaki ulazni podatak doprinosi u mreži i pristranost $b_j^{[1]}$ koji omogućuje fleksibilnost modela. Za svaki skriveni neuron računa se linearna kombinacija ulaznih vrijednosti

$$z_j = w_j^{[1]}x + b_j^{[1]}, \quad w_j^{[1]} \in \mathbb{R}^d, \ b_j^{[1]} \in \mathbb{R},$$

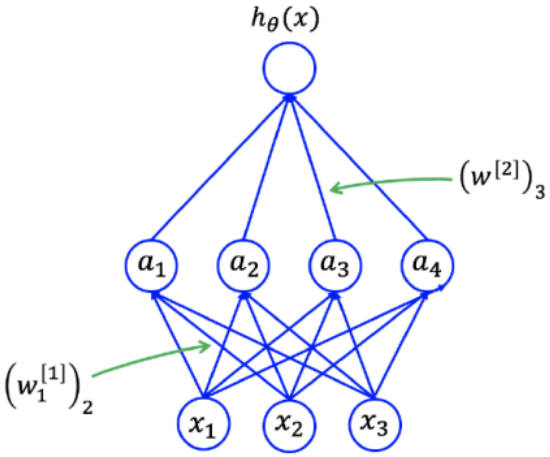
i na vrijednost linearne kombinacije se primjenjuje ReLU funkcija

$$a_j = \mathrm{ReLU}(z_j).$$

Vektor $a = [a_1 \ a_2 \ \ldots \ a_m]^T \in \mathbb{R}^m$ je vektor izlaznih vrijednosti neurona skrivenog sloja. Izlazni sloj ima težinski vektor $w^{[2]} \in \mathbb{R}^m$ koji kombinira izlazne vrijednosti neurona skrivenog sloja kako bi dobio vrijednost konačnog izlaza mreže:

$$h_\theta(x) = w^{[2]}a + b^{[2]}, \quad w^{[2]} \in \mathbb{R}^m, \ \ b^{[2]} \in \mathbb{R}, \ \forall j \in \{1, \ldots, m\}.$$

Na Slici 9, [24] prikazana je jednostavna, potpuno povezana neuronska mreža.



Slika 9: Grafički prikaz potpuno povezane dvoslojne neuronske mreže, ulaznih vektora, skrivenih neurona i pripadnih težina

Neuronske mreže se najčešće sastoje od velikog broja skrivenih slojeva koji su međusobno povezani s velikim brojem veza te su ulazni podaci visokodimenzionalni, stoga se javlja potreba za vektorizacijom procesa kako se sve težine i aktivacijske funkcije ne bi računale postupno. Vektorizacija koristi matrični zapis i matrične operacije kako bi se izračuni ubrzali. Potpuno povezanu dvoslojnu neuronsku mrežu u matričnom obliku zapisujemo:

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} - - & w_1^{[1]^T} & - - \\ - - & w_2^{[1]^T} & - - \\ & \vdots & \\ - - & w_m^{[1]^T} & - - \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ \vdots \\ b_m^{[1]} \end{bmatrix}, \tag{20}$$

ili

$$z = W^{[1]}x + b^{[1]}. \tag{21}$$

Vektor skrivenog sloja i izlazna vrijednost neuronske mreže su tada definirani kao:

$$\begin{aligned} a &= \mathrm{ReLU}(W^{[1]}x + b^{[1]}) \\ h_\theta(x) &= W^{[2]}a + b^{[2]}. \end{aligned} \tag{22}$$

Potpuno povezana neuronska mreža s r slojeva se tada definira kao:

$$\begin{aligned} a^{[1]} &= \mathrm{ReLU}(W^{[1]}x + b^{[1]}) \\ a^{[2]} &= \mathrm{ReLU}(W^{[2]}a^{[1]} + b^{[2]}) \\ &\vdots \\ a^{[r-1]} &= \mathrm{ReLU}(W^{[r-1]}a^{[r-2]} + b^{[r-1]}) \\ h_\theta(x) &= W^{[r]}a^{[r-1]} + b^{[r]}, \end{aligned} \tag{23}$$

gdje su $W^{[1]}, W^{[2]}, \ldots, W^{[r]}$ matrice težina i $b^{[1]}, b^{[2]}, \ldots, b^{[r]}$ pristranosti. Umjesto aktivacijske funkcije ReLU mogu se koristiti i druge aktivacijske funkcije poput sigmoidne funkcije, σ[14] ili tangensa hiperbolnog, th[15]. Nakon definiranja strukture neuronske mreže (određivanja broja slojeva, broja neurona u svakom sloju i aktivacijske funkcije koja se koristi), potrebno je provjeriti efikasnost, odnosno točnost neuronske mreže i po potrebi korigirati težine u cilju njenog poboljšanja. U tome nam pomaže korak koji nazivamo backpropagacija. Funkcija gubitka mjeri koliko su predikcije mreže $h_\theta(x)$ različite od stvarnih predikcija, a algoritam backpropagacije je algoritam pomoću kojeg se tijekom treniranja modela korigiraju težine kako bi greška predviđanja bila što je moguće manja. Neka je $o = h_\theta(x)$, pa tada vrijedi da je $J = \frac{1}{2}(y-o)^2$ funkcija gubitka. S obzirom da je potrebno odrediti minimum funkcije gubitka, određujemo derivaciju funkcije J u ovisnosti o parametru θ. Ako je zadana neuronska mreža od jednog neurona, odnosno

$$\begin{aligned} z &= w^T x + b, \\ o = h_\theta(x) &= \mathrm{ReLU}(z), \\ J &= \frac{1}{2}(o-y)^2, \end{aligned}$$

vrijedi

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial o}\frac{\partial o}{\partial z} = \frac{\partial J}{\partial o}\frac{\partial o}{\partial z}\frac{\partial z}{\partial w_i}.$$

Kako je

$$\frac{\partial J}{\partial o} = o - y, \quad \frac{\partial o}{\partial z} = \mathrm{ReLU}'(z), \quad \frac{\partial z}{\partial w_i} = x_i,$$

vrijedi

$$\frac{\partial J}{\partial w_i} = (o-y)\mathrm{ReLU}'(z)\cdot x_i,$$

što se u vektorskom obliku može zapisati kao

$$\nabla_w J = (o-y)\mathrm{ReLU}'(z)\cdot x.$$

Također, vrijedi

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial o}\frac{\partial o}{\partial z}\frac{\partial z}{\partial b} = (o-y)\cdot \mathrm{ReLU}'(z).$$

Time je određen gradijent funkcije gubitka.

Za optimizaciju funkcije gubitke $J(\theta)$ mogu se koristiti razni pristupi, a jedan od najčešće korištenih je algoritam gradijentnog spusta [18], koji zapisujemo

$$\theta := \theta - \alpha \nabla_w J(\theta),$$

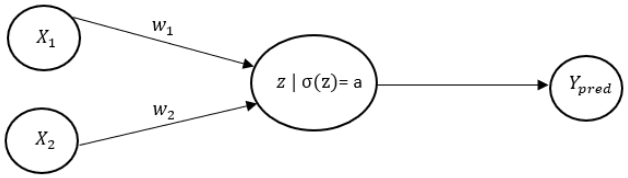
gdje je α stopa učenja. Analogno se određuje gradijent funkcije gubitka kod neuronske mreže s više slojeva [24].

Primjer 4.3.1.1. Neka je zadana jednostavna neuronska mreža, s jednim skrivenim

[14] Sigmoidna funkcija je definirana kao $\sigma(z) = \frac{1}{1+e^{-z}}$.

[15] Tangens hiperbolni je definiran kao $\mathrm{th}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$.

slojem s jednim neuronom i dva ulazna podatka kao što je prikazano na Slici 10. Pretpostavimo da ulazni podaci poprimaju vrijednosti $x_1 = \{1, 2, 3\}$, $x_2 = \{4, 5, 6\}$, a izlazni podatak Y poprima vrijednosti $y = \{5, 7, 9\}$, odnosno vrijedi $Y = X_1 + X_2$. Neka je dan vektor težina $w = (w_1, w_2) = (0.5, 0.4)$, pristranost $b = 0.1$, stopa učenja za algoritam gradijentnog spusta je $\alpha = 0.1$ te je kao aktivacijska funkcija dana sigmoidna funkcija $\sigma(z) = \frac{1}{1+e^{-z}}$.



Slika 10: Prikaz neuronske mreže s dva ulazna podatka i jednim skrivenim slojem koji se sastoji od jednog neurona

Potrebno je provesti algoritam backpropagacije kako bi se pripadne težine korigirale s obzirom na predviđene izlazne vrijednosti, y_{pred} i ispravne izlazne vrijednosti, y_{true}. U početnom koraku računa se linearna kombinacija ulaznih vrijednosti i pripadnih težina s obzirom na zadanu pristranost, odnosno

$$z = w_1 x_1 + w_2 x_2 + b.$$

Dobivamo vrijednosti:

$$\begin{aligned} \text{za} \quad x_1 = 1,\ x_2 = 4 \quad \text{vrijedi} \quad z_1 &= 1 \cdot 0.5 + 4 \cdot 0.4 + 0.1 = 2.2, \\ \text{za} \quad x_1 = 2,\ x_2 = 5 \quad \text{vrijedi} \quad z_2 &= 2 \cdot 0.5 + 5 \cdot 0.4 + 0.1 = 3.1, \\ \text{za} \quad x_1 = 3,\ x_2 = 6 \quad \text{vrijedi} \quad z_3 &= 3 \cdot 0.5 + 6 \cdot 0.4 + 0.1 = 4. \end{aligned}$$

Zatim se na neuron skrivenog sloja primjenjuje aktivacijska funkcija i dobivamo predikcije izlaznih vrijednosti:

$$y_{pred_1} = \sigma(z_1) = 0.9, \quad y_{pred_2} = \sigma(z_2) = 0.9569, \quad y_{pred_3} = \sigma(z_3) = 0.982.$$

Srednjekvadratna funkcija gubitka tada iznosi:

$$J(\theta) = \frac{2}{3} \sum_{i=1}^{3} (y_{true} - y_{pred})^2 = \frac{2}{3} \left[(5 - 2.2)^2 + (7 - 3.1)^2 + (9 - 4)^2 \right] = \frac{2}{3} \cdot 48.05 = 32.03.$$

Dobivena srednjekvadratna greška je dosta velika pa se metoda backpropagacije koristi kako bi se korigirale težine i smanjila razlika između predviđenih i stvarnih promatranih vrijednosti. Ta razlika je definirana funkcijom gubitka, stoga je potrebno odrediti njen minimum. U tu je svrhu potrebno odrediti vrijednosti parcijalnih derivacija funkcije gubitka po w_1, w_2 i b. Vrijedi

$$\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial \sigma} \frac{\partial \sigma}{\partial z} \frac{\partial z}{\partial w_1}, \qquad \frac{\partial J}{\partial w_2} = \frac{\partial J}{\partial \sigma} \frac{\partial \sigma}{\partial z} \frac{\partial z}{\partial w_2}, \qquad \frac{\partial J}{\partial b} = \frac{\partial J}{\partial \sigma} \frac{\partial \sigma}{\partial z} \frac{\partial z}{\partial b}.$$

Kako je

$$\frac{\partial J}{\partial z} = \frac{1}{3}\sum_{i=1}^{3}(y_{true} - y_{pred}),$$

$$\frac{\partial \sigma}{\partial z} = \left(\frac{1}{1+e^{-z}}\right)' = e^{-z}\cdot\frac{1}{(1+e^{-z})^2} = \sigma(z)\cdot(1-\sigma(z)) = y_{pred}\cdot(1-y_{pred}),$$

$$\frac{\partial z}{\partial w_1} = x_1, \qquad \frac{\partial z}{\partial w_2} = x_2, \qquad \frac{\partial z}{\partial b} = 1,$$

dobivamo

$$\frac{\partial J}{\partial w_1} = \frac{1}{3}\sum_{i=1}^{3}(y_{true} - y_{pred})\cdot y_{pred}\cdot(1-y_{pred})\cdot x_1,$$

$$\frac{\partial J}{\partial w_2} = \frac{1}{3}\sum_{i=1}^{3}(y_{true} - y_{pred})\cdot y_{pred}\cdot(1-y_{pred})\cdot x_2,$$

$$\frac{\partial J}{\partial b} = \frac{1}{3}\sum_{i=1}^{3}(y_{true} - y_{pred})\cdot y_{pred}\cdot(1-y_{pred}).$$

Uvrštavanjem vrijednosti dobivamo:

$$\frac{\partial J}{\partial w_1} = \frac{1}{3}[(0.9-5)\cdot 0.9\cdot(1-0.9)\cdot 1 + (0.9569-7)\cdot 0.9569\cdot(1-0.9569)\cdot 2+$$

$$(0.982-9)\cdot 0.982\cdot(1-0.982)\cdot 3] = \frac{1}{3}(-0.3697-0.4985-1.7) = \frac{1}{3}\cdot(-2.5675) = -0.856,$$

$$\frac{\partial J}{\partial w_2} = -1.19, \qquad \frac{\partial J}{\partial b} = 0.253.$$

Ažuriraju se vrijednosti težina i koeficijenta pristranosti (pomaka). Nove vrijednosti parametara dobivene algoritmom backpropagacije su:

$$w_1 = w_1 - \alpha\frac{\partial J}{\partial w_1} = 0.5 - 0.1\cdot(-0.856) = 0.5856,$$

$$w_2 = w_2 - \alpha\frac{\partial J}{\partial w_2} = 0.4 - 0.1\cdot(-1.19) = 0.519,$$

$$b = b - \alpha\frac{\partial J}{\partial b} = 0.1 - 0.1\cdot(-0.253) = 0.1253.$$

Za dobivene težine w_1, w_2 i pristranost b ponavljamo cijeli postupak. Dobivamo nove vrijednosti:

$$\text{za} \quad x_1 = 1,\ x_2 = 4 \quad \text{vrijedi} \quad z_1 = 1\cdot 0.5856 + 4\cdot 0.519 + 0.1253 = 2.7869,$$

$$\text{za} \quad x_1 = 2,\ x_2 = 5 \quad \text{vrijedi} \quad z_2 = 2\cdot 0.5856 + 5\cdot 0.519 + 0.1253 = 3.8915,$$

$$\text{za} \quad x_1 = 3,\ x_2 = 6 \quad \text{vrijedi} \quad z_3 = 3\cdot 0.5856 + 6\cdot 0.519 + 0.1253 = 4.9961.$$

Vrijednosti nakon primjene aktivacijske funkcije su:

$$y_{pred_1} = \sigma(z_1) = 0.942, \quad y_{pred_2} = \sigma(z_2) = 0.98, \quad y_{pred_3} = \sigma(z_3) = 0.9933,$$

a vrijednost srednjekvadratne funkcije gubitka je:

$$J(\theta) = \frac{2}{3}\sum_{i=1}^{3}(y_{true} - y_{pred})^2 = \frac{2}{3}\left[(5 - 2.7869)^2 + (7 - 3.8915)^2 + (9 - 4.9961)^2\right] = \frac{2}{3} \cdot 30.592 = 20.395.$$

Možemo primjetiti da se vrijednost srednjekvadratne funkcije gubitka smanjila nakon prve korekcije težina i pristranosti. Nove vrijednosti težina su:

$$\frac{\partial J}{\partial w_1} = -0.208, \quad \frac{\partial J}{\partial w_2} = -0.602, \quad \frac{\partial J}{\partial b} = -0.1311,$$
$$w_1 = w_1 - \alpha\frac{\partial J}{\partial w_1} = 0.5856 - 0.1 \cdot (-0.208) = 0.6064,$$
$$w_2 = w_2 - \alpha\frac{\partial J}{\partial w_2} = 0.519 - 0.1 \cdot (-0.602) = 0.5792,$$
$$b = b - \alpha\frac{\partial J}{\partial b} = 0.1253 - 0.1 \cdot (-0.1311) = 0.1384.$$

Uvrštavanjem dobivenih težina w_1, w_2 i pristranosti b dobivamo vrijednosti:

$$\text{za} \quad x_1 = 1,\ x_2 = 4 \quad \text{vrijedi} \quad z_1 = 1 \cdot 0.6064 + 4 \cdot 0.5792 + 0.1384 = 3.0616,$$
$$\text{za} \quad x_1 = 2,\ x_2 = 5 \quad \text{vrijedi} \quad z_2 = 2 \cdot 0.6064 + 5 \cdot 0.5792 + 0.1384 = 4.4272,$$
$$\text{za} \quad x_1 = 3,\ x_2 = 6 \quad \text{vrijedi} \quad z_3 = 3 \cdot 0.6064 + 6 \cdot 0.5792 + 0.1384 = 5.4328.$$

Srednjekvadratna funkcija gubitka iznosi:

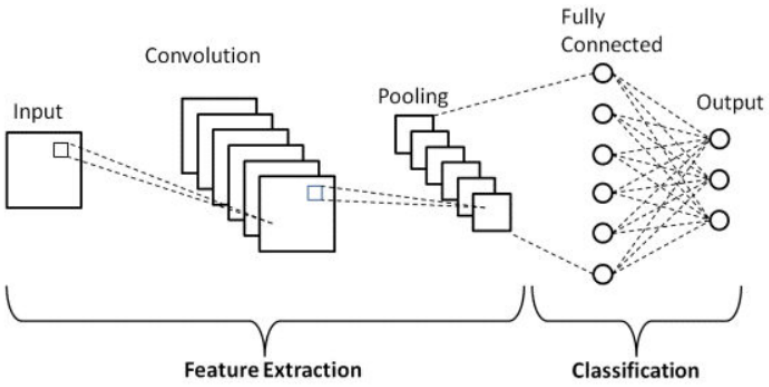
$$J(\theta) = \frac{2}{3}\sum_{i=1}^{3}(y_{true} - y_{pred})^2 = \frac{2}{3}\left[(5 - 3.0616)^2 + (7 - 4.2472)^2 + (9 - 5.4328)^2\right] = 15.4.$$

Nakon druge iteracije, srednjekvadratna greška se smanjila, što znači da su predviđene izlazne vrijednosti bliže stvarnim izlaznim vrijednostima. Manja vrijednost funkcije gubitka ukazuje na to da je neuronska mreža nakon druge korekcije težina bolja u predviđanju izlaznih vrijednosti te su dobivene težine bolje nego na početku. Postupak možemo ponavljati dok god srednjekvadratna pogreška nije prihvatljiva. Ukoliko nakon određene iteracije nove vrijednosti težina i koeficijenta pristranosti ne utječu na smanjenje greške, već je greška veća ili cijeli proces dovodi do divergencije, potrebno je provjeriti je li odabrana adekvatna funkcija aktivacije i ukoliko nije zamijeniti je s nekom drugom [38].

4.3.2 Konvolucijske neuronske mreže

Konvolucijske neuronske mreže (eng. *Convolutional Neural Network, CNN*) su vrsta neuronskih mreža koja se koristi u računalnom vidu, odnosno za obradu, klasifikaciju i tumačenje slika ili drugih tipova slikovnih podataka. Neuronska mreža obično se sastoji od 3 vrste slojeva - ulaznog sloja, skrivenog ili skrivenih slojeva te izlaznog sloja, dok se konvolucijska neuronska mreža sastoji od više različitih vrsta slojeva kao što su ulazni sloj, konvolucijski slojevi, slojevi za poduzorkovanje ili skupni slojevi (eng. *pooling layer*) i potpuno povezani slojevi. Konvolucijske neuronske mreže se najčešće koriste za klasifikacijske probleme. Kod prepoznavanja lica to uključuje probleme određivanja nalazi li se

na slici ljudsko lice ili ne te klasifikaciju koja osoba se nalazi na slici s obzirom na danu bazu podataka. Ulazna slika prolazi kroz niz konvolucijskih slojeva i slojeva za poduzorkovanje kako bi se raspoznale i naučile bitne značajke na slici. Konvolucijski slojevi koriste filtere (koji se ponekad nazivaju i jezgrama) za raspoznavanje različitih karakteristika na slici lica kao što su oblici, teksture i rubovi. Slojevi za poduzorkovanje smanjuju dimenzionalnost značajki zadržavajući samo najvažnije informacije. Te informacije se zatim prosljeđuju potpuno povezanim slojevima konvolucijske neuronske mreže (koji obavljaju zadaću kao standardni slojevi u svakoj neuronskoj mreži) u obliku jednodimenzionalnog vektora. Potpuno povezani slojevi zatim rješavaju problem klasifikacije i predviđaju izlaz ulaznih podataka. Na Slici 11, [19] dana je shema prikaza konvolucijske neuronske mreže [42, 28].



Slika 11: Shema prikaza arhitekture konvolucijske neuronske mreže i svih njezinih slojeva

Konvolucijske neuronske mreže su dobile ime po konvolucijskim slojevima koji su temeljni slojevi za izdvajanje i učenje značajki, odnosno konvolucijskoj operaciji[16] koja se u njima odvija. Konvolucija se provodi korištenjem filtera. Neka je svaka slika I predstavljena kao trodimenzionalni niz

$$I = m_1 \times m_2 \times m_c,$$

gdje je m_1 visina slike, m_2 širina slike i m_c dubina (ukoliko se koriste slike u boji onda je $m_c = 3$, ako su crno bijele slike, onda je $m_c = 1$). Zatim se na svaku sliku I primjenjuje filter

$$K = n_1 \times n_2 \times n_c,$$

gdje su n_1, n_2 visina i širina filtera, respektivno, a $n_c = m_c$. Filteri su uglavnom puno nižih dimenzija nego dimenzije ulazne slike, što znači da neuron u konvolucijskom sloju vidi samo mali dio ulaznih podataka. Time konvolucijski slojevi uče značajke koje su prostorno lokalizirane. Filter se pomiče preko piksela slike slijeva na desno, od gore prema dolje s obzirom na zadani broj pomaka i provodi se operacija konvolucije. Operacija konvolucije uključuje množenje piksela na mjestu preklapanja slike I i filtera K te zbrajanje dobivenih

[16] matematička funkcija nastala integriranjem umnoška dviju funkcija po intervalu njihove definicije gdje su te fukcije ravnopravne tako da svaka infinitezimalna promjena jedne funkcije utječe na drugu funkciju u cijelome intervalu definicije [23]

produkata. Dobiveni rezultat množenja, odnosno konvolucijske operacije je izlazna mapa značajki F dimenzije

$$dim(F) = \left(\frac{m_1 - n_1}{s} + 1\right) \times \left(\frac{m_2 - n_2}{s} + 1\right) \times 1,$$

gdje s određuje broj piksela za koje se filter pomiče. Vrijedi

$$F_{ij} = \sum_{x}^{m_1} \sum_{y}^{m_2} \sum_{z}^{m_c} K[x, y, z] I[i + x - 1, j + y - 1, z],$$

pri čemu su i i j indeksi retka odnosno stupca izlazne mape značajki, a x, y, z iteriraju kroz dimenzije filtera K. Postupak se ponavlja primjenom različitih vrsta filtera koji određuju različite značajke slike poput zamućenja, oštrine i slično. Na primjer, ako se izvodi konvolucija nad slikom dimenzija $6 \times 6 \times 1$ s filterom $3 \times 3 \times 1$ i definiranim pomakom za 1 piksel, dobiva se mapa značajki dimenzija $4 \times 4 \times 1$. Time se slika smanjuje prilikom provedbe svake konvolucije pa se operacija može provesti samo ograničeni broj puta. Također, s obzirom da se filter kreće od gore prema dolje, slijeva na desno pikseli u središtu slike imaju veći utjecaj u dobivenoj mapi značajki. Kako bi se spriječili ti problemi, slika se obloži dodatnim rubom na način da se taj rub popuni nulama (dodaju se nul stupci i retci na rub matrice koja prikazuje ulaznu sliku I). Nakon provođenja konvolucijske operacije, primjenjuje se aktivacijska funkcija na dobivenu mapu značajki kojom se uvodi nelinearnost u model. Najčešće se koristi ReLU aktivacijska funkcija,

$$c = F + b \quad \Rightarrow \quad \text{ReLU}(c).$$

Radi lakšeg razumijevanja, pretpostavimo da je slika prikazana matricom piksela I i da je zadan filter K te neka je definirani pomak $s = 1$.

$$I = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 0 & 1 & 2 \\ 3 & 4 & 0 & 1 \\ 2 & 3 & 4 & 0 \end{bmatrix} \qquad K = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}.$$

Filterom K prolazimo po slici I te množimo pripadne elemente filtera s pripadnim elementima dijela slike s kojim se filter poklapa te dobivene rezultate zbrojimo. Dobivena vrijednost je izlazna vrijednost u mapi značajki. Pomak filtera je za 1 piksel jer je defini-

rano $s = 1$. Preciznije, imamo:

$$I_1 = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} \quad \Rightarrow \quad I_1 \cdot K = 0 \cdot 1 + 1 \cdot 0 + 4 \cdot (-1) + 0 \cdot 1 = -4,$$

$$I_2 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \quad \Rightarrow \quad I_2 \cdot K = 1 \cdot 1 + 2 \cdot 0 + 0 \cdot (-1) + 1 \cdot (-1) = 1 - 1 = 0,$$

$$I_3 = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \quad \Rightarrow \quad I_3 \cdot K = 2 \cdot 1 + 3 \cdot 0 + 1 \cdot (-1) + 2 \cdot 1 = 2 - 1 + 2 = 3,$$

$$I_4 = \begin{bmatrix} 4 & 0 \\ 3 & 4 \end{bmatrix} \quad \Rightarrow \quad I_4 \cdot K = 4 \cdot 1 + 0 \cdot 0 + 3 \cdot (-1) + 4 \cdot 1 = 4 - 3 + 4 = 5,$$

$$I_5 = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} \quad \Rightarrow \quad I_5 \cdot K = 0 \cdot 1 + 1 \cdot 0 + 4 \cdot (-1) + 0 \cdot 1 = -4,$$

$$I_6 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \quad \Rightarrow \quad I_6 \cdot K = 1 \cdot 1 + 2 \cdot 0 + 0 \cdot (-1) + 1 \cdot 1 = 1 + 1 = 2,$$

$$I_7 = \begin{bmatrix} 3 & 4 \\ 2 & 3 \end{bmatrix} \quad \Rightarrow \quad I_7 \cdot K = 3 \cdot 1 + 4 \cdot 0 + 2 \cdot (-1) + 3 \cdot 1 = 3 - 2 + 3 = 4,$$

$$I_8 = \begin{bmatrix} 4 & 0 \\ 3 & 4 \end{bmatrix} \quad \Rightarrow \quad I_8 \cdot K = 4 \cdot 1 + 0 \cdot 0 + 3 \cdot (-1) + 4 \cdot 1 = 4 - 3 + 4 = 5,$$

$$I_9 = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} \quad \Rightarrow \quad I_9 \cdot K = 0 \cdot 1 + 1 \cdot 0 + 4 \cdot (-1) + 0 \cdot 1 = -4.$$

Dobivena mapa značajki nakon provedene konvolucijske operacije za ulaznu sliku I i filter K je

$$F = \begin{bmatrix} -4 & 0 & 3 \\ 5 & -4 & 2 \\ 4 & 5 & -4 \end{bmatrix}.$$

Ako želimo spriječiti da pikseli u središtu slike imaju veći utjecaj na dobivenu mapu značajku umjesto matrice I koja reprezentira sliku, promatramo matricu piksela I' koja izgleda

$$I' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 0 & 1 & 2 & 0 \\ 0 & 3 & 4 & 0 & 1 & 0 \\ 0 & 2 & 3 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

i provedemo analogan postupak kako bismo dobili mapu značajki.

Nakon prolaska kroz konvolucijske slojeve, podaci prolaze kroz slojeve za poduzorkovanje ili skupne slojeve čiji je cilj redukcija dimenzionalnosti mape značajki, ubrzanje

izračuna i smanjenje potrebe za memorijom. Oni sažimaju informacije s mape značajki tako da zadrže bitne značajke, a smanjuju manje važne informacije. Ovi slojevi su poprilično jednostavni - dobivena mapa značajki se podijeli na više dijelova, a zatim se primjeni neka operacija na svaki dio koja generira rezultat. Operacija može biti uzimanje najvećeg elementa (najčešće se koristi jer najbolje naglašava najbitnije značajke), zbrajanje elemenata ili uzimanje prosjeka. Pretpostavimo li da je ulazna mapa značajki dimenzija $p_1 \times p_2 \times d$. Dimenzije mape značajki nakon provedbe operacije su

$$dim(P) = \left(\frac{m_1 + 2p - n_1}{s}\right) \times \left(\frac{m_2 + 2p - n_2}{s}\right) \times m_c,$$

gdje je p broj dodanih redaka, odnosno stupaca kod dodavanja rubova ulazne slike. Nakon odabira značajki, mapa značajki se pretvara u jednodimenzionalni vektor. Opisani postupak možemo primijeniti na dobivenu mapu značajki F. Pretpostavimo da mapu značajki F podijelimo na regije veličine 2×2 te za svaku regiju uzimamo najveći element (eng. *max pooling*). Dobivamo sljedeće podregije i nad njima provodimo operaciju poduzorkovanja:

$$\begin{aligned}
F_1 = \begin{bmatrix} -4 & 0 \\ 5 & -4 \end{bmatrix} \quad &\Rightarrow \quad \max(F_1) = \max(\{-4, 0, 5, -4\}) = 5, \\
F_= \begin{bmatrix} 0 & 3 \\ -4 & 2 \end{bmatrix} \quad &\Rightarrow \quad \max(F_2) = \max(\{0, 3, -4, 2\}) = 3, \\
F_3 = \begin{bmatrix} 5 & -4 \\ 4 & 5 \end{bmatrix} \quad &\Rightarrow \quad \max(F_3) = \max(\{5, -4, 4\}) = 5, \\
F_4 = \begin{bmatrix} -4 & 2 \\ 5 & -4 \end{bmatrix} \quad &\Rightarrow \quad \max(F_4) = \max(\{-4, 2, 5\}) = 5.
\end{aligned}$$

Dobivena mapa značajki nakon prolaska kroz sloj za poduzorkovanje je

$$F = \begin{bmatrix} 5 & 3 \\ 5 & 5 \end{bmatrix}.$$

Prolazak kroz konvolucijske i slojeve za poduzorkovanje omogućuje izdvajanje najvažnijih značajki ulazne slike uz smanjenje prostorne dimenzije što omogućuje učinkovito učenje u potpuno povezanim slojevima konvolucijske neuronske mreže. Nakon što podaci prođu kroz konvolucijske i skupne slojeve, izlazna mapa značajki transformira se u jednodimenzionalni vektor. Ovaj vektor je ulazni podatak potpuno povezanih slojeva. Potpuno povezani slojevi slični su onima u običnim neuronskim mrežama. Svi neuroni jednog sloja povezani su sa svim neuronima idućeg sloja. Ulazni vektor prolazi kroz nekoliko slojeva s različitim težinama i pristranostima kako bi se naučili složeni nelinearni odnosi između značajki i izlaznih klasa. Aktivacijska funkcija korištena u potpuno povezanim slojevima je najčešće ReLU, osim kod zadnjeg potpuno povezanog sloja. Posljednji, potpuno povezani sloj kao aktivacijsku funkciju koristi logističku ili softmax funkciju koje vrše klasifikaciju izračunavanjem vjerojatnosti za svaku klasu, a broj neurona posljednjeg sloja odgovara početnom broju klasa.

Konvolucijsku mrežu možemo opisati sljedećim koracima:

1. ulazna slika prolazi kroz niz konvolucijskih slojeva koji vrše ekstrakciju bitnih značajki pomoću filtera, a kao izlazni podatak daju mapu značajki,

2. slojevi za poduzorkovanje smanjuju dimenzionalnost mape značajki i time ubrzavaju model i smanjuju potrebe za memorijom,

3. dobivena mapa značajki smanjene dimenzije se pretvara u jednodimenzionalni vektor koji postaje ulazni podatak potpuno povezanih slojeva koji provode klasifikaciju te korigiraju težine na analogan način kao slojevi obične neuronske mreže,

4. izlazni podatak je klasa za ulaznu sliku [42].

Opisane konvolucijske neuronske mreže mogu se promatrati kao algoritmi za prepoznavanje $2D$ modela lica i $3D$ modela lica. Razlika između navedenih algoritama konvolucijskih mreža je u dimenziji korištenih filtera unutar konvolucijskih slojeva i ulaznih podataka, to jest slika koje se obrađuju. Kod $2D$ konvolucijskih mreža, ulazni podatak je obično slika predstavljena matricom piksela, dok je kod $3D$ konvolucijskih mreža ulazni podatak niz $2D$ slika tijekom vremena, a korišteni filteri imaju tri dimenzije (visinu, širinu i dubinu). To omogućuje mreži da zahvati prostorne i vremenske uzorke u podacima. Stoga, ključna razlika između $2D$ i $3D$ konvolucijskih mreža je u dimenziji korištenih filtera unutar konvolucijskih slojeva. Kod $2D$ CNN-a, filteri su $2D$ matrice koje se kreću preko $2D$ ulazne slike. Kod $3D$ CNN-a, filteri su $3D$ tenzori koji se kreću kroz $3D$ ulazne podatke, što dovodi do mape značajki koja odražava te tri dimenzije [26].

Konvolucijske neuronske mreže često se koriste za prepoznavanje lica jer se putem filtera konvolucijskih slojeva mogu naučiti izdvojiti lice sa slike i prepoznati njegove bitne značajke. Treniranjem konvolucijske neuronske mreže može se postići da se algoritam bolje adaptira na varijacije u pozi, izrazu lica i osvjetljenju te su kao takvi prilagodljivi velikim skupovima podataka nad kojima je onda obrada brza i može se efikasno odvijati čak i u realnom vremenu (sustavi nadzora). Također, bolje su prilagođeni na šumove u ulaznim podacima nego većina drugih algoritama. Glavni nedostatak konvolucijskih neuronskih mreža je što zahtijevaju veliku količinu označenih podataka za treniranje kako bi se postigla zadovoljavajuća razina točnih predikcija. Trening skupovi za prepoznavanje lica trebaju biti raznoliki i sadržavati veliki broj slika lica osoba različitih spolova, godina, rasa, etničke pripadnosti te sva lica trebaju biti slikana u različitim uvjetima osvjetljenja, s različitim pozama i izrazima lica. Izrada takve baze podataka može biti duga, skupa, a ponekad i teško ostvariva zbog brige o zaštiti privatnosti. Izrada baze podataka još je zahtjevnija kod $3D$ konvolucijskih neuronskih mreža koje zahtijevaju i treću dimenziju ulaznih podataka. Također, treniranje algoritama konvolucijskih neuronskih mreža je računalno zahtjevno, dugotrajno i skupo.

5 Programska realizacija nekih algoritama

`Python` je programski jezik opće namjene objavljen 1991. godine poznat po svojoj jednostavnosti i čitljivosti. Karakterizira ga veliki skup ugrađenih modula i biblioteka koje omogućuju razvoj softvera i aplikacija. Popularan je jezik za rad na znanstvenim istraživanjima, modelima strojnog učenja te je jezik koji podržava grafički prikaz podataka.

Neka je zadan Olivetti skup lica [31] koji je prikupljen između travnja 1992. i travnja 1994. godine. U skupu se nalazi po 10 slika od 40 osoba, što ukupno čini bazu podataka od 400 slika lica. Slike su slikane u različito vrijeme, pod različitim osvjetljenjem i različitim izrazima lica. Sve slike su crno-bijele s crnom pozadinom i veličina slika je 64×64. Vrijednosti piksela fotografija su skalirane na interval $[0, 1]$, a osobe su označene brojevima od 0 do 39. Odredimo prosječno lice i vlastita lica pomoću Pythona. Kako bismo odredili prosječno lice i vlastita lica koristimo analizu glavnih komponenti da bismo smanjili dimenzionalnost.

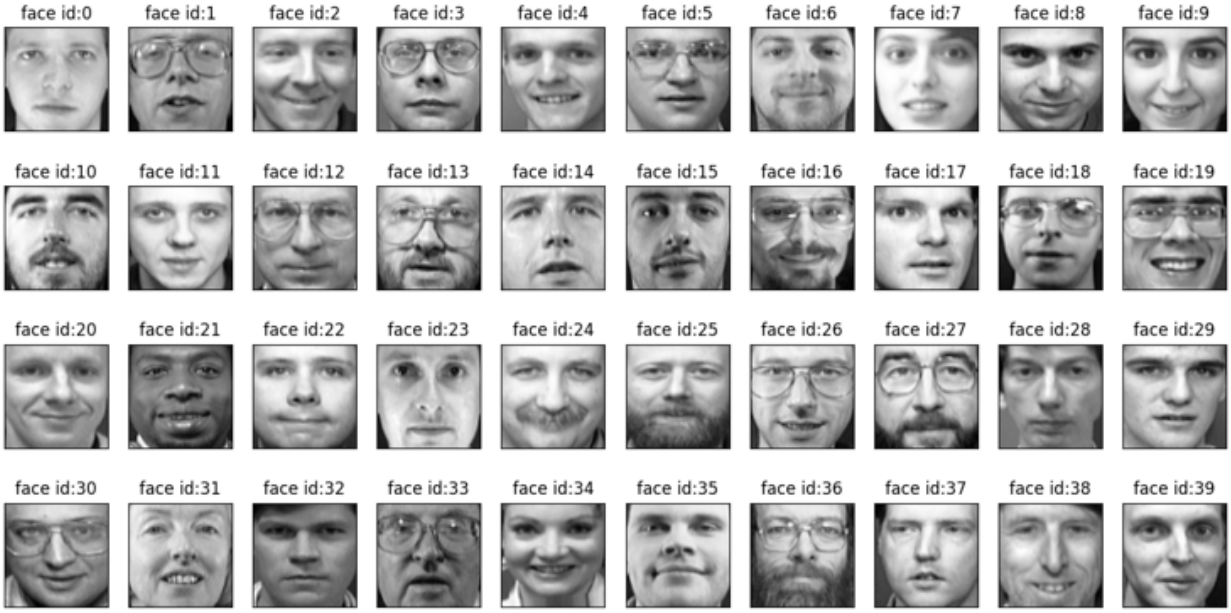
Na početku uvodimo osnovne biblioteke koje nam omogućuju rad s poljima i vizualni prikaz podataka. `Numpy` je standardna biblioteka unutar Pythona prilagođena radu s visokodimenzionalnim nizovima i matricama. Sadrži matematičke funkcije i podržava operacije linearne algebre. `Matplotlib` je biblioteka za vizualizaciju podataka i stvaranje animacija u Pythonu.

```
import numpy as np
import matplotlib.pyplot as plt

data = np.load('olivetti_faces.npy')
target = np.load('olivetti_faces_target.npy')
```

Grafički prikazujemo slike lica osoba pohranjenih u bazi te za jedno odabrano lice prikazujemo 10 različitih prikaza lica.

```
def Baza_podataka(slike, jedinstveni_ids):
    fig, ax = plt.subplots(nrows = 4, ncols = 10, figsize = (16, 8))
    ax = ax.flatten()
    for u_id in jedinstveni_ids:
        indeks_slike = u_id*10
        ax[u_id].imshow(slike[indeks_slike], cmap='gray')
        ax[u_id].set_xticks([])
        ax[u_id].set_yticks([])
        ax[u_id].set_title("face id:{}" .format(u_id))
    plt.savefig('olivetti_lica.png')

def Slike_jedne_osobe(slike, id_osobe):
    cols=10
    rows=(len(id_osobe)*10)/cols
    rows=int(rows)
    fig, ax = plt.subplots(nrows = rows, ncols = cols, figsize=(15,8))
```

```
    for i, s_id in enumerate(id_osobe):
        for j in range(cols):
            indeks_slike = s_id*10 + j
            ax[i,j].imshow(slike[indeks_slike], cmap="gray")
            ax[i,j].set_xticks([])
            ax[i,j].set_yticks([])
            ax[i,j].set_title("face id:{}".format(s_id))
    plt.savefig('lica10.png')

Baza_podataka(data, np.unique(target))
Slike_jedne_osobe(data, [0, 1])
```



Slika 12: Prikaz svih 40 različitih lica unutar dane baze podataka

Slika 13: Prikaz različitih 10 slika iste osobe pohranjenih u bazi

Prije provedbe PCA analize slika koje su reprezentirane kao polje matrica piksela, moramo ih pretvoriti u jednodimenzionalne vektore. To postižemo sljedećom naredbom

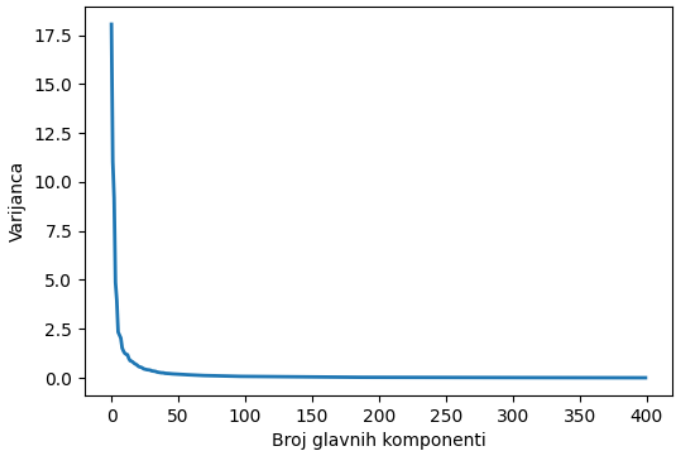
```
X = data.reshape((data.shape[0], data.shape[1]*data.shape[2])),
```

a zatim provodimo PCA analizu i pri tome koristimo biblioteku sklearn ili scikit-learn. `Sklearn` je biblioteka koja omogućuje rad s algoritmima strojnog učenja, uključujući funkcije za klasifikaciju i regresiju, redukciju dimenzionalnosti, pretprocesiranje podataka

(podjela na trening skup podataka i testni skup podataka) i evaluaciju modela. Kao trening skup podataka uzimamo 70% slika pohranjenih u bazi, a odabir koje su to slike je nasumičan, ali udio svake klase u ciljnoj varijabli je jednak.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, target, test_size = 0.3,
stratify = target, random_state = 0)
#stratify je naredba za očuvanja klase target varijable

from sklearn.decomposition import PCA
pca = PCA(n_components = 2)              # dimenzija je 2
pca.fit(X)
X_pca = pca.transform(X)
```

Grafički prikazujemo varijancu svake glavne komponente i na temelju dijagrama donosimo odluku koliko je glavnih komponenti potrebno uzeti.



Slika 14: Grafički prikaz varijance glavnih komponenata provedene PCA analize na danom skupu podataka

Zaključujemo da je nakon 90 glavnih komponenti varijacija između podataka jednaka 0, odnosno da oni predstavljaju iste podatke, što je i vidljivo na Slici 14. Stoga, za dani skup podataka promatramo 90 vlastitih lica. Grafički prikazujemo vlastita lica i računamo prosječno lice za dani skup podataka. Na kraju, ulazne podatke projiciramo u koordinatni sustav koji je razapet s glavnim komponentama provedene PCA analize. Time je smanjena dimenzionalnost ulaznog skupa podataka.

```
gl_komponente = 90
pca = PCA(n_components = gl_komponente, whiten=True)
pca.fit(X_train)

fig,ax = plt.subplots(1,1,figsize=(6,6))
```

```
ax.imshow(pca.mean_.reshape((64,64)), cmap="gray")
ax.set_xticks([])
ax.set_yticks([])
plt.savefig('prosjecno_lice_svm.png')

br_svojstvenih_lica = len(pca.components_)
eigen_faces=pca.components_.reshape((br_svojstvenih_lica, data.shape[1],
data.shape[2]))

cols = 10
rows = int(br_svojstvenih_lica/cols)
fig, ax = plt.subplots(nrows = rows, ncols = cols, figsize=(15,15))
ax = ax.flatten()
for i in range(br_svojstvenih_lica):
    ax[i].imshow(eigen_faces[i], cmap = "gray")
    ax[i].set_xticks([])
    ax[i].set_yticks([])
    ax[i].set_title("eigen id:{}" .format(i))
plt.savefig('eigenfaces_primjer_svm.png')

# transformacija ulaznih podataka u novodobiveni koordinatni sustav čije su
# osi glavne komponente provedene PCA analize
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
```

Slika 15: Prikaz prosječnog lica za dani skup podataka

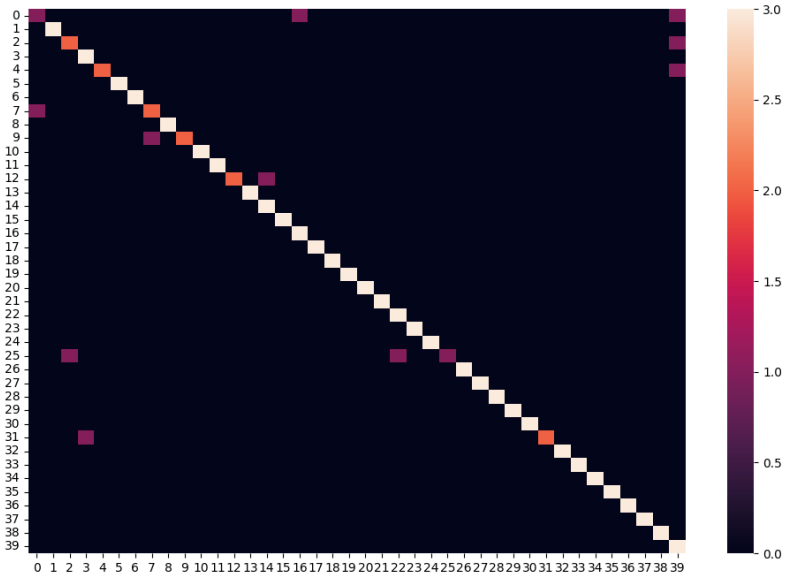
Slika 16: Prikaz vlastitih lica za dani skup podataka

U nastavku smo proveli SVM algoritam za prepoznavanje lica. Trening skup podataka činilo je 70% slika u ulaznom skupu podataka. Za provedbu SVM algoritma koristimo SVC klasifikator koji želi odrediti najbolje razdvajajuću hiperravninu između zadanih klasa, u našem slučaju 40 lica osoba u bazi podataka. Nakon provedenog algoritma računamo postotak točnih predikcija, uspoređujući predviđenu vrijednost algoritma na testnom skupu i pravu vrijednost.

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix

clf = SVC()
clf.fit(X_train_pca, y_train)
y_pred = clf.predict(X_test_pca)
print("accuracy score:{:.2f}".format(accuracy_score(y_test, y_pred)))
```

Dobiveni postotak točnih predikcija je 92%. Za provjeru točnosti procjene algoritama strojnog učenja mogu se koristiti različite metrike, a jedna od njih je matrica zabune. Matrica zabune daje vizualni prikaz točnih i netočnih predikcija. Na dijagonali se nalaze sve točne predikcije, a izvan dijagonale se nalaze netočne predikcije modela.

```
import seaborn as sns
plt.figure(1, figsize=(12,8))
sns.heatmap(confusion_matrix(y_test, y_pred))
plt.savefig('heatmap_svm.png')
```



Slika 17: Grafički prikaz matrice zabune za SVM algoritam na Olivetti skupu

Nakon provedbe SVM algoritma provodimo algoritam $2D$ konvolucijskih neuronskih mreža. Unutar Pythona, često se za modeliranje neuronskih mreža koristi `TensorFlow`. `TensorFlow` je platforma otvorenog koda koju je razvio Google Brain tim za provođenje istraživanja algoritama strojnog učenja i dubokih neuronskih mreža. Koristi se za razvoj različitih modela poput modela obrade prirodnog jezika, prepoznavanje slika, prepoznavanje rukopisa i za provođenje različitih simulacija kao što su parcijalne diferencijalne jednadžbe [43]. Uz to koristimo standardne biblioteke za rad s algoritmima strojnog učenja.

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

data = np.load('olivetti_faces.npy')
target = np.load('olivetti_faces_target.npy')
```

Ulazni skup podataka predimenziramo u slike oblika $64 \times 64 \times 1$ jer je veličina slika u bazi 64×64, a slike su crno-bijele pa je treća dimenzija $m_c = 1$.

```
X = data.reshape(400, 64, 64, 1)
```

Ciljanu varijablu pretvorimo u kategoričku varijablu jer se radi o problemu klasifikacije.

```
from tensorflow.keras.utils import to_categorical
y = to_categorical(target)
```

Trening skup podataka čini 80% ulaznih podataka, a testni skup 20%.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 42)
```

Definiramo sljedeće parametre modela:

- koristimo Sequantial model neuronskih mreža koji je linearan model, odnosno svaki sloj ima jedan ulazni podatak i izlazni podatak. Izlazni podaci jednog sloja su ulazni podaci drugog,
- definiramo $2D$ konvolucijske slojeve jer je ulazni skup podataka baza $2D$ slika lica. Koristimo 32 filtera i svaki filter je zadužen za prepoznavanje različite značajke lica. Veličina filtera K je 3×3 što znači da ulaznu sliku dijelimo u regije veličine 3×3 i izvodi se konvolucijska operacija (množenje pripadnih piksela i zbrajanje dobivenih produkata). Definirana aktivacijska funkcija je ReLU. Ulazne slike su 64×64 i slike su crno-bijele pa je parametar koji se odnosi na boju jednak 1,
- slojevi za poduzorkovanje su definirani kao *MaxPooling* slojevi, korišteni filter je veličine 2×2, a promatrana funkcija je funkcija maksimuma,
- definiramo sloj koji je zadužen za pretvaranje $2D$ matrice u jednodimenzionalni vektor koji se prosljeđuje potpuno povezanim slojevima konvolucijske neuronske mreže,
- u potpuno povezanim slojevima definiramo da je broj neurona 128, aktivacijska funkcija ReLU (osim za zadnji sloj) i da se tijekom treniranja ne promatraju svi neuroni nego samo 50% njih (težina tih neurona se tada postavlja na 0, a to omogućuje da model nije previše prilagođen pojedinim neuronima). Izlazni sloj ima 40 neurona (jer je 40 različitih klasa) i funkcija aktivacije izlaznog sloja je softmax funkcija jer se radi o klasifikacijskom problemu s 40 klasa. Softmax funkcija je funkcija koja vektoru $x \in \mathbb{R}^k$ pridružuje distribuciju vjerojatnosti od mogućih K ishoda i koristi se kod problema multinomijalne logističke regresije,
- kao optimizacijski algoritam gradijentnog spusta koristimo Adam (eng. *Adaptive Moment Estimation*) jer je prilagođen problemima s mnogo podataka i parametara,
- definirana funkcija gubitka je kategorička krosentropija (eng. *categorical crossentropy*) koja se često koristi kao funkcija gubitka kod softmax regresije, odnosno kod problema klasifikacije s više klasa, pri čemu se klase međusobno isključuju. Funkcija je definirana:

$$\mathrm{CE} = L(y, h(x)) = \sum_i y_{true_i} \cdot \log y_{pred_i},$$

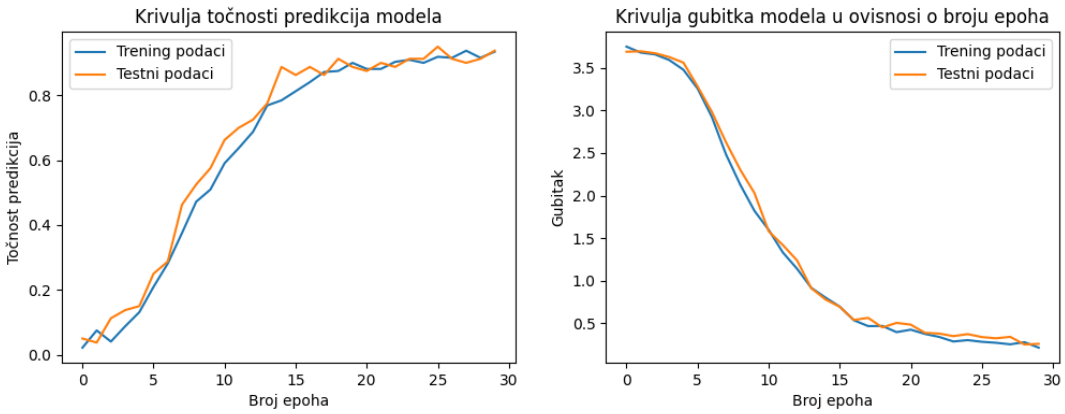
- broj prolaza kroz konvolucijsku neuronsku mrežu je 30, a broj podataka koji se obrađuju istovremeno je 32.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

model = Sequential([
    Conv2D(32, (3, 3), activation = 'relu', input_shape = (64, 64, 1)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation = 'relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation = 'relu'),
    Dropout(0.5),
    Dense(40, activation = 'softmax')])
model.summary()

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy',
metrics=['accuracy'])
cnn = model.fit(X_train, y_train, epochs=30, batch_size=32,
validation_data=(X_test, y_test))
test_loss, accuracy = model.evaluate(X_test, y_test, verbose=2)

from sklearn.metrics import accuracy_score
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)
y_test_classes = np.argmax(y_test, axis = 1)
print(accuracy_score(y_pred_classes, y_test_classes))
```

Dobivena točnost predikcije je 93.75%. Na slici u nastavku su prikazane krivulje točnosti predikcije i gubitka za provedeni model. Ako točnost predikcije raste kroz vrijeme na trening skupu podataka, onda model uči iz podataka za treniranje, a ako točnost predikcije raste kroz vrijeme na testnom skupu, onda je model dobar na novim, nepoznatim podacima. Krivulja gubitka pokazuje grešku modela na podacima za treniranje i testnim podacima kroz vrijeme. Ako se gubitak smanjuje kroz vrijeme, to znači da model uči i postaje bolji u predviđanju podataka za treniranje, a ako gubitak opada kod podataka za treniranje, znači da model postaje precizniji u predviđanjima kod nepoznatih podataka. Ako kroz vrijeme krivulje točnih predikcija na trening podacima i testnim podacima rastu i konvergiraju jedna ka drugoj, onda model dobro uči i predviđa. Analogno vrijedi za krivulje gubitka, no one opadaju i konvergiraju. Ostala kretanja krivulja mogu ukazivati

na podnaučenost[17] i prenaučenost[18].

```
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(cnn.history['accuracy'])
plt.plot(cnn.history['val_accuracy'])
plt.title('Krivulja točnosti predikcija modela')
plt.ylabel('Točnost predikcija')
plt.xlabel('Broj epoha')
plt.legend(['Trening podaci', 'Testni podaci'], loc='best')

plt.subplot(1, 2, 2)
plt.plot(cnn.history['loss'])
plt.plot(cnn.history['val_loss'])
plt.title('Krivulja gubitka modela u ovisnosi o broju epoha')
plt.ylabel('Gubitak')
plt.xlabel('Broj epoha')
plt.legend(['Trening podaci', 'Testni podaci'], loc='best')
plt.savefig('cnn_krivulje')
plt.show()
```



Slika 18: Grafički prikaz krivulje točnosti predikcije i krivulje gubitka na trening skupu podataka i testnom skupu u ovisnosti o broju epoha

Ako usporedimo točnost predikcije SVM algoritma i $2D$ konvolucijskih mreža zaključujemo da je algoritam $2D$ konvolucijskih neuronskih mreža precizniji. Često se uz točnost predikcija algoritama uspoređuje i vrijeme potrebno za obradu podataka i klasifikaciju. Dok

[17] model strojnog učenja nije dobro prepoznao i naučio uzorke u trening podacima i nije sposoban generalizirati naučene uzorke na novim podacima. Takav model ima veliku stopu netočnih predikcija na trening skupu podataka i nije pouzdan za primjenu na novim, nepoznatim podacima. Posljedica je pristranosti i nedovoljnog izvora varijacije.

[18] model jako dobro opisuje trening podatke ili im u potpunosti odgovara, no takav model se ne može generalizirati na novim, nepoznatim podacima. Najčešće je posljedica trening skupa s malim brojem podataka ili pretjeranog prilagođavanja parametara modela trening podacima.

SVM algoritam ne troši "ništa" vremena (potrebno vrijeme za klasifikaciju je manje od jedne sekunde), algoritmu $2D$ konvolucijskih neuronskih mreža potrebno je vrijeme od jedne minute da izvrši klasifikaciju. Stoga, vremenski je optimalnije koristit SVM algoritam. Također, odabir SVM-a može biti generalno bolji odabir jer razlika u točnosti predikcije nije velika (manja od 2%), a postoji značajnija razlika u vremenu izvršavanja. Treba uzeti u obzir da je promatrana baza slika malena (sastoji se od ukupno 400 slika) i već na tako malom skupu podataka vidljivo je da je algoritam konvolucijskih neuronskih mreža precizniji, ali i sporiji od drugih algoritama.

6 Primjena algoritama za prepoznavanje lica

Tehnologija prepoznavanja lica vrlo je primjenjiva i koristi se za identifikaciju i verifikaciju osobe na osnovu njenog lica. Algoritmi za prepoznavanje lica primjenjuju se u različite svrhe. Jedna od najčešćih primjena prepoznavanja lica je kao biometrijska metoda autentifikacije. Mnoge tvrtke, banke i ostali koriste prepoznavanje lica kao jednu od metoda provjere identiteta zaposlenika za pristup sigurnosnim podacima s ograničenim pristupom. Ovisno o razini sigurnosti koja se želi postići, potrebno je pronaći balans između stope lažnog prihvaćanja i zadovoljstva korisnika korištenjem tehnologije. Najčešće niska stopa lažnog prihvaćanja uzrokuje i krive klasifikacije ovlaštenog pristupa, odnosno ne prepoznaje lice osobe koja ima pristup sustavu kao ovlašteno lice, čime osoba mora više puta pristupiti sustavu. Sve češće se prepoznavanje lica koristi i u komercijalne svrhe. Većina pametnih mobilnih telefona svojim korisnicima omogućuje otključavanje uređaja skeniranjem lica. Predvodnik te tehnologije bio je Apple sa mogućnošću FaceID identifikacije. FaceID koristi TrueDepth kameru koja uključuje infracrvenu kameru, osvjetljivač i projektor točaka. Osvjetljivač osvjetljava lice korisnika nevidljivim infracrvenim svjetlom, što omogućuje skeniranje lica i u uvjetima lošeg osvjetljenja. Projektor točaka projicira oko 30000 nevidljivih infracrvenih točaka na lice korisnika koje stvaraju jedinstvenu dubinsku mapu mjereći izobličenja točaka kako se spuštaju na konture lica. Infracrvena kamera snima uzorke reflektiranog svjetla i položaj svake od točaka na licu korisnika. Sustav zatim koristi te informacije za stvaranje detaljne $3D$ geometrijske mape lica, hvatajući dubinu i konture crta lica kao što su oči, nos i usta. Informacije se zatim obrađuju korištenjem neuronskih mreža. Kada korisnik pokuša otključati svoj uređaj, TrueDepth kamera ponovno stvara dubinsku mapu i snima infracrvenu sliku korisnikovog lica. Novi podaci o licu obrađuju se pomoću neuronske mreže, a generira se novi model lica. Ovaj model se zatim uspoređuje s pohranjenim modelom. Ako se modeli dovoljno podudaraju (na temelju unaprijed postavljenog praga), uređaj se otključava. FaceID koristi adaptivno učenje kako bi s vremenom poboljšao svoju preciznost. Kontinuirano ažurira pohranjeni model lica novim podacima kada dođe do malih promjena u izgledu korisnika, kao što su rast brade, nošenje naočala, promjena frizure ili starenje, što osigurava da FaceID ostane točan i pouzdan čak i kada se korisnikov izgled postupno mijenja. Vjerojatnost da nasumična osoba može otključati Iphone uređaj pogledom je manja od 1 naspram 1000000 (uz iznimku korištenja

FaceID tehnologije s maskom ili ako se radi o jednojajčanim blizancima koji izgledaju jako slično), a kao dodatna zaštita je postavljeno ograničenje na 5 neuspješnih pokušaja podudaranja lica prije zahtjeva za unosom lozinke [2]. Kako se postavljena biometrija uređaja može povezati s korištenjem ostalih aplikacija, većina aplikacija mobilnog bankarstva koristi prepoznavanje lica kao jednu od biometrijskih metoda za autentifikaciju. To omogućuje korisnicima prijavu na svoje račune i potvrdu transakcija bez potrebe za unosom lozinke. Također, prepoznavanje lica može se koristiti prilikom online kupovine ili kod upotrebe ostalih aplikacija. Prepoznavanje lica koristi se u nadzornim sustavima kako bi se identificirali pojedinci na javnim mjestima poput aerodroma, željezničkih i autobusnih stanica, stadiona, trgovačkih centara i slično. Sustavi mogu automatski prepoznati i pratiti poznate kriminalce ili nestale osobe što pomaže u provedbi zakona. Nadzorni sustavni najčešće koriste modele temeljene na dubokom učenju poput konvolucijskih neuronskih mreža koje otkrivaju lica s velikom preciznošću i brzinom, čak i onda kad su ulazne slike niske kvalitete. Nakon toga se mogu koristiti algoritmi za poravnanje lica i izdvajanje značajki te se uvijek koriste neke od spomenutih metoda za smanjenje dimenzionalnosti poput PCA analize. Morfabilni modeli se često koriste kod sustava koji trebaju pretvoriti $2D$ sliku lica u $3D$ prikaz lica. Jedna od najširih primjera algoritma za prepoznavanje lica je na društvenim mrežama za generiranje predložaka osoba na fotografijama. Algoritmi za prepoznavanje lica koriste se na nekim sveučilištima i sveučilišnim kampusima kako bi se povećala razina sigurnosti, ali isto tako pratila dolaznost studenata na predavanja. Pametni domovi koriste tehnologiju prepoznavanja lica kao dio sigurnosnog sustava, ali isto tako mogu podesiti ugođaj prostorije s obzirom na to koji se ukućanin nalazi u njoj i njegove preference o sobnoj temperaturi, svjetlini prostorije i slično. Algoritmi za prepoznavanje lica mogu pomoći u dijagnosticiranju nekih bolesti koristeći specifične značajke na nosu, obrazima i drugim dijelovima ljudskog lica. Oslanjajući se na razvijene skupove podataka, strojno učenje korišteno je za prepoznavanje genetskih abnormalnosti samo na temelju dimenzija lica. Jedna od novijih tehnologija koja se razvija je razvoj algoritama za detekciju emocija na ljudskom licu [13].

6.1 Prednosti i mane korištene tehnologije

Jedna od prednosti sustava za prepoznavanje lica je mogućnost masovne identifikacije budući da se tehnologija može koristiti i bez znanja pojedinca. Međutim, u usporedbi s drugim biometrijskim tehnikama, prepoznavanje lica možda nije najpouzdanije i najučinkovitije. Mjere kvalitete vrlo su važne u sustavima za prepoznavanje lica jer su moguće velike varijacije u slikama lica. Čimbenici kao što su osvjetljenje, izraz lica, položaj i šum tijekom snimanja lica mogu utjecati na performanse sustava za prepoznavanje lica. Među svim biometrijskim sustavima, prepoznavanje lica ima najveću stopu lažnog prihvaćanja i odbijanja, stoga su se postavila pitanja o učinkovitosti ili pristranosti algoritama za prepoznavanje lica. Prepoznavanje lica manje je učinkovito ako se izrazi lica razlikuju. Također, postoji nedosljednost u skupovima podataka koje koriste istraživači. Iako se za neke sustave prepoznavanja lica tvrdi visok stupanj točnosti, ti rezultati nisu univerzalni. Dosljedno se najlošija stopa točnosti pokazala za osobe između 18 do 30 godina, Afroamerikance ili osobe ženskog spola, dok su najbolja predviđanja kod bijelih

muškaraca u srednjim godinama. Organizacije za građanska prava i zaštitu privatnosti tvrde da je privatnost pojedinca ugrožena korištenjem tehnologija nadzora. Prepoznavanje lica može se koristiti ne samo za identifikaciju osobe, već i za otkrivanje drugih osobnih podataka povezanih s osobom. Sukladno tome, pojedine države su donijele zakone koje zabranjuju ili ograničavaju korištenje tehnologija prepoznavanja lica na javnim površinama. Europska unija je u siječnju 2020. godine predložila, ali zatim i brzo odbacila, predloženi moratorij na prepoznavanje lica u javnim prostorima [13, 27].

7 Zaključak

U diplomskom radu smo opisali povijesni razvoj i neke od najkorištenijih algoritama za prepoznavanje lica. Prepoznavanje lica je biometrijski alat koji na temelju lica može prepoznati osobu te izvršavati dvije osnovne funkcije: identifikaciju i autentifikaciju. Svaki algoritam kao ulazni podatak prima sliku osobe koju želi prepoznati i bazu podataka koja se sastoji od slika svih osoba s kojima ih uspoređuje, a zatim detektira lice na slici, analizira ga i izdvaja bitne značajke koje se uspoređuju s bazom lica u svrhu prepoznavanja. Osnovna podjela algoritama za prepoznavanje lica je na $2D$ i $3D$ algoritme. $2D$ algoritmi kao ulazni podatak primaju $2D$ sliku i neki od najpoznatijih su Eigenface algoritam, metoda potpornih vektora i konvolucijske neuronske mreže. Ograničenja $2D$ algoritama su promjene u fizičkom izgledu, promjena orijentacije glave ili uvjeti osvjetljenja na slici. Ti problemi se mogu premostiti korištenjem $3D$ algoritama za prepoznavanje lica koji obuhvaćaju trodimenzionalnu geometriju lica. Najpoznatiji $3D$ modeli su model $3D$ prepoznavanja bez rekonstrukcije lica, morfabilni model i $3D$ konvolucijske neuronske mreže. S obzirom na to da svi algoritmi kao ulazne podatke primaju slike koje se prikazuju kao matrice ili vektori piksela velikih dimenzija, jedan od problema algoritama je reducirati dimenziju prostora, a da se pri tome sačuvaju sve najbitnije značajke lica. Jedan od načina na koji se to postiže je korištenjem analize glavnih komponenata, PCA ili korištenjem filtera kod konvolucijskih neuronskih mreža. Uz matematičku pozadinu navedenih algoritama za prepoznavanje $2D$ i $3D$ modela lica, u radu je dana programska realizacija analize glavnih komponenata, SVM algoritma i $2D$ konvolucijskih neuronskih mreža na Olivetti skupu podatak. Uspoređena je točnost predikcija SVM algoritma i algoritma $2D$ konvolucijskih neuronskih mreža. Kao precizniji, ali ne i brži algoritam pokazao se $2D$ CNN, čime su istaknute njegove glavne prednosti, ali i mane u odnosu na druge algoritme. SVM algoritam je neznatno lošiji u predviđanju ishoda od $2D$ CNN algoritma, ali je znatno brži, stoga je optimalniji za korištenje na danoj bazi podataka.

Tehnologija prepoznavanja lica vrlo je primjenjiva i koriste je sigurnosni sustavi i sustavi za nadzor za zaštitu podataka i provedbu zakona, pametni uređaji za otključavanje i kao dio multifaktorske autentifikacije, a počinje se koristiti i u medicini za prepoznavanje genetskih bolesti. Prepoznavanje lica omogućuje visoku razinu sigurnosti i jednostavnosti za korisnike, ali zahtijeva balans između niske stope lažnog prihvaćanja i pozitivnog korisničkog iskustva. Međutim, unatoč brojnim prednostima, glavni problem s kojima se tehnologija susreće je zaštita privatnosti. Ostaje uvijek aktualno pitanje koliko je tehnologija precizna te poštuje li sva etička načela prilikom kreiranja velikih baza podataka.

Popis slika

Literatura

[1] *A Brief History of Face Recognition*, FACEFIRST, URL: https://www.facefirst.com/post/a-brief-history-of-face-recognition, (preuzeto: 1.4.2024.)

[2] *About Face ID advanced technology*, Apple, URL: https://support.apple.com/en-us/102381 (preuzeto: kolovoz 2024.)

[3] A. J. Goldstein, L. D. Harmon and A. B. Lesk, *Identification of human faces*, in Proceedings of the IEEE, vol. 59, no. 5, pp. 748-760, May 1971,

[4] Andrew Ng, *Lecture Notes*, Machine Learning, URL: https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf (preuzeto: ožujak 2024.)

[5] Blanz V., Vetter T., *A Morphable Model for the Synthesis of 3D Faces*, Proc. of the SIGGRAPH'99, Los Angeles, USA, 1999., 187-194 ,

[6] Bronstein A., Bronstein M., Kimmel R., Spira A., *3D face recognition without facial surface reconstruction*, in Proceedings of ECCV 2004, Prague, Czech Republic, 2004., 11-14,

[7] Camarillo E., *Illinois Google users to receive about* 95 *as part of privacy lawsuit settlement*, URL: https://www.wglt.org/illinois/2023-06-06/illinois-google-users-to-receive-about-95-as-part-of-privacy-lawsuit-settlement (preuzeto: srpanj 2024.)

[8] Chellappa, R., Phillips, P. J., Rosenfeld, A., Zhao, W, *Face recognition: A literature survey*, ACM Computing Surveys (CSUR), str. 399-458

[9] Computer Vision and Intelligence Group, *3D Morphable Models*, URL: https://medium.com/@cvigroup.cfi/3d-morphable-models-b485fefc1761 (preuzeto: srpanj 2024.)

[10] *Convolutional Neural Network (CNN)*, TensorFlow, URL: https://www.tensorflow.org/tutorials/images/cnn (preuzeto: kolovoz 2024.)

[11] *Deep Face Recognition*, geeksforgeeks, URL: https://www.geeksforgeeks.org/deep-face-recognition/ (preuzeto: travanj 2024.)

[12] *Face Recognition Technology (FERET)*, NIST, URL: https://www.nist.gov/programs-projects/face-recognition-technology-feret, (preuzeto: 3.4.2024.)

[13] *Facial recognition system*, Wikipedia, URL: https://en.wikipedia.org/wiki/Facial_recognition_system, (preuzeto: 27.3.2024.)

[14] *Facial Recognition Technology*, Inovatrics, URL: https://www.innovatrics.com/facial-recognition-technology/, (preuzeto: travanj 2024.)

[15] *Gabor filter*, Wikipedia, URL: https://en.wikipedia.org/wiki/Gabor_filter (preuzeto: rujan 2024.)

[16] Gordon, G., *Face Recognition Based on Depth and Curvature Features*, 1992., IEEE Computer Society Conference on Computer Vision and Pattern Recognition

[17] Gordon G., *Face recognition from frontal and profile views*, Proc. Int'l Workshop on Face and Gesture Recognition, pp. 47-52, 1996.

[18] *Gradient Descent Algorithm in Machine Learning*, GeeksForGeeks URL: `https://www.geeksforgeeks.org/gradient-descent-algorithm-and-its-variants/` (preuzeto: kolovoz 2024.)

[19] Gurucharan M.K., *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network*, upGrad, URL: `https://www.upgrad.com/blog/basic-cnn-architecture/` (preuzeto: rujan 2024.)

[20] Horvatić K., *Linearna algebra*, Golden marketing - Tehnička knjiga, 2004. god.

[21] *Identification and Authentication: Similarities and Differences*, okta, URL: `https://www.okta.com/identity-101/identification-vs-authentication/` (preuzeto: kolovoz 2024.)

[22] *Kernel Trick in Support Vector Classification*, GeeksForGeeks, URL: `https://www.geeksforgeeks.org/kernel-trick-in-support-vector-classification/` (preuzeto: rujan 2024.)

[23] *konvolucija*, struna, Institut za hrvatski jezik i jezikoslovlje, URL: `http://struna.ihjj.hr/naziv/konvolucija/19228/` (preuzeto: rujan 2024.)

[24] Materijali s predavanja kolegija Strojno učenje 2024., doc. dr. sc. Sanda Bujačić Babić

[25] *Mathematical Approach to PCA*, geeksFforgeeks, URL: `https://www.geeksforgeeks.org/mathematical-approach-to-pca/`, (preuzeto: lipanj 2024.)

[26] Mersico, *Understanding 1D, 2D and 3D Convolution Network*, kaggle, URL: `https://www.kaggle.com/code/mersico/understanding-1d-2d-and-3d-convolution-network` (preuzeto: kolovoz 2024.)

[27] Mildebrath H., Madiega T., *Regulating facial recognition in the EU*, European Parliamentary Research Service, 2021.

[28] Mishra M., *Convolutional Neural Networks, Explained*, Towards Data Science, 2020., URL: `https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9` (preuzeto: kolovoz 2024.)

[29] *Multivariate normal distribution*, Wikipedia, URL: `https://en.wikipedia.org/wiki/Multivariate_normal_distribution` (preuzeto: rujan 2024.)

[30] Parke F. I., *A Parametric Model of Human Faces*, PhD thesis, University of Utah, Salt Lake City, 1974.

[31] Peldek S., *Face Recognition on Olivetti Dataset*, URL : https://www.kaggle.com/code/serkanpeldek/face-recognition-on-olivetti-dataset/notebook, (preuzeto: 6.6.2024.)

[32] Pentland A., Turk M., *Eigenfaces for Recognition*, Journal of Cognitive Neurosicence, Vol. 3, No. 1, 1991, 71-86.

[33] Philis P. J., *Support Vector Machines Applied to Face, National Institute of Standards and Technology, Recognition*, URL: https://proceedings.neurips.cc/paper_files/paper/1998/file/a2cc63e065705fe938a4dda49092966f-Paper.pdf, (preuzeto: 5.6.2024.)

[34] python, URL: https://www.python.org/about/ (preuzeto: kolovoz 2024.)

[35] Raviv S., *The Secret History of Facial Recognition*, spicework, URL: https://www.wired.com/story/secret-history-facial-recognition/, (preuzeto: ožujak 2024.)

[36] Reimann N., *Texas Suing Meta Over Facial Recognition Technology—Seeking Hundreds Of Billions In Penalties*, Forbes, URL: https://www.forbes.com/sites/nicholasreimann/2022/02/14/texas-suing-meta-over-facial-recognition-technology-seeking-hundreds-of-billions-in-penalties/?sh=513d83d2574c (preuzeto: srpanj 2024.)

[37] Sirovich L., Kirby M. (1987.), *Low-dimensional procedure fot the characterzation of human faces*, Journal of the Optical Society of America, Vol. 4, URL: https://www.researchgate.net/publication/19588504_Low-Dimensional_Procedure_for_the_Characterization_of_Human_Faces (16.4.2024.)

[38] Soudani A., *Building an Artificial Neural Network model by hand*, Medium, URL: https://medium.com/@soudanik/building-a-deep-learning-model-by-hand-bd51feccdfc7 (preuzeto: kolovoz 2024.)

[39] Sullivan E. *Facial Recognition Technology*, Economic Affairs Interim Committee, URL: https://leg.mt.gov/content/Committees/Interim/2021-2022/Economic%20Affairs/Studies/HJR-48/facial-recognition-technology.pdf (preuzeto: ožujak 2024.)

[40] *Support vector machine*, Wikipedia, URL: https://en.wikipedia.org/wiki/Support_vector_machine (preuzeto: lipanj 2024.)

[41] Šmuc. T., *Metoda potpornih vektora (SVM – Support Vector Machines)*

[42] Tariq F.,*Breaking Down the Mathematics Behind CNN Models: A Comprehensive Guide*, Medium, URL: https://medium.com/@beingfarina/breaking-down-the-mathematics-behind-cnn-models-a-comprehensive-guide-1853aa6b011e (preuzeto: kolovoz 2024.)

[43] TensorFlow, URL: https://www.tensorflow.org/ (preuzeto: rujan 2024.)

[44] *Top 11 Facial Recognition Software in 2021*, spicework, URL: https://www.spiceworks.com/it-security/identity-access-management/articles/facial-recognition-software/, (preuzeto: 25.3.2024.)

[45] *Using a Hard Margin vs Soft Margin in SVM*, GeeksForGeeks, URL: https://www.geeksforgeeks.org/using-a-hard-margin-vs-soft-margin-in-svm/ (preuzeto: lipanj 2024.)

[46] Vilas H. Gaidhane, Y. V. Hote, Singh V., *An efficient approach for face recognition based on common eigenvalues*, SemanticSholar, URL: https://www.semanticscholar.org/paper/An-efficient-approach-for-face-recognition-based-on-Gaidhane-Hote/bd21028c19e7b2a35dbdf8794d845b15fe7e3aac/figure/11, (preuzeto: ožujak 2024.)

[47] Wilimitis D., *The Kernel Trick in Support Vector Classification*, Medium, URL: https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f, (preuzeto: 15.7.2024.)

[48] Xiaoli J., *Understanding The Math Behind Dimension Reduction in Facial Recognition(3)*, Medium, URL : https://medium.com/swlh/understanding-the-math-behind-dimension-reduction-in-facial-recognition-3-5209af1f1596, (preuzeto: 2.6.2024.)