

# Primjena algoritama strojnog učenja u prepoznavanju lica

---

**Puljko, Benjamin**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:060378>

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

*Download date / Datum preuzimanja:* **2025-03-15**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**PRIMJENA ALGORITAMA STROJNOG UČENJA U  
PREPOZNAVANJU LICA**

**Diplomski rad**

**Benjamin Puljko**

**Osijek, 2020.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 14.07.2020.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za diplomski ispit**

<b>Ime i prezime studenta:</b>	Benjamin Puljko
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije
<b>Mat. br. studenta, godina upisa:</b>	D-15ARK, 04.10.2019.
<b>OIB studenta:</b>	90394759476
<b>Mentor:</b>	Izv. prof. dr. sc. Emmanuel Karlo Nyarko
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	Doc.dr.sc. Ratko Grbić
<b>Član Povjerenstva 1:</b>	Izv. prof. dr. sc. Emmanuel-Karlo Nyarko
<b>Član Povjerenstva 2:</b>	Izv. prof. dr. sc. . Damir Filko
<b>Naslov diplomskog rada:</b>	Primjena algoritama strojnog učenja u prepoznavanju lica
<b>Znanstvena grana rada:</b>	<b>Umjetna inteligencija (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	Treba istražiti različite algoritme strojnog učenja prikladne za prepoznavanje lica. Implementirati i usporediti nekoliko takvih algoritama za prepoznavanje lica.
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene mentora:</b>	14.07.2020.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 21.07.2020.

**Ime i prezime studenta:**

Benjamin Puljko

**Studij:**

Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije

**Mat. br. studenta, godina upisa:**

D-15ARK, 04.10.2019.

**Turnitin podudaranje [%]:**

5

Ovom izjavom izjavljujem da je rad pod nazivom: **Primjena algoritama strojnog učenja u prepoznavanju lica**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Emmanuel Karlo Nyarko

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

1. UVOD .....	1
2. ALGORITMI STROJNOG UČENJA .....	4
2.1. Analiza glavnih komponentata (PCA).....	4
2.2. Logistička regresija .....	7
2.3. Analiza linearnih diskriminanti (LDA) .....	8
2.4. Metoda potpornih vektora (SVM).....	10
2.5. K najbližih susjeda .....	12
2.6. Konvolucijske neuronske mreže .....	13
3. PRIPREMA SKUPA PODATAKA .....	15
3.1. Prikupljanje podataka .....	15
3.2. Filtriranje podataka.....	15
3.3. Detekcija i poravnanje lica .....	16
3.3.1. Viola-Jones algoritam.....	16
3.3.2. Poravnanje lica.....	19
3.4. Priprema lica za upotrebu u klasifikaciji .....	20
4. EKSPERIMENTALNA ISTRAŽIVANJA.....	23
4.1. Eksperiment 1 .....	24
4.1.1. K najbližih susjeda .....	24
4.1.2. Logistička regresija .....	26
4.1.3. Analiza linearnih diskriminanti.....	28
4.1.4. SVM klasifikacija.....	30
4.2. Eksperiment 2 .....	32
4.3. Eksperiment 3 .....	34
4.4. Eksperiment 4 .....	37
4.4.1. Vgg16 .....	37
4.4.2. Facenet.....	40
5. ZAKLJUČAK.....	43
LITERATURA.....	44
SAŽETAK .....	47
ABSTRACT .....	48
ŽIVOTOPIS.....	49
PRILOZI .....	50

## 1. UVOD

Ranijih godina, tehnologiju prepoznavanja lica većina ljudi doživljavala je kao znanstvenu fantastiku, no u posljednjem desetljeću tehnologija prepoznavanja lica nije samo postala stvarna, već i raširena. Danas ljudi mogu svugdje s lakoćom čitati članke i vijesti o prepoznavanju lica. Tehnologija prepoznavanja lica zajedno s umjetnom inteligencijom (engl. Artificial intelligence, AI) i dubokim učenjem imaju koristi u nekoliko industrija. Te industrije uključuju zakonodavne institucije, zračne luke, tvrtke za proizvodnju mobilnih telefona, tvrtke za proizvodnju kućanskih aparata i sl. Danas čak i trgovci upotrebljavaju prepoznavanje lica zasnovano na umjetnoj inteligenciji kako bi spriječili nasilje i kriminal. Zračne luke dobivaju bolje zaštićeno okruženje, proizvođači mobilnih uređaja rabe prepoznavanje lica kako bi u uređaje unijeli biometrijsku sigurnosnu značajku.

Svijet vjeruje da je Woodrow Wilson Bledsoe otac prepoznavanja lica [1]. Šezdesetih godina prošlog stoljeća Bledsoe je stvorio sustav koji je mogao ručno organizirati fotografije lica upotrebljavajući RAND tablet. Tablet je uređaj koji bi ljudi mogli rabiti za unošenje vertikalnih i horizontalnih koordinata na rešetku uz pomoć olovke koja ispušta elektromagnetske impulse. Ljudi su taj sustav upotrebljavali za ručno snimanje koordiniranih područja lica, poput očiju, nosa, usta i kose. Ručno snimljene metrike moguće je kasnije spremirati u bazu podataka, a kad je nova fotografija pojedinca unesena u sustav, bila je u mogućnosti dobiti najsličniju sliku preko baze podataka.

Sedamdesetih godina, Harmon, Goldstein i Lesk [2] unaprijedili su ručni sustav za prepoznavanje lica da bude precizniji. Primjenjivali su 21 marker za lice, uključujući debljinu usana i boju kose za automatsko otkrivanje lica. Međutim, Bledsoe-ev sustav se i dalje izračunava sa stvarnim biometrijskim podacima, rađenima ručno. Sirovich i Kirby počeli su rabiti linearnu algebru za problem prepoznavanja lica 1986. godine [3]. Pristup koji su upotrebljavali nazvali su „Eigenface“ pristup ili pristup svojstvenih lica. Prikazivanje je počelo kao potraga za manje dimenzionalnim prikazom lica. Njihov tim je dokazao da analiza značajki na prikupljenim slikama u bazi može tvoriti skup osnovnih značajki. Također su mogli objasniti kako se za precizno kodiranje slike lica može upotrijebiti manje od sto vrijednosti. 1991. godine Pentland i Turk nastavili su dalje raditi primjenom metode svojstvenih vrijednosti pronalazeći načine otkrivanja lica unutar slika [4]. Njihov rad bio je

prvi pokušaj automatskog prepoznavanja lica za koje su rabili tehnološke i okolišne čimbenike.

U razdoblju od 1993.-2000. godine DARPA i NIST objavili su program FERET kako bi potaknuli komercijalno tržište prepoznavanja lica. U 2002. godini, službenici za provođenje zakona primijenili su prepoznavanje lica u kritičnim testiranjima tehnologije. Početkom 2010. godine, facebook je počeo upotrebljavati svojstvo prepoznavanja lica koje je pomoglo u otkrivanju ljudi u fotografijama koje su ažurirali korisnici facebook-a. Dok je ažuriranje stvorilo ushićenje u medijskoj industriji, facebook je ostao suzdržan po tom pitanju budući da nije bilo očiglednih negativnih utjecaja na popularnost i upotrebu web stranice. Između ostalog, američki policijski službenici i vojnici upotrebljavali su prepoznavanje lica za identifikaciju mrtvih tijela. Ovom tehnologijom vojska je uspjela potvrditi identitet Osame Bin Ladena. Zakonodavne vlasti usvojile su 2014. godine mobilno prepoznavanje lica te je tehnologija postala neizbježna za maloprodajnu industriju u 2017. godini [5].

Vlade širom svijeta sve više ulažu svoje resurse u tehnologiju prepoznavanja lica, a posebno SAD i Kina koji su vodeći na tržištu prepoznavanja lica. Vlada SAD-a odlučila je poboljšati sigurnost aerodroma sustavom prepoznavanja lica za identifikaciju i registraciju posjetitelja. SAD ima nekoliko država koje su omogućile policijskim službama pretraživanje u bazi podataka – ta pretraživanja uključuju detalje vozačke dozvole i fotografije osobnih iskaznica. Prepoznavanje lica i tehnike pretraživanja mogu se također rabiti u policijskim provjerama. Kina već izvodi nekoliko projekata prepoznavanja lica, dok su ostale zemlje još uvijek u fazi planiranja. Čitav svijet primjenjuje ovu tehnologiju i njene prednosti. U Indiji, banke upotrebljavaju prepoznavanje lica kako bi spriječile prevare na bankomatima. Također se upotrebljava za prijavljivanje dvojnika glasača, provjeru putovnice i vize, vozačke dozvole i sl. Očekuje se da će tehnologija rasti i stvoriti ogromne prihode u narednim godinama. Nadzor i sigurnost glavne su industrije na koje će tehnologija intenzivno utjecati. Škole, sveučilišta, pa čak i zdravstvo, planiraju primijeniti tehnologiju prepoznavanja lica u svojim prostorijama za bolje upravljanje. Komplicirana tehnologija koja se rabi u tehnologiji prepoznavanja lica također ulazi i u industriju robotike [5].

U ovom radu ispitat će se točnosti i pouzdanosti pojedinih algoritama za klasifikaciju lica. Upotrebljavat će se analiza linearnih diskriminanti, logistička regresija, K-najbližih susjeda te SVM klasifikator. Za izdvajanje značajki lica primijenit će se facenet naučeni

model [6] te PCA za smanjivanje dimenzionalnosti podataka. Kao usporedbu testirat će se i već naučena duboka neuronske mreža vgg16 [7]. Kvalitativno i kvantitativno će se prikazati ovisnosti ulaznih parametara o točnosti prepoznavanja lica. U drugom poglavlju rada objašnjene su metode i algoritmi koji se upotrebljavaju, u trećem poglavlju objašnjen je način prikupljanja skupa podataka te predprocesiranje slika i pripremanje za primjenu u prepoznavanju lica te su u četvrtom poglavlju rada odrađena četiri eksperimenta koja prikazuju rezultate klasifikacije pojedinih algoritama. Rad završava petim poglavljem u kojem je dan zaključak.



## 2. ALGORITMI STROJNOG UČENJA

Strojno učenje je praksa upotrebljavanja algoritama da parsiraju (raščlane, analiziraju) podatke, uče iz njih i onda odluče o nečemu ili predvide nešto. Ovo područje znanosti nastalo je iz raspoznavanja uzoraka i računalne teorije učenja u području umjetne inteligencije. To je programiranje računala na način da optimiziraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva [8]. Neka je iskustvo  $E$ , zadatak  $T$  i mjera performansi  $P$ . Cilj je učiti obavljati zadatak  $T$  sve uspješnije s obzirom na mjeru performansi  $P$  kako se iskustvo  $E$  povećava. Područje strojnog učenja jedno je od najbrže rastućih polja računalne znanosti. Poznate su tri vrste strojnog učenja. Nadzirano, nenadzirano i podržano učenje. Kod nadziranog učenja je cilj odrediti nepoznatu funkcionalnu ovisnost između ulaznih veličina i izlazne veličine na temelju podatkovnih primjera, gdje su podatkovni primjeri parovi koji se sastoje od vektora ulaznih veličina i vrijednosti izlazne veličine  $y$ . Ako je diskretna veličina  $y$ , govori se o klasifikacijskom problemu, a ako je veličina  $y$  kontinuirana, tada je to regresijski problem. Kod nenadziranog učenja na raspolaganju su samo podatci o ulaznim veličinama, dok izlazna veličina ne postoji. Potrebno je pronaći pravilnosti u podacima, odnosno između mjernih uzoraka. Podržano učenje omogućava agentu, koji poduzima akcije u svom okolišu da bi riješio problem, da samostalno otkrije optimalno ponašanje metodom pokušaja i pogrešaka. Taj agent dobiva odgovarajuću povratnu informaciju za svaku akciju koju izvede (nagrada ili kazna). Nakon određenog broja pokušaja, agent bi trebao naučiti najbolji način odabira akcije u svakom koraku, što je zapravo niz akcija koji maksimizira ukupnu nagradu [8].

Algoritmi koji će se primijenjivati u ovom radu su logistička regresija,  $k$ -najbližih susjeda, analiza linearnih diskriminanti (engl. LDA) te SVM klasifikator. Metode koje će se upotrebljavati za smanjenje dimenzionalnosti i izdvajanje značajki će biti analiza glavnih komponenti (engl. PCA) te ranije spomenuti, unaprijed naučeni facenet model [6].

### 2.1. Analiza glavnih komponentata (PCA)

Analiza glavnih komponentata (engl. Principal Component Analysis) je metoda smanjenja dimenzija koja se često upotrebljava za smanjenje dimenzionalnosti velikih skupova podataka, pretvarajući veliki skup varijabli u manji koji još uvijek sadrži većinu informacija u velikom skupu. Smanjivanje broja varijabli u skupu naravno negativno utječe na točnost, ali trik u smanjenju dimenzionalnosti je razmjena malo točnosti za jednostavnost

izračuna. Budući da je manji skup lakše za istražiti i vizualizirati te olakšava analiziranje podataka, to čini stvari lakše za algoritme strojnog učenja. Ideja je jednostavna, smanjiti broj varijabli skupa podataka, a pritom sačuvati što više informacija.

Slika lica, predstavljena dvodimenzionalnom matricom, se može prikazati kao jednodimenzionalan vektor spajanjem pojedinih redova (ili stupaca) u jedan vektor [9]. Pretpostavimo  $M$  vektora veličine  $N$  (broj redova \* broj stupaca slike) te  $p_j$  koja predstavlja vrijednost piksela.

$$x_i = [p_1 \dots p_N]^T, i = 1, \dots, M \quad (2-1)$$

Slike su usrednjavane oduzimanjem srednje vrijednosti od svakog vektora slike. Neka  $m$  predstavlja srednju vrijednost vektora slike.

$$m = \frac{1}{M} \sum_{i=1}^M x_i \quad (2-2)$$

I neka je  $\omega_i$  srednja vrijednost vektora slike.

$$\omega_i = x_i - m \quad (2-3)$$

Cilj je pronaći skup  $e_i$  koji imaju najveću moguću projekciju na svaki  $\omega_i$ , te pronaći skup  $M$  ortonormalnih vektora  $e_i$  za koje je vrijednost

$$\lambda_i = \sum_{n=1}^M (e_i^T \omega_n)^2 \quad (2-4)$$

maksimalna s obzirom na ortonormalna ograničenja

$$e_i^T e_k = \delta_{ik} \quad (2-5)$$

Pokazano je da su  $e_i$  i  $\lambda_i$  svojstveni vektori i svojstvene vrijednosti matrice kovarijance

$$C = WW^T \quad (2-6)$$

gdje je  $W$  matrica sastavljena od stupaca vektora  $\omega_i$  postavljenih jedan pored drugog. Veličina  $C$  je  $N \times N$  što bi moglo biti poprilično veliko. Na primjer, slike veličine  $64 \times 64$  stvaraju matricu kovarijance veličine  $4096 \times 4096$ . Nije praktično za računanje svojstvenih vektora direktno iz matrice kovarijance  $C$ . Teorem iz linearne algebre kaže da se vektori  $e_i$  i

skalarnu vrijednost  $\lambda_i$  mogu dobiti računanjem svojstvenih vektora i svojstvenih vrijednosti  $M \times M$  matrice  $W^T W$ . Neka su  $d_i$  i  $\mu_i$  svojstveni vektori i svojstvene vrijednosti matrice  $W^T W$ .

$$W^T W d_i = \mu_i d_i \quad (2-7)$$

Kada se pomnože obje strane s  $W$

$$W W^T (W d_i) = \mu_i (W d_i) \quad (2-8)$$

što znači da su prvih  $M - 1$  svojstvenih vektora  $e_i$  i svojstvenih vrijednosti  $\lambda_i$  od  $W W^T$  dobiveni pomoću  $W d_i$  i  $\mu_i$ .  $W d_i$  mora biti normaliziran da bi mogao biti jednak  $e_i$ . Budući da se zbraja konačan broj vektora slika,  $M$ , rang matrice kovarijance ne može biti veći od  $M - 1$  (ovaj -1 dolazi od oduzimanja srednjeg vektora  $m$ ). Svojstveni vektori matrice kovarijance koji odgovaraju svojstvenim vrijednostima različitim od nule stvaraju ortonormalnu bazu za podprostor unutar kojeg se većina slikovnih podataka može predstaviti s malom pogreškom. Svojstveni vektori su sortirani od više na niže prema odgovarajućim svojstvenim vrijednostima. Svojstveni vektor povezan s najvećom svojstvenom vrijednosti je onaj koji odražava najveću varijancu u slici. Odnosno, najmanja svojstvena vrijednost povezana je sa svojstvenim vektorom koji pronalazi najmanju varijancu. Smanjuju se eksponencijalno, što znači da je otprilike 90% ukupne varijance sadržano u prvih 5-10% dimenzija. Slika lica se može projicirati na  $M'$  (puno manji od  $M$ ) dimenzija računajući

$$\Omega = [v_1 v_2 \dots v_{M'}]^T \quad (2-9)$$

gdje je  $v_i = e_i^T \omega_i$ .  $v_i$  je  $i$ -ta koordinata slike lica u novom prostoru, koja je postala glavna komponenta. Vektori  $e_i$  su također slike svojstvenih lica, tzv. „*eigenlica*“. Na njih se može gledati kao na slike i doista izgledaju poput lica. Dakle,  $\Omega$  opisuje odnos svakog svojstvenog lica u predstavljanju slike lica tretirajući svojstveno lice kao osnovu za sliku lica. Najjednostavnija metoda za određivanje koja klasa lica najbolje opisuje ulazne slike lica je pronaći klasu lica  $k$  koja minimalizira Euklidsku udaljenost

$$\epsilon_k = ||(\Omega - \Omega_k)|| \quad (2-10)$$

gdje je  $\Omega_k$  vektor koji opisuje  $k$ -tu klasu lica. Ako je  $\epsilon_k$  manji od neke unaprijed definirane granice  $\Theta_\epsilon$ , lice se klasificira da pripada klasi  $k$ .

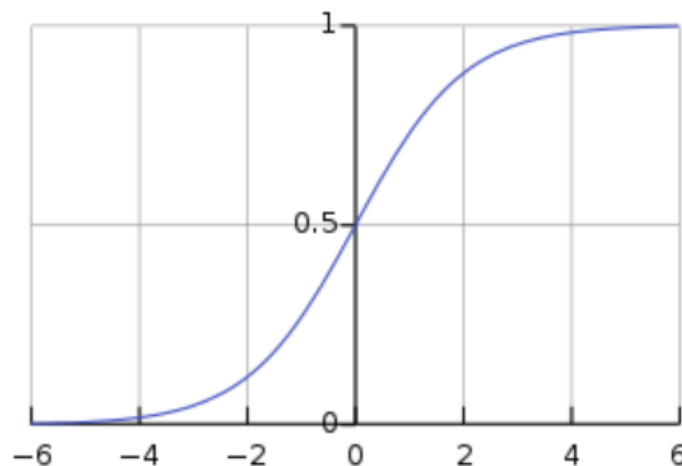
Analiza glavnih komponenti, koja je predstavljena svojim skupom vektora za učenje, određuje bazne vektore prostora. Ti bazni vektori, zapravo svojstveni vektori, određeni pomoću PCA algoritma su u smjeru najveće varijance skupa vektora za učenje. To su ranije spomenuta svojstvena lica. Svako svojstveno lice se može gledati kao značajka. Kada se određeno lice projektira na prostor lica, njegov vektor u prostoru lica opisuje važnost pojedine značajke na licu. Lice se u prostoru lica izražava svojim koeficijentima svojstvenih lica ili težinama. Može se upravljati velikim ulaznim vektorom, slikom lica, samo uzimajući njegov umanjeni vektor težina u prostoru lica. To znači da se može rekonstruirati izvorno lice s nekom pogreškom, jer je dimenzionalnost prostora slike puno veća od one u prostoru lica [9].

## 2.2. Logistička regresija

Logistička regresija [10] algoritam je strojnog učenja koji se upotrebljava za probleme klasifikacije. To je algoritam predviđanja i temelji se na konceptu vjerojatnosti. Sigmoidna funkcija mapira svaku vrijednost u drugu koja ima vrijednost između 0 i 1. U strojnom učenju, sigmoidna funkcija se upotrebljava za mapiranje predviđanja i vjerojatnosti. Njena formula glasi:

$$\sigma(t) = \frac{1}{1+e^{-t}} \quad (2-11)$$

Izgled funkcije prikazan je na slici 2.1.



**SI.2.1.** Izgled sigmoidne funkcije [11]

Neka je  $t$  linearna funkcija u regresijskom modelu.

$$t = \beta_0 + \beta_1 x \quad (2-12)$$

Sada logistička funkcija ima oblik:

$$f(x) = \sigma(t) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \quad (2-13)$$

Gdje  $f(x)$  predstavlja vjerojatnost dobrog predviđanja varijable  $Y$ , dok se  $\beta_0$  i  $\beta_1$  dobiju iz skupa podataka za učenje, odnosno treniranje. S obzirom da je logistička regresija uglavnom klasifikacija binarnih slučajeva, na skupu u ovom radu će se morati primijenjivati višeklasna binarna logistička regresija. Drugim riječima umjesto da  $y$  može poprimiti vrijednosti 0 i 1, može poprimiti vrijednosti od 1 do  $n$ . U suštini vrši se binarna klasifikacija više puta, jednom za svaku klasu. Za svaku se klasifikaciju uzima jedna klasa na jednu stranu, te sve ostale klase na drugu stranu i zatim se odabire klasa s najvećom vjerojatnošću.

### 2.3. Analiza linearnih diskriminanti (LDA)

Analiza linearnih diskriminanti [12] (engl. Linear Discriminant Analysis) nije samo alat za smanjenje dimenzionalnosti podataka (za to je već upotrebljen PCA algoritam) već i robustan klasifikacijski algoritam. To je statistička tehnika klasifikacije objekata u međusobno isključive grupe bazirane na mjerenim svojstvima objekata. Unatoč svojoj jednostavnosti, LDA daje pristojne rezultate klasifikacije koji se mogu lako interpretirati. Pri rješavanju problema klasifikacije u stvarnom svijetu, LDA je često prva metoda usporedne analize prije no što se upotrebe složenije metode. Primijenjujući ovu metodu, potrebno je pripaziti na dvije glavne stvari. Koja svojstva objekta će odrediti pripadnost pojedine grupe i koji model ili pravilo najbolje razlučuje pojedine grupe. Za primjer klasifikacije, neka postoji  $K$  klasa te ulazni vektor  $X$ . Vektor  $X$  klasificiran je tako da pripada klasi  $i$  maksimiziranjem  $P(Y=i|X)$ . Klasifikacija se oslanja na Bayesov teorem koji glasi:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \quad (2-14)$$

Ovo klasifikacijsko pravilo kaže da se objekt s najvećom uvjetovanom vrijednošću pridjeljuje grupi. To znači da se može povezati vjerojatnost da  $X$  pripada svakoj klasi, s vjerojatnošću da ulazna vrijednost poprimi vrijednost  $X$ -a u svakoj klasi.  $P(X|Y)$  se može gledati kao funkciju gustoće vjerojatnosti čija vrijednost funkcije govori vjerojatnost ulaznog podatka poprimanja vrijednosti  $X$ . Postoji  $K$  ploha, po jedna za svaku klasu. Ukoliko je vrijednost funkcije grafa

velika, postoji velika vjerojatnost dobivanja vrijednosti  $X$ , ako je vrijednost funkcije mala, i vjerojatnost je mala. Odabire se funkcija gustoće vjerojatnosti koja ima dobra matematička svojstva, normalna distribucija dobivena jednadžbom:

$$\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2} \quad (2-15)$$

Ovo je ključna pretpostavka analize linearnih diskriminanti. Algoritam će bolje raditi ako su ulazne varijable raspodijeljene po Normalnoj distribuciji. Ako se primijeni normalna distribucija na Bayesov teorem dobije se:

$$P(Y = i | X = a) = \frac{\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{X-\mu_i}{\sigma}\right)^2} P(Y=i)}{P(X=a)} \quad (2-16)$$

Za primjer neka klase dijele jednaku varijancu za  $X$ . Želi se pronaći klasa koja maksimizira tu vrijednost. Izraz se pojednostavljuje logaritmiranjem:

$$\log\left(\frac{\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{X-\mu_i}{\sigma}\right)^2} P(Y=i)}{P(X=a)}\right) \quad (2-17)$$

$$= \log\left(\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{X-\mu_i}{\sigma}\right)^2}\right) + \log(P(Y = i)) - \log P(X = a) \quad (2-18)$$

$$= \log\left(\frac{1}{\sqrt{2\pi\sigma}}\right) - \frac{1}{2}\left(\frac{X-\mu_i}{\sigma}\right)^2 + \log(P(Y = i)) - \log P(X = a) \quad (2-19)$$

$$= \log\left(\frac{1}{\sqrt{2\pi\sigma}}\right) - \frac{1}{2}\frac{X^2}{\sigma^2} + \frac{\mu_i X}{\sigma^2} - \frac{1}{2}\frac{\mu_i^2}{\sigma^2} + \log(P(Y = i)) - \log P(X = a) \quad (2-20)$$

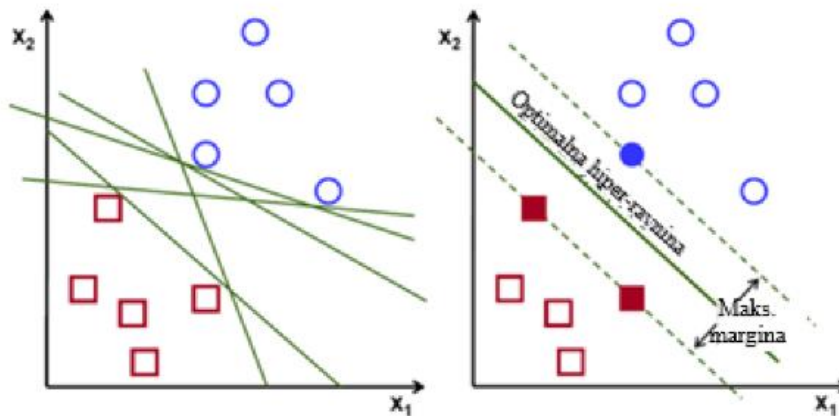
Mnogo ovih izraza ima konstantnu vrijednost kroz sve klase i ne doprinose izračunavanju klase s najvećom vjerojatnošću. Kada se maknu, ostaje funkcija diskriminante:

$$\delta(X) = \frac{\mu_i X}{\sigma^2} - \frac{1}{2}\frac{\mu_i^2}{\sigma^2} + \log(P(Y = i)) \quad (2-21)$$

$X$  se može klasificirati da pripada klasi koja daje najveću vrijednost funkcije diskriminante koja predstavlja razmak „različitosti“ pojedinih grupa [13].

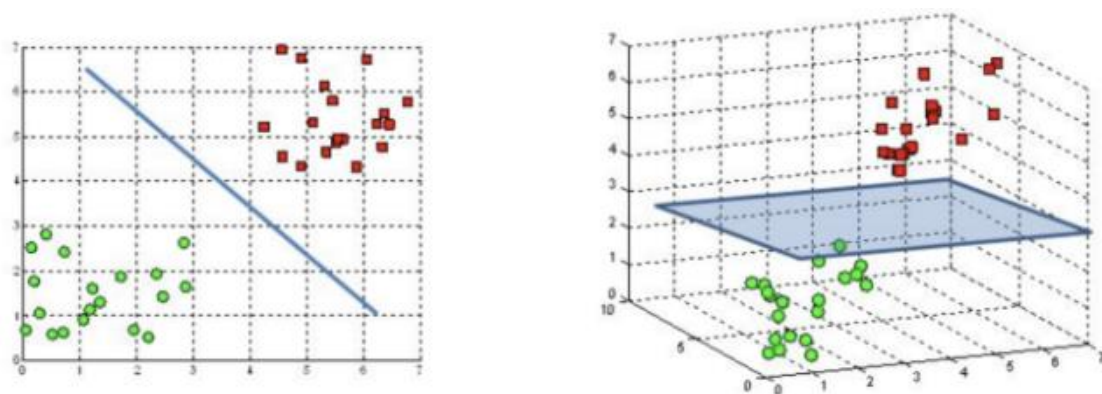
## 2.4. Metoda potpornih vektora (SVM)

Metoda potpornih vektora [14] (engl. Support Vector Machine) je algoritam strojnog učenja koji se upotrebljava za probleme klasifikacije. Cilj metode potpornih vektora je pronaći hiper-ravninu u  $N$ -dimenzionalnom prostoru (gdje je  $N$  broj značajki) koja jasno razvrstava točke podataka.



Sl.2.2. Moguće hiper-ravnine

Da bi se razdvojile dvije klase podatkovnih točaka, moguće je odabrati mnogo hiper-ravnina. Cilj je pronaći ravninu koja ima maksimalnu granicu, tj. maksimalnu udaljenost između podatkovnih točaka obje klase. Maksimiziranje margine omogućava da se buduće točke klasificiraju s većom pouzdanošću.



Sl.2.3. Hiper-ravnine u 2D i 3D prostoru [14]

Hiper-ravnine su granice odluka koje pomažu u klasificiranju podataka. Točke podataka koje se nalaze s bilo koje strane hiper-ravnine mogu se pripisati različitim klasama. Također,

dimenzija hiper-ravnine ovisi o broju značajki. Ako je broj ulaznih značajki 2, tada je hiper-ravnina linija. Ukoliko je broj značajki 3, tada je hiper-ravnina dvodimenzionalna ravnina. Kada je prisutno  $n$  značajki, postaje teško zamisliti  $n$ -dimenzionalni prostor.

Potporni vektori su točke podataka koje su bliže hiper-ravnini i utječu na poziciju i orijentaciju iste. Upotrebom potpornih vektora, maksimizirana je margina klasifikatora. Uklanjanje potpornih vektora mijenja poziciju hiper-ravnine. Ovom metodom upotrebljena je izlazna vrijednost linearne funkcije. Ako je izlazna vrijednost veća od 1, identificira se s jednom klasom, a ako je vrijednost  $-1$ , identificira se drugom klasom. Kombiniranjem više binarnih klasifikatora, dobije se višeklasni klasifikator [14].

U SVM algoritmu, funkcija koja pomaže maksimiziranju margine je funkcija gubitka zavisnosti (eng. hinge loss):

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \quad (2-22)$$

Ta funkcija se može prikazati kao:

$$c(x, y, f(x)) = (1 - y * f(x))_+ \quad (2-23)$$

Trošak funkcije je 0 ako su predviđena vrijednost i stvarna vrijednost istog predznaka. Ako nisu, izračunava se vrijednost gubitka. Također se dodaje i parametar regularizacije funkcije troška. Cilj regularizacijskog parametra je izbalansirati maksimiziranje margine i gubitak. Nakon što se doda parametar regularizacije, funkcija troška izgleda ovako:

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^n (1 - y(x_i, w))_+ \quad (2-24)$$

Sad kada postoji funkcija gubitka, parcijalno se derivira s obzirom na težine da bi se pronašli gradijenti. Upotrebom gradijenata, mogu se ažurirati težine. Gradijenti su prikazani ispod.

$$\frac{\delta}{\delta w_k} \lambda \|w\|^2 = 2\lambda w_k \quad (2-25)$$

$$\frac{\delta}{\delta w_k} (1 - y_i(x_i, w))_+ = \begin{cases} 0, & \text{if } y_i(x_i, w) \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases} \quad (2-26)$$



Kada nema krive klasifikacije, odnosno model točno predvidi klasu podatkovne točke, potrebno je samo ažurirati gradijent regularizacijskog parametra:

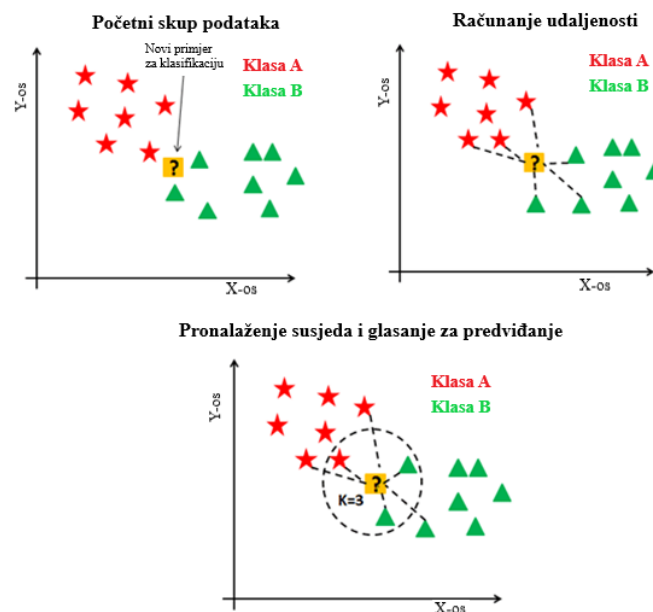
$$\omega = \omega - \alpha * (2\lambda\omega) \quad (2-27)$$

Kada postoji slučaj krive klasifikacije, odnosno model napravi pogrešku u predviđanju klase, u obzir se uzima i gubitak zajedno s regularizacijskim parametrom da bi se ažurirao gradijent:

$$\omega = \omega + \alpha * (y_i * x_i - 2\lambda\omega) \quad (2-28)$$

## 2.5. K najbližih susjeda

K najbližih susjeda algoritam je strojnog učenja koji se upotrebljava za klasifikacijske probleme.  $K$  predstavlja broj susjeda, koji je odlučujući faktor u ovom algoritmu.  $K$  je najčešće neparan broj ukoliko postoji samo dvije klase. Neka je  $P_1$  točka koju treba predvidjeti. Prvo se pronade  $k$  najbližih točaka do  $P_1$ , a zatim se točka razvrstava većinom glasova njenih  $k$ -susjeda. Svaki objekt glasa za svoju klasu, te se klasa s najviše glasova uzima kao predviđanje. Za pronalaženje najbližih sličnih točaka primjenjuju se mjere udaljenosti poput Euklidske udaljenosti, Hammingove udaljenosti, Manhattan udaljenosti te Minkowski udaljenosti. Koraci algoritma su računanje udaljenosti, pronalaženje najbližih susjeda te glasanje za predviđanje [15]. Na slici 2.4. može se vidjeti primjena koraka za algoritam  $k$  najbližih susjeda.



SI.2.4. Koraci algoritma  $k$  najbližih susjeda

$K$  najbližih susjeda najbolje funkcionira s malim brojem značajki. Može se reći da kad se poveća broj značajki, algoritam zahtijeva više podataka. Povećanje dimenzija također dovodi do problema prekomjernog prilagođavanja trening setu, tzv. *overfitting*. Da bi se to izbjeglo, potrebni podaci moraju rasti eksponencijalno s brojem dimenzija. Da bi se riješio problem prevelike dimenzionalnosti, potrebno je primijeniti neku od tehnika smanjenja dimenzionalnosti, poput analize osnovnih komponenata. Broj susjeda je hiperparametar koji je potrebno odabrati dok se određuje model.  $K$  se može smatrati kontrolnom varijablom za model predviđanja. Eksperimentalno ne postoji optimalan broj susjeda za sve vrste podataka. U slučaju malog broja susjeda, šum će imati više utjecaja, dok će velik broj susjeda biti problem zbog vremena izračuna. Istraživanja su također pokazala da je mali broj susjeda najfleksibilnije rješenje koje će imati niski bias, ali visoku varijancu, dok će veći broj susjeda imati glađu granicu odluke, što znači manju varijancu, ali veliki bias. Znanstvenici uglavnom upotrebljavaju neparni broj susjeda ukoliko je broj klasa paran [15].

## 2.6. Konvolucijske neuronske mreže

Konvolucijska neuronska mreža (CNN) je algoritam dubokog učenja koji može uzeti sliku, dodijeliti važnost različitim objektima na slici i biti u mogućnosti razlikovati jedan objekt od drugog. Pretprocesiranje potrebno za CNN je znatno manje u usporedbi s ostalim algoritmima. Dok se u primitivnim metodama filtri izrađuju ručno, uz dovoljno treninga, CNN ima mogućnost naučiti te filtre. Arhitektura mreže je analogna povezanosti neurona u ljudskom mozgu. Pojedini neuroni reagiraju na podražaje samo u ograničenom području vidnog polja. Zbir takvih polja preklapa se tako da pokrije čitavo vizualno područje.

CNN se sastoji od nekoliko slojeva od kojih je najvažniji konvolucijski sloj [16]. On nastoji sačuvati strukturu ulaznog podatka jer svaku komponentu (RGB) gleda i procesira zasebno. Uloga konvolucijske mreže je da smanji sliku u formu koju je lakše procesirati, bez gubitka značajki koje su kritične za dobro predviđanje. Filter se pomiče preko slike s lijeva na desno i za svako pomicanje množi matricu filtra s elementima na slici preko kojeg je postavljen. Cilj konvolucije je predstaviti značajke kao što su oblici pomoću značajki nižih razina poput rubova [16].

U ovom radu upotrebljavat će se već naučene konvolucijske mreže facenet [6] te vgg16 [7]. Facenet je tensorflow implementacija prepoznavanja lica objašnjena u znanstvenom članku [17]. Upotrebljavat će se model naučen na CASIA-WebFace skupu podataka. VGG16

naučeni model će se upotrijebiti za izdvajanje značajki jednako kao i facenet. Facenet model ima izlazni vektor značajki veličine 128, dok model vgg16 arhitekture ima izlazni vektor značajki veličine 512.

### 3. PRIPREMA SKUPA PODATAKA

Prvi korak u testiranju algoritama prepoznavanja lica je prikupiti skup podataka. Postoje velike baze lica na internetu, raznih značajki, poput emocija, no u ovom radu će biti napravljen vlastiti skup podataka. Slike će biti fotografije lica javnih osoba iz svijeta glume, sporta, politike i sl. Skup će se sastojati od 50 različitih klasa tj. osoba, gdje unutar svake klase imamo 100 slika. Skup za učenje će se sastojati od 75, dok će se testni skup sastojati od preostalih 25 slika. Slike su strukturirane tako da se unutar trening mape nalazi 50 mapa s imenima klasa (ili osoba) te unutar tih mapa po 75 slika za svaku klasu. Unutar testne mape nalazi se 50 mapa s imenima istih klasa te 25 slika za svaku klasu.

#### 3.1. Prikupljanje podataka

Bilo bi vrlo zamorno ručno skidati toliku količinu slika s interneta te spremati u odgovarajuće mape. Zbog toga je rabljena skripta *search\_bing\_api.py* [18]. Sve što je potrebno napraviti je registrirati se na Bing Image Search API preko svog github ili google računa. Nakon registracije se dobiju dva API ključa za upotrebu u trajanju od tjedan dana i unesu se na mjesto u kod. Unutar koda unesen je i maksimalan broj rezultata koji se želi preuzeti. Tako je broj rezultata ograničen na 300, kako bi se izbjegao nedovoljan broj slika zbog lošeg formata, loše kvalitete, krive slike i sl.

```
# set your Microsoft Cognitive Services API key along with (1) the
# maximum number of results for a given search and (2) the group size
# for results (maximum of 50 per request)
API_KEY = "INSERT_YOUR_API_KEY_HERE"
MAX_RESULTS = 300
GROUP_SIZE = 50

# set the endpoint API URL
URL = "https://api.cognitive.microsoft.com/bing/v7.0/images/search"
```

Sl.3.1. Podešavanje *search\_bing\_api.py* skripte [18]

#### 3.2. Filtriranje podataka

Za filtraciju nepotrebnih slika, zbog loše kvalitete, pogrešnog sadržaja, dupliciranih slika i sl., napravljena je funkcija koja će to isto napraviti. Prvo su filtrirane datoteke koje nisu .jpeg ili .png formata. Za pronalazak duplikata rabi se „md5“ algoritam za generiranje hash vrijednosti svake datoteke te se sprema u rječnik ukoliko se već ne nalazi u njemu, u protivnom se sprema u listu duplikata te se na kraju sve slike iz liste duplikata brišu [19].

```

def obrisi_duplikate(direktorij):

    os.chdir(direktorij)
    print ("\nMapa za pretražiti duplikate: ", os.getcwd())

    lista_datoteka = os.listdir()
    print ("\nPronađeno {} datoteka u mapi!".format(len(lista_datoteka)))

    i = 0

    duplikati = []
    hash_kljucevi = dict()
    for index, datoteka in enumerate(os.listdir('.')):
        if os.path.isfile(datoteka):
            if datoteka.endswith(".jpg") or datoteka.endswith(".png"):
                with open(datoteka, 'rb') as f:
                    hash_vrijednost = hashlib.md5(f.read()).hexdigest()
                    if hash_vrijednost not in hash_kljucevi:
                        hash_kljucevi[hash_vrijednost] = index
                    else:
                        duplikati.append((index,hash_kljucevi[hash_vrijednost]))
            else:
                print("Brisanje {}".format(datoteka))
                os.remove(os.path.join(direktorij,datoteka))
                print("{} obrisana!".format(datoteka))

    print ("\nPronađeno {} duplikata u mapi!\n".format(len(duplikati)))
    print (duplikati)
    for index in duplikati:
        os.remove(lista_datoteka[index[0]])

dataset = 'D:\\Diplomski_v2\\dataset\\dataset\\'
datoteke = os.listdir(dataset)

for datoteka in datoteke:
    putanja = os.path.join(dataset,datoteka)
    obrisi_duplikate(putanja)

```

### Sl.3.2. Funkcija za brisanje nepravilnih i dupliciranih slika

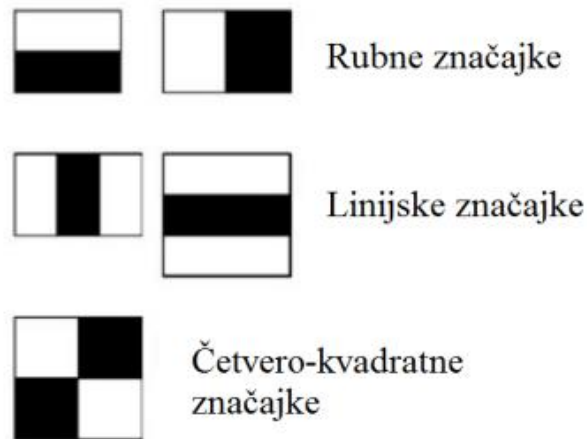
## 3.3. Detekcija i poravnanje lica

Kako bi se uspješno proveli algoritmi prepoznavanja lica, potrebno je ista detektirati na slici. Za detekciju lica primijenit će se Viola-Jones algoritam koji upotrebljava kaskadu određenih značajki da bi donio odluku o tome nalazi li se lice na slici ili ne. Također, kako bi se povećala preciznost samih algoritama, detektirana lica će biti poravnata s obzirom na položaj očiju, nosa i usta. Dakle, sve slike će biti ispravljene da im se na istim mjestima nalaze prethodno navedene značajke zbog lakše i točnije klasifikacije.

### 3.3.1. Viola-Jones algoritam

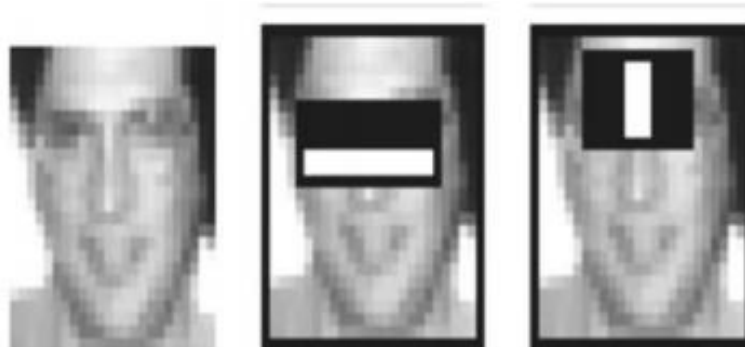
Viola-Jones algoritam metoda je brzog detektiranja objekata na slici upotrebom Haarovih kaskadnih klasifikatora, točnije Haarovih značajki. Tu metodu u svom radu „Rapid Object Detection using a Boosted Cascade of Simple Features“, predložili su Paul Viola i Micheal Jones [20]. Haarove značajke slične su konvolucijskim kernelima, koji će biti opisani kasnije u radu, a upotrebljavaju se za detekciju određene značajke na slici. Primjenjuje se tako

da se određenom značajkom prelazi preko elemenata slike te se u njoj traže određene odgovarajuće karakteristike lica. Na slici 3.3. dani su tipovi osnovnih Haarovih značajki [21].



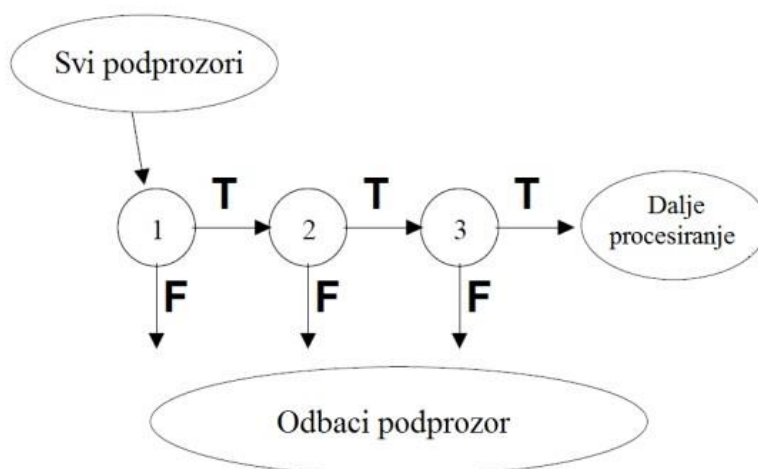
**Sl.3.3.** Osnovne Haarove značajke [21]

Sve ostale značajke dobivaju se translacijom, skaliranjem ili rotacijom osnovnih značajki. Primjena svake značajke rezultira jedinstvenom vrijednošću koja je izračunata oduzimanjem sume intenziteta elemenata slike ispod bijelog pravokutnika od sume intenziteta elemenata slike ispod crnog pravokutnika značajke. Ako se promatra kao konvolucijski kernel, crno područje ima vrijednost 1, bijelo -1. Linijska značajka je, primjerice, namijenjena detekciji nosa, jer je područje nosa svjetlije od ruba nosa, koji je tamniji. Primjenom te značajke na cijelu sliku, dobit će se velike vrijednosti samo na lokacijama onih elemenata slike koji pripadaju nosu. Rubna značajka će najbolje izdvojiti područje očiju, zbog svijetlih područja ispod očiju i tamnih područja samih očiju. Jednako kao i značajka za nos, ova će značajka dati veće vrijednosti na lokacijama gdje su oči [21].



**Sl.3.4.** *Primjena Haarovih značajki na sliku (rubna značajka služi za detekciju očiju, dok linijska služi za detekciju nosa) [21]*

Sada se sve moguće veličine i lokacije svake jezgre upotrebljavaju za izračunavanje mnogo značajki te već pri veličini od 24x24 piksela postoji preko 160000 značajki [22]. Primjenom tehnike zvanom integralna slika, ubrzan je postupak proračuna značajki jer zahtijeva vrijednosti samo četiri piksela. No, među svim tim značajkama, većina je nebitna. Da bi se što brže riješio problem detekcije lica te da bi se algoritam mogao primijeniti u stvarnom vremenu, uvodi se AdaBoost algoritam kojim se eliminiraju sve nebitne značajke iz dane slike. Ideja algoritma je da stvori jedan jaki klasifikator pomoću više slabih klasifikatora. Linearnom kombinacijom slabih klasifikatora dobije se jaki klasifikator. Kaskadnim ulančavanjem jakih klasifikatora znatno se ubrzava postupak detekcije lica. Na slici 3.5. prikazan je primjer kaskade klasifikatora. Prvi eliminira velik broj uzoraka uz vrlo malo procesiranja, sljedeće razine eliminiraju još negativnih uzoraka, ali uz više procesiranja [21].



**Sl.3.5.** *Kaskada klasifikatora [21]*

Viola-Jones algoritam ima ekstremno brz proračun značajki, učinkovit je u odabiru istih te je detektor invarijantan na veličinu i lokaciju objekta. Problem je što se može primijeniti samo za slike lica sprijeda, ne može izaći na kraj s rotacijom i vrlo je osjetljiv na uvjete osvjetljenja.

Implementacija detekcije lica u Pythonu pomoću Haarovih klasifikatora radi se pomoću OpenCV biblioteke. Funkcija `cv2.CascadeClassifier` rabi prethodno naučeni klasifikator. Ti klasifikatori se mogu pronaći na stranici OpenCV-a [23]. Upotrebom metode `detectMultiScale` pronalazi se područje od interesa (engl. Region of Interest) za svako detektirano lice. U ovom slučaju, eliminirane su sve slike s više lica ili bez lica, jer za svaku klasu treba samo jedno lice na slici. Željena veličina slika je 160x160 piksela pa će svako lice biti promijenjeno na tu veličinu. Na slici 3.6. prikazano je kako funkcija prepoznaje lice.



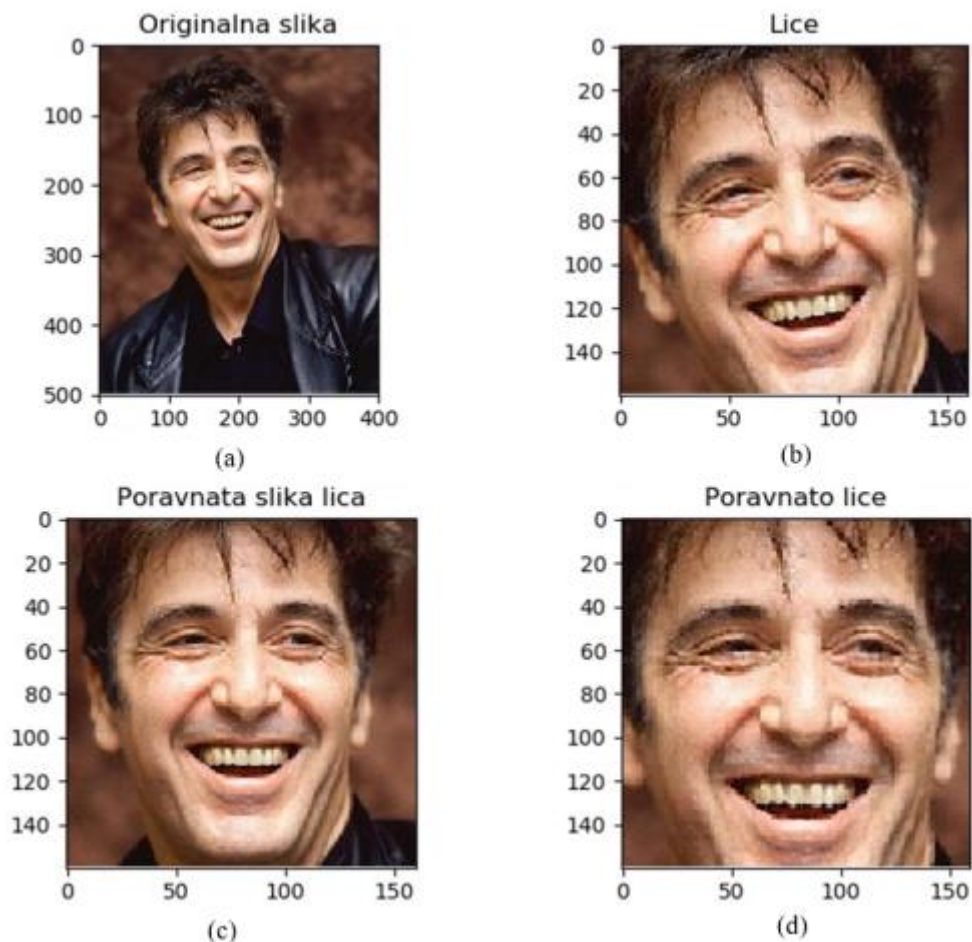
**Sl.3.6.** *Primjer rada Haarovih klasifikatora za detekciju lica*

### 3.3.2. Poravnanje lica

Važno je primijetiti kako je lice dobro izrezano, međutim nije poravnato da ne bude rotirano. Za poravnanje lica upotrebljava se funkcija iz `dlib` biblioteke za rotiranje lica [24]. Na slici 3.7. može se vidjeti primjena detekcije lica pomoću Haarovih klasifikatora na



originalnu sliku i poravnatu sliku lica. Ovim postupkom pretprocesiranja pospješuje se točnost daljnjeg prepoznavanja lica. Prema prikazanom, poravnata slika veća je no samo lice, stoga će biti potrebno nad tim poravnatim slikama izvršiti detekciju lica, te rezultat te detekcije, točnije područje od interesa, spremi u polje za kasniju primjenu u algoritmima za prepoznavanje lica.



*Sl.3.7. Primjena (a) detekcije lica na originalnu, (b) rezultat detekcije, primjena (c) detekcije na poravnatoj i rotiranoj slici lica te rezultat (d)*

### 3.4. Priprema lica za upotrebu u klasifikaciji

Za lakšu manipulaciju skupom podataka, a da uz to nije potrebno svaki puta detektirati lica jer je vremenski zahtjevno, detektirana lica bit će spremljena u polje te pohranjena za upotrebu kasnije. Na slici 3.8. prikazana je funkcija koja sprema liste dohvaćenih lica i imena u dva polja koja su izmiješana zajedno da bi se očuvalo određeno lice uz određeno ime. Prvo su inicijalizirane dvije liste u koje će se spremati lica i imena. Zatim je potrebno proći kroz

pojedine direktorije i zapisati klase (jer je ime direktorija ujedno i ime klase). Tada pomoću funkcije `get_face` treba pronaći lice na slici te isto spremi u listu. Nakon toga liste se miješaju i pretvaraju u polje. Nakon što su lica pohranjena i spremna za upotrebu, na isti način spremaju se i izdvojene značajke primjenom PCA i faceneta te provode eksperimenti.

```
def load_dataset(directory):
    X, y = list(), list()
    for root, dirs, files in os.walk(directory):
        if len(dirs) == 0:
            class_path = os.path.join(directory, root).replace("\\", "/")
            class_name = os.path.basename(class_path)
            print("Working on {} folder".format(class_name))
            for file in files:
                file_path = os.path.join(class_path, file).replace("\\", "/")
                face = get_face(file_path, cascade, (160, 160))
                if face is not None:
                    X.append(face)
                    y.append(class_name)
                else:
                    print("Skipped adding {}".format(file_path.split("/")[-1]))
    c = list(zip(X, y))
    random.shuffle(c)
    X, y = zip(*c)
    return np.asarray(X), np.asarray(y)

train_path = "D:\\Diplomski\\Experiment_One_Two\\train"
test_path = "D:\\Diplomski\\Experiment_One_Two\\test"
cascade = "D:\\Diplomski\\Cascades\\haarcascade_frontalface_default.xml"

# load train dataset
print("Loading training dataset..")
trainX, trainy = load_dataset(train_path)
print(trainX.shape, trainy.shape)
# load test dataset
print("Loading testing dataset..")
testX, testy = load_dataset(test_path)
print(testX.shape, testy.shape)
# save arrays to one file in compressed format
print("Saving numpy arrays..")
np.savez_compressed('Small_dataset_shuffled.npz', trainX, trainy, testX, testy)
print("Finished!")
```

### SI.3.8. Funkcija za spremanje skupa podataka u polja

Sad kada je skup podataka u obliku datoteke, nije nužno trošiti mnogo vremena za izvođenje eksperimenata jer je potrebno samo učitati skup podataka koji je sada predstavljen samo s poljima lica te zauzima višestruko manji prostor od cijelih fotografija. Na slici 3.9. prikazan je primjer učitavanja PCA značajki s 93% varijance te pretvaranje imena u ID za upotrebu u klasifikaciji.

```
data = np.load('Embeddings_PCA_93.npz')
trainX_pca, trainy, testX_pca, testy = data['arr_0'], data['arr_1'], data['arr_2'], data['arr_3']
imena_klasa = sorted(set(trainy))

# pretvaranje labela u ID
out_encoder = LabelEncoder()
out_encoder.fit(trainy)
trainy = out_encoder.transform(trainy)
testy = out_encoder.transform(testy)
```

### SI.3.9. Funkcija za učitavanje skup podataka

## 4. EKSPERIMENTALNA ISTRAŽIVANJA

Za mjerenje točnosti pojedinog algoritma upotrebljavat će se izvještaj klasifikacije koji je također implementiran unutar biblioteke *scikit learn*. Izvještaj klasifikacije rabi se za mjerenje kvalitete predviđanja algoritma klasifikacije. Kao što je vidljivo u izvještaju, ima nekoliko metrika koje se računaju prilikom klasifikacije. One se izračunavaju upotrebljavajući stvarne i lažne pozitivne te stvarne i lažne negativne vrijednosti. Stvarna pozitivna je ako je vrijednost klasificirana kao pozitivna i uistinu jest pozitivna. Stvarna negativna je ako je klasificirana kao negativna i jest negativna. Lažna pozitivna je ako je vrijednost bila negativna, ali je klasificirana kao pozitivna, dok je lažna negativna ako je bila pozitivna, a klasificirana je kao negativna. Prva metrika, preciznost (engl. *precision*), govori koliko predviđenih slika je bilo točno klasificirano. Preciznost je sposobnost klasifikatora da ne označi instancu kao pozitivnu ako je zapravo negativna. Za svaku se klasu definira omjer stvarnih pozitivnih te zbroja stvarnih i lažnih pozitivnih vrijednosti [25].

$$Precision = \frac{True\ Positive}{True\ Positive + True\ Negative}$$

Druga metrika, odziv (engl. *recall*), je sposobnost klasifikatora da pronade sve pozitivne instance. Za svaku se klasu definira kao omjer stvarnih pozitivnih te zbroja stvarnih pozitivnih i lažnih negativnih vrijednosti.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Treća metrika, f1-mjera (engl. *f1-score*), je ponderirana harmonijska sredina preciznosti i recall-a, takva da je najbolja ocjena 1.0, a najgora 0.0. Općenito, f1-mjere manje su od mjera točnosti jer uključuju preciznost i recall u svoje izračune. Ponderirani prosjek f1 mjera trebao bi se rabiti za usporedbu modela klasifikatora, a ne globalne točnosti.

$$F_1\ score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Zadnji parametar, *support*, prikazuje koliko je testnih slika bilo za svaku klasu, u prvom i drugom eksperimentu po 25 slika za svaku od 6 klasa.

Također će se upotrebljavati tzv. matrica zbuđenosti (engl. *confusion matrix*) koja prikazuje 2D polje odnosa stvarnih i lažnih pozitivnih te negativnih vrijednosti.

## 4.1. Eksperiment 1

U prvom eksperimentu upotrebljavat će se PCA algoritam za smanjenje dimenzionalnosti i testirati sa svim ranije navedenim algoritmima za klasifikaciju. Veličina skupa podataka je 6 klasa sa po 100 slika, od kojih je 75 slika za učenja te 25 testnih slika.

### 4.1.1. K najbližih susjeda

Prvi algoritam koji će se upotrebljavati za klasifikaciju našeg skupa podataka je k najbližih susjeda. Upotrebljavat će se klasifikator implementiran unutar *scikit learn* biblioteke, *KNeighborsClassifier()*. Taj klasifikator prima nekoliko parametara, unutar kojih su najvažniji broj susjeda te metrika udaljenosti koja se upotrebljava. Kako bi se pronašli najbolji hiperparametri, upotrebljavat će se *GridSearchCV()* funkcija kojoj će se predati raspon susjeda koje bi testirali te odabrati onaj koji najbolje odgovara za skup podataka. Broj susjeda odabran za PCA od 93% je 5, broj susjeda odabran za PCA od 95% je 9 te je za PCA od 99% odabrano 17 susjeda. U tablici 4.1. prikazan je izvještaj klasifikacije za PCA od 93%, u tablici 4.2. izvještaj klasifikacije za PCA od 95% te u tablici 4.3. izvještaj za PCA od 99%.

**Tab 4.1.** *Izvjestaj klasifikacije za k najbližih susjeda uz PCA od 93%*

KLASA	PRECIZNOST	ODZIV	F1-REZULTAT
Angelina Jolie	0.67	0.72	0.69
Chris Pratt	0.52	0.88	0.66
David Beckham	0.8	0.32	0.46
Meghan Markle	0.78	0.56	0.65
Rowan Atkinson	0.83	0.2	0.32
Tom Hanks	0.43	0.8	0.56
<b>srednja vrijednost</b>	<b>0.67</b>	<b>0.58</b>	<b>0.56</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.58</b>

**Tab 4.2.** Izvještaj klasifikacije za k najbližih susjeda uz PCA od 95%

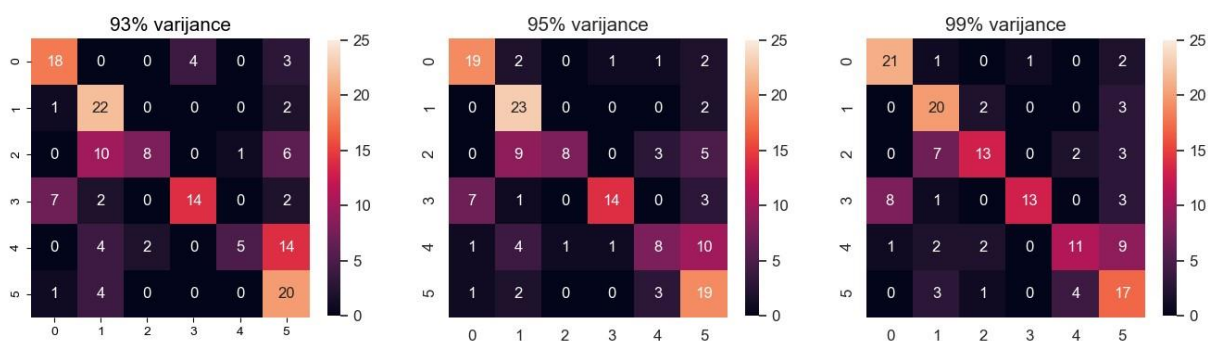
KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.68	0.76	0.72
Chris Pratt	0.56	0.92	0.7
David Beckham	0.89	0.32	0.47
Meghan Markle	0.88	0.56	0.68
Rowan Atkinson	0.53	0.32	0.4
Tom Hanks	0.46	0.76	0.58
<b>srednja vrijednost</b>	<b>0.67</b>	<b>0.61</b>	<b>0.59</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.6067</b>

**Tab 4.3.** Izvještaj klasifikacije za k najbližih susjeda uz PCA od 99%

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.7	0.84	0.76
Chris Pratt	0.59	0.8	0.68
David Beckham	0.72	0.52	0.6
Meghan Markle	0.93	0.52	0.67
Rowan Atkinson	0.65	0.44	0.52
Tom Hanks	0.46	0.68	0.55
<b>srednja vrijednost</b>	<b>0.67</b>	<b>0.63</b>	<b>0.63</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.6333</b>

Kao što se može vidjeti iz izvještaja klasifikacije, točnost algoritma uz upotrebu PCA sa 93% očuvane varijance je 58%, sa 95% varijance je 60,67%, dok je pri 99% varijance 63,33%. Na slici 4.1. prikazana je matrica zbunjenosti za k najbližih susjeda.

## K najbližih susjeda



Sl.4.1. Matrica zbunjenosti za  $k$  najbližih susjeda

Iz matrice zbunjenosti se vidi da se povećanjem varijance generalna točnost popravila, no da su zbog toga neke osobe bile klasificirane s manjom preciznošću.

### 4.1.2. Logistička regresija

Drugi algoritam koji će se upotrijebiti za klasifikaciju je logistička regresija. Klasifikator `LogisticRegression()` također prima nekoliko parametara. Parametar koji će se mijenjati je  $C$ , inverz snage regularizacije. Kao kod SVM-a, manje vrijednosti znače snažniju regularizaciju. U tablici 4.4. prikazan je izvještaj klasifikacije za logističku regresiju uz PCA od 93%, u tablici 4.5. izvještaj uz PCA od 95% te u tablici 4.6. izvještaj uz PCA od 99%. Na slici 4.2. prikazana je matrica zbunjenosti za logističku regresiju.

Tab 4.4. Izvještaj klasifikacije za logističku regresiju uz PCA od 93%

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.84	0.84	0.84
Chris Pratt	1	0.92	0.96
David Beckham	0.83	0.96	0.89
Meghan Markle	0.79	0.88	0.83
Rowan Atkinson	0.95	0.84	0.89
Tom Hanks	0.87	0.8	0.83
<b>srednja vrijednost</b>	<b>0.88</b>	<b>0.87</b>	<b>0.87</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.8733</b>

**Tab 4.5.** Izvještaj klasifikacije za logističku regresiju uz PCA od 95%

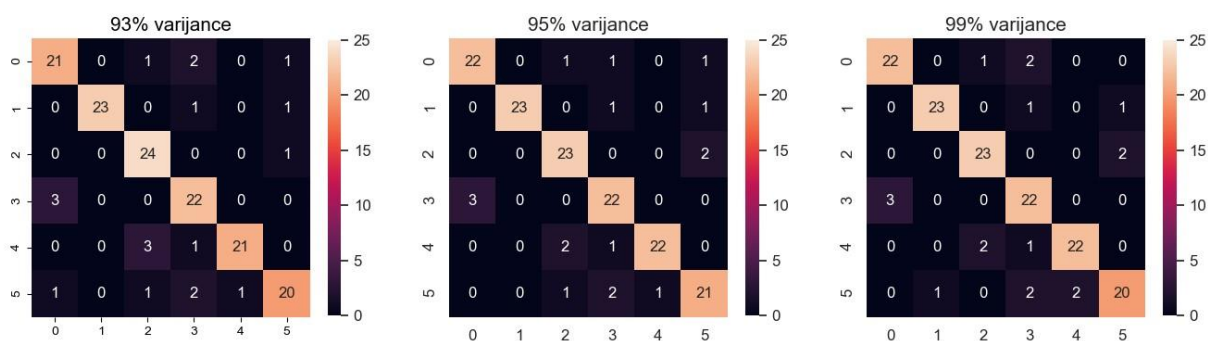
KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.88	0.88	0.88
Chris Pratt	1	0.92	0.96
David Beckham	0.85	0.92	0.88
Meghan Markle	0.81	0.88	0.85
Rowan Atkinson	0.96	0.88	0.92
Tom Hanks	0.84	0.84	0.84
<b>srednja vrijednost</b>	<b>0.89</b>	<b>0.89</b>	<b>0.89</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.8867</b>

**Tab 4.6.** Izvještaj klasifikacije za logističku regresiju uz PCA od 99%

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.88	0.88	0.88
Chris Pratt	0.96	0.92	0.94
David Beckham	0.88	0.92	0.9
Meghan Markle	0.79	0.88	0.83
Rowan Atkinson	0.92	0.88	0.9
Tom Hanks	0.87	0.8	0.83
<b>srednja vrijednost</b>	<b>0.88</b>	<b>0.88</b>	<b>0.88</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.88</b>



## Logistička regresija



### SI.4.2. Matrica zbunjenosti za logističku regresiju

Iz izvještaja se može vidjeti da se točnost logističke regresije na skupu podataka kreće oko 88% za sve tri vrijednosti PCA. Za razliku od k najbližih susjeda, točnost se povećala za 20%, što je poprilično velika razlika.

### 4.1.3. Analiza linearnih diskriminanti

Treći algoritam koji će se upotrebljavati za klasifikaciju je analiza linearnih diskriminanti. Klasifikator *LinearDiscriminantAnalysis()* se ne mora mijenjati jer ima dobro određene početne parametre za klasifikaciju. U tablici 4.7. prikazan je izvještaj klasifikacije za analizu linearnih diskriminanti uz PCA od 93%, u tablici 4.8. izvještaj uz PCA od 95% te u tablici 4.9. izvještaj uz PCA od 99%. Na slici 4.3. prikazana je matrica zbunjenosti.

**Tab 4.7.** Izvještaj klasifikacije za analizu linearnih diskriminanti uz PCA od 93%

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.84	0.84	0.84
Chris Pratt	0.92	0.96	0.94
David Beckham	0.92	0.88	0.9
Meghan Markle	0.92	0.92	0.92
Rowan Atkinson	0.96	0.92	0.94
Tom Hanks	0.85	0.88	0.86
<b>srednja vrijednost</b>	<b>0.9</b>	<b>0.9</b>	<b>0.9</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.9</b>

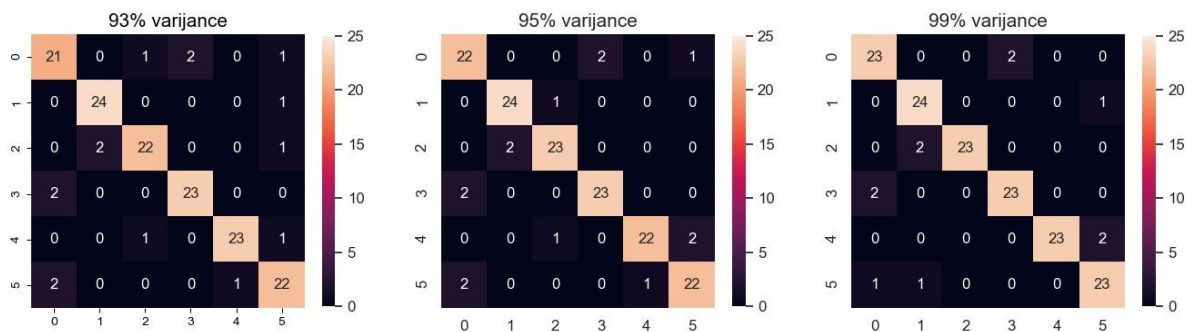
**Tab 4.8.** *Izvyještaj klasifikacije za analizu linearnih diskriminanti uz PCA od 95%*

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.85	0.88	0.86
Chris Pratt	0.92	0.96	0.94
David Beckham	0.92	0.92	0.92
Meghan Markle	0.92	0.92	0.92
Rowan Atkinson	0.96	0.88	0.92
Tom Hanks	0.88	0.88	0.88
<b>srednja vrijednost</b>	<b>0.91</b>	<b>0.91</b>	<b>0.91</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.9067</b>

**Tab 4.9.** *Izvyještaj klasifikacije za analizu linearnih diskriminanti uz PCA od 99%*

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.88	0.92	0.9
Chris Pratt	0.89	0.96	0.92
David Beckham	1	0.92	0.96
Meghan Markle	0.92	0.92	0.92
Rowan Atkinson	1	0.92	0.96
Tom Hanks	0.88	0.92	0.9
<b>srednja vrijednost</b>	<b>0.93</b>	<b>0.93</b>	<b>0.93</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.9267</b>

## Analiza Linearnih Diskriminanti



### SI.4.3. Matrica zbunjenosti za analizu linearnih diskriminanti

Kao što se može vidjeti iz izvještaja, točnost za 93% varijance je 90%, za 95% je 90,67% te za 99% varijance 92,67%. Točnost analize linearnih diskriminanti je za gotovo 5% veća od logističke regresije, te gotovo 30% veća od k najbližih susjeda.

### 4.1.4. SVM klasifikacija

Četvrti algoritam koji će se upotrebljavati je SVM klasifikacija. Klasifikator `SVC()` također prima nekoliko parametara. Mijenjat će se `C`, inverz snage regularizacije, koji kontrolira odnos između dobro određenih granica odluke i točno klasificiranih trening podataka, `kernel-linear`, `rbf` `sigmoid` i `poly`, koji označava tip hiper-ravnine koja će se rabiti, te `gamma` koja određuje koliko se klasifikator prilagođava trening setu. U tablici 4.10. prikazan je izvještaj klasifikacije za SVM uz PCA od 93%, u tablici 4.11. izvještaj uz PCA od 95% te u tablici 4.12. izvještaj uz PCA od 99%. Na slici 4.4. prikazana je matrica zbunjenosti.

**Tab 4.10.** Izvještaj klasifikacije za metodu potpornih vektora uz PCA od 93%

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.88	0.92	0.9
Chris Pratt	0.85	0.92	0.88
David Beckham	0.88	0.92	0.9
Meghan Markle	0.85	0.88	0.86
Rowan Atkinson	0.88	0.88	0.88
Tom Hanks	0.95	0.76	0.84
<b>srednja vrijednost</b>	<b>0.88</b>	<b>0.88</b>	<b>0.88</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.88</b>

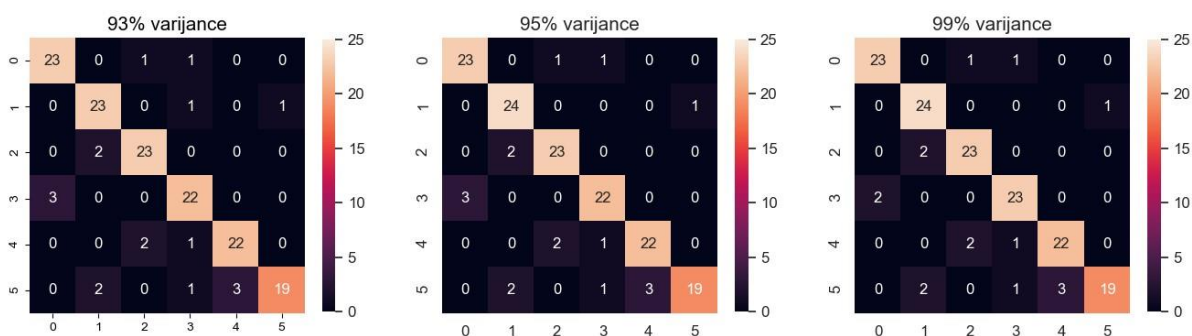
**Tab 4.11.** Izvještaj klasifikacije za metodu potpornih vektora uz PCA od 95%

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.88	0.92	0.9
Chris Pratt	0.86	0.96	0.91
David Beckham	0.88	0.92	0.9
Meghan Markle	0.88	0.88	0.88
Rowan Atkinson	0.88	0.88	0.88
Tom Hanks	0.95	0.76	0.84
<b>srednja vrijednost</b>	<b>0.89</b>	<b>0.89</b>	<b>0.89</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.8867</b>

**Tab 4.12.** Izvještaj klasifikacije za metodu potpornih vektora uz PCA od 99%

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	0.92	0.92	0.92
Chris Pratt	0.86	0.96	0.91
David Beckham	0.88	0.92	0.9
Meghan Markle	0.88	0.92	0.9
Rowan Atkinson	0.88	0.88	0.88
Tom Hanks	0.95	0.76	0.84
<b>srednja vrijednost</b>	<b>0.9</b>	<b>0.89</b>	<b>0.89</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>0.8933</b>

SVM



**Sl.4.4.** Matrica zbunjenosti za SVM klasifikaciju

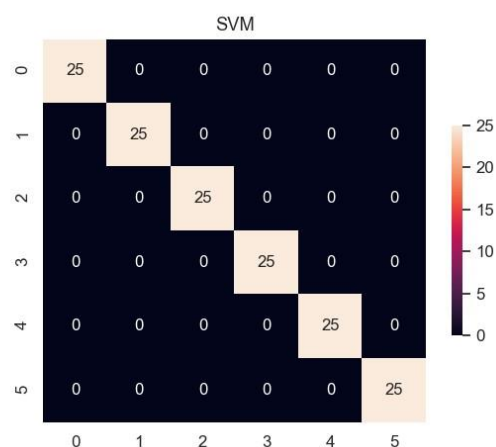
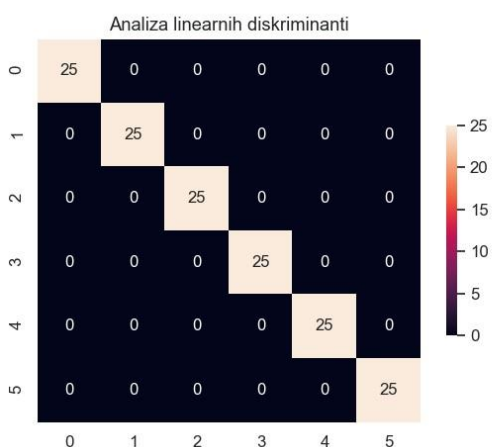
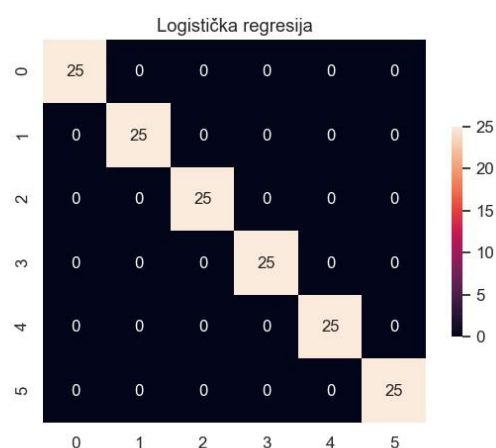
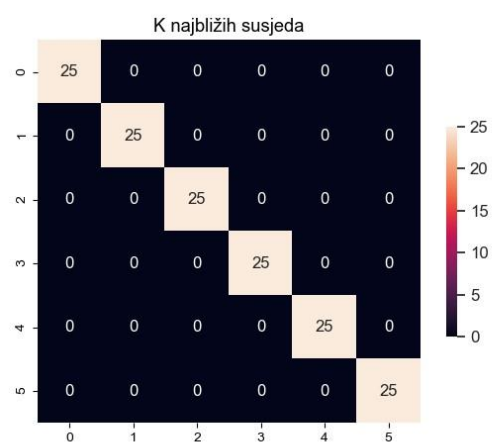
Iz izvještaja klasifikacije može se vidjeti da je točnost 88-89% za sve tri vrijednosti varijance. Točnost SVM klasifikacije veća je od logističke regresije, no manja od analize linearnih diskriminanti za 3%.

## 4.2. Eksperiment 2

U drugom će se eksperimentu upotrebljavati facenet za izdvajanje značajki. Veličina vektora smanjena je s 25600 na 128, što omogućuje već naučena facenet mreža. U tablici 4.13. prikazan je izvještaj klasifikacije za sve algoritme, a na slici 4.5. prikazana je matrica zbunjenosti za iste.

**Tab 4.13.** Izyještaj klasifikacije za facenet za sve algoritme

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Angelina Jolie	1	1	1
Chris Pratt	1	1	1
David Beckham	1	1	1
Meghan Markle	1	1	1
Rowan Atkinson	1	1	1
Tom Hanks	1	1	1
<b>srednja vrijednost</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Ukupan broj slika</b>			<b>150</b>
<b>TOČNOST</b>			<b>1</b>

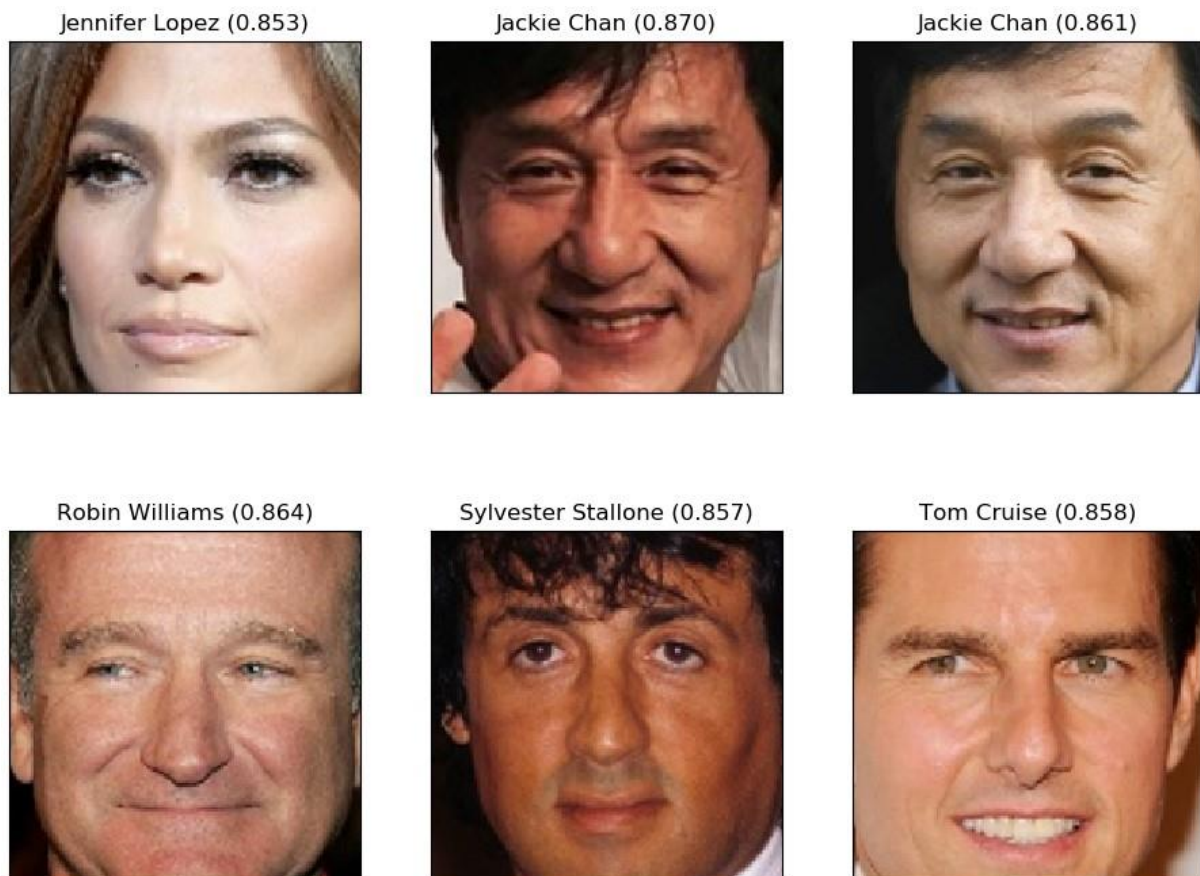


**SI.4.5.** Matrica zbunjenosti za facenet sa 6 klasa

Kao što se može vidjeti iz izvještaja i matrice zbunjenosti, bez obzira na klasifikacijski algoritam koji se rabi za klasifikaciju, točnost je 100%. Razlog tome je relativno jednostavan skup podataka. Ovaj skup podataka sadrži samo 100 slika po osobi, dok je facenet naučen na milijunima slika.

### 4.3. Eksperiment 3

U trećem eksperimentu će se testirati facenet na 50 klasa. Zadržana je struktura od 75 trening slika i 25 testnih slika za pojedinu klasu. Kao što se ustanovilo iz prva dva eksperimenta, SVM klasifikator, LDA klasifikator te klasifikator logističke regresije su superiorniji za prepoznavanje lica. Pošto se LDA algoritam najčešće upotrebljava za smanjenje dimenzionalnosti, a ne klasifikaciju velikog broja klasa, rabić će se SVM klasifikator za sljedeća dva eksperimenta. Na slici 4.6. prikazana su neka lica koja su klasificirana sa sigurnošću manjom od 90%.



Sl.4.6. Vjerojatnost predviđanja klasa uz facenet

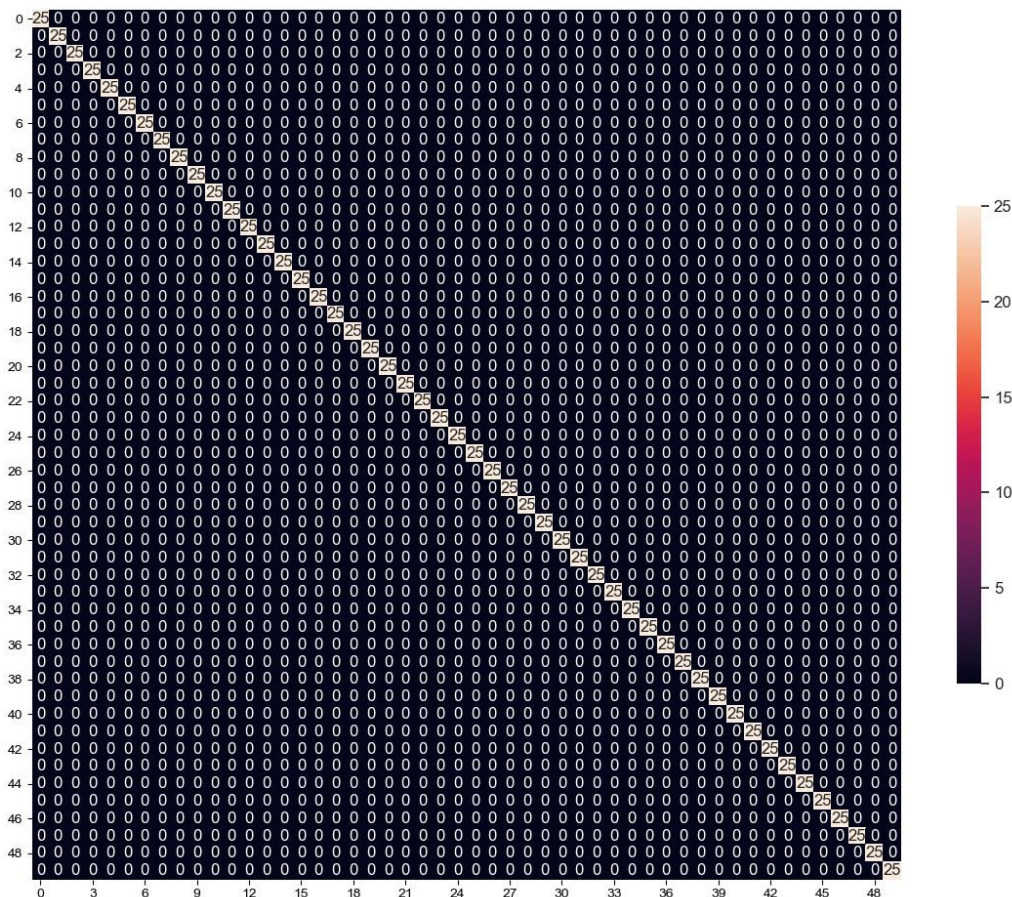
Kao što se može vidjeti sa slike, vjerojatnosti predviđanja nisu uvijek preko 90%. Prikazane su neke slike koje su predviđene s vjerojatnošću oko 86% što je isto poprilično dobro s obzirom da postoji 50 klasa u skupu podataka. U tablici 4.14. prikazan je izvještaj klasifikacije za facenet s 50 klasa, upotrebom SVM klasifikatora, dok je na slici 4.7. prikazana matrica zbunjenosti za facenet s 50 klasa.

**Tab 4.14.** *Izvjestaj klasifikacije za facenet s 50 klasa*

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Al Pacino	1	1	1
Angelina Jolie	1	1	1
Anne Hathaway	1	1	1
Barack Obama	1	1	1
Bill Gates	1	1	1
Brad Pitt	1	1	1
Chris Hemsworth	1	1	1
Chris Pratt	1	1	1
Christian Bale	1	1	1
Colin Farrell	1	1	1
Danny DeVito	1	1	1
David Beckham	1	1	1
Dwayne Johnson	1	1	1
Elon Musk	1	1	1
Emilia Clarke	1	1	1
Francesco Totti	1	1	1
Gabriel Macht	1	1	1
George Clooney	1	1	1
Hugh Jackman	1	1	1
Jackie Chan	1	1	1
Jason Momoa	1	1	1
Jason Statham	1	1	1
Jennifer Lawrence	1	1	1
Jennifer Lopez	1	1	1
Jerry Seinfeld	1	1	1
Justin Timberlake	1	1	1
Keanu Reeves	1	1	1
Kevin Costner	1	1	1
Kevin Spacey	1	1	1
Kit Harington	1	1	1
Magnus Carlsen	1	1	1
Megan Fox	1	1	1



Meghan Markle	1	1	1
Morgan Freeman	1	1	1
Natalie Portman	1	1	1
Nicole Scherzinger	1	1	1
Penelope Cruz	1	1	1
Robert De Niro	1	1	1
Robin Williams	1	1	1
Rowan Atkinson	1	1	1
Ryan Reynolds	1	1	1
Salma Hayek	1	1	1
Shakira	1	1	1
Steve Jobs	1	1	1
Sylvester Stallone	1	1	1
Terry Crews	1	1	1
Tom Cruise	1	1	1
Tom Hanks	1	1	1
Tom Holland	1	1	1
Will Smith	1	1	1
<b>srednja vrijednost</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Ukupan broj slika</b>	<b>1250</b>		
<b>TOČNOST</b>	<b>1</b>		



SI.4.7. Matrica zbunjenosti za facenet s 50 klasa

Kao što se može vidjeti iz izvještaja klasifikacije i matrice zbunjenosti, sve slike su točno klasificirane, što je odličan rezultat. Također, ova točnost od 100% može se djelomično pripisati jednostavnosti skupa podataka i malom broju testnih slika.

#### 4.4. Eksperiment 4

U posljednjem će se eksperimentu upotrebljavati tzv. *one-shot learning*, a to znači da će se uzeti značajke samo jedne slike za svaku klasu te ih testirati sa ostalih 99 slika. Algoritam koji će se rabiti za klasifikaciju će biti već ranije spomenuta metoda potpornih vektora (SVM). Usporediti će se učinkovitost izdvajanja značajki pomoću faceneta i vgg16 modela.

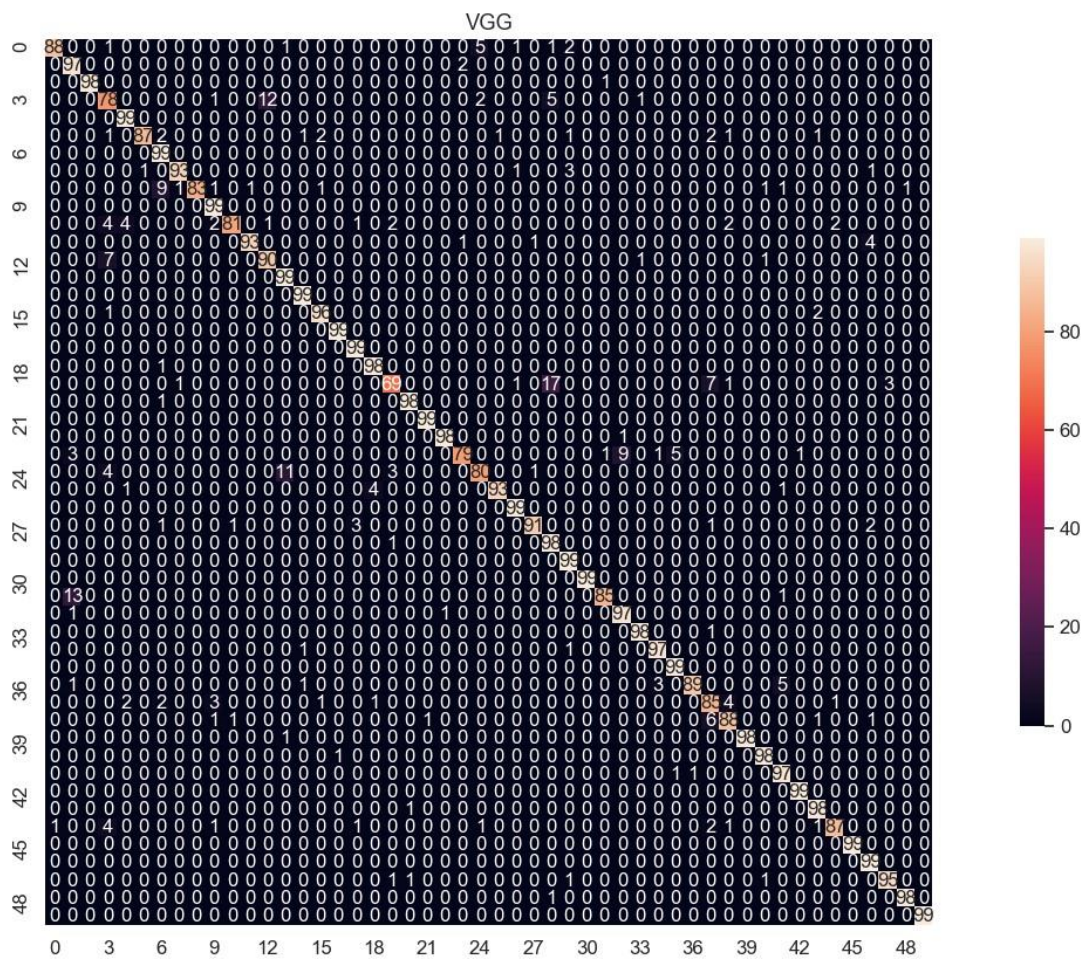
##### 4.4.1. Vgg16

Prvo će se testirati vgg16 model koji ima izlazni vektor značajki veličine 512. U tablici 4.15. prikazan je izvještaj klasifikacije za vgg16 model te je na slici 4.8. prikazana matrica zbunjenosti za vgg16 model.

**Tab 4.15.** Izveštaj klasifikacije za vgg16 s 1 trening slikom

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Al Pacino	0.99	0.89	0.94
Angelina Jolie	0.84	0.98	0.91
Anne Hathaway	1	0.99	0.99
Barack Obama	0.78	0.79	0.78
Bill Gates	0.93	1	0.97
Brad Pitt	0.99	0.88	0.93
Chris Hemsworth	0.86	1	0.93
Chris Pratt	0.98	0.94	0.96
Christian Bale	1	0.84	0.91
Colin Farrell	0.92	1	0.96
Danny DeVito	0.98	0.82	0.89
David Beckham	0.99	0.94	0.96
Dwayne Johnson	0.87	0.91	0.89
Elon Musk	0.88	1	0.94
Emilia Clarke	0.97	1	0.99
Francesco Totti	0.96	0.97	0.96
Gabriel Macht	0.99	1	0.99
George Clooney	0.95	1	0.98
Hugh Jackman	0.95	0.99	0.97
Jackie Chan	0.91	0.7	0.79
Jason Momoa	0.98	0.99	0.98
Jason Statham	0.99	1	0.99
Jennifer Lawrence	0.99	0.99	0.99
Jennifer Lopez	0.96	0.8	0.87
Jerry Seinfeld	0.91	0.81	0.86
Justin Timberlake	0.99	0.94	0.96
Keanu Reeves	0.97	1	0.99
Kevin Costner	0.98	0.92	0.95
Kevin Spacey	0.8	0.99	0.89
Kit Harington	0.93	1	0.96
Magnus Carlsen	1	1	1
Megan Fox	0.98	0.86	0.91
Meghan Markle	0.91	0.98	0.94
Morgan Freeman	0.98	0.99	0.98
Natalie Portman	0.96	0.98	0.97
Nicole Scherzinger	0.94	1	0.97
Penelope Cruz	0.99	0.9	0.94
Robert De Niro	0.82	0.86	0.84
Robin Williams	0.91	0.89	0.9

Rowan Atkinson	1	0.99	0.99
Ryan Reynolds	0.97	0.99	0.98
Salma Hayek	0.92	0.98	0.95
Shakira	0.99	1	0.99
Steve Jobs	0.95	0.99	0.97
Sylvester Stallone	0.97	0.88	0.92
Terry Crews	1	1	1
Tom Cruise	0.93	1	0.96
Tom Hanks	0.97	0.96	0.96
Tom Holland	0.99	0.99	0.99
Will Smith	1	1	1
<b>srednja vrijednost</b>	<b>0.95</b>	<b>0.95</b>	<b>0.95</b>
<b>Ukupan broj slika</b>	<b>4950</b>		
<b>TOČNOST</b>	<b>0.94586</b>		



**SI.4.8.** Matrica zbunjenosti za vgg16 s 1 trening slikom

Kao što se može vidjeti iz izvještaja, naučeni vgg16 model ima točnost od 94,586% što je izvrsno s obzirom na to da je SVM klasifikator imao samo jednu sliku kao trening podatak. Iz matrice zbunjenosti vidljivo je da je u ovom slučaju nekoliko osoba lažno pozitivno označene kao druge, no nema niti jedne klase koja je označena ispod 70%, što je odličan rezultat.

#### 4.4.2. Facenet

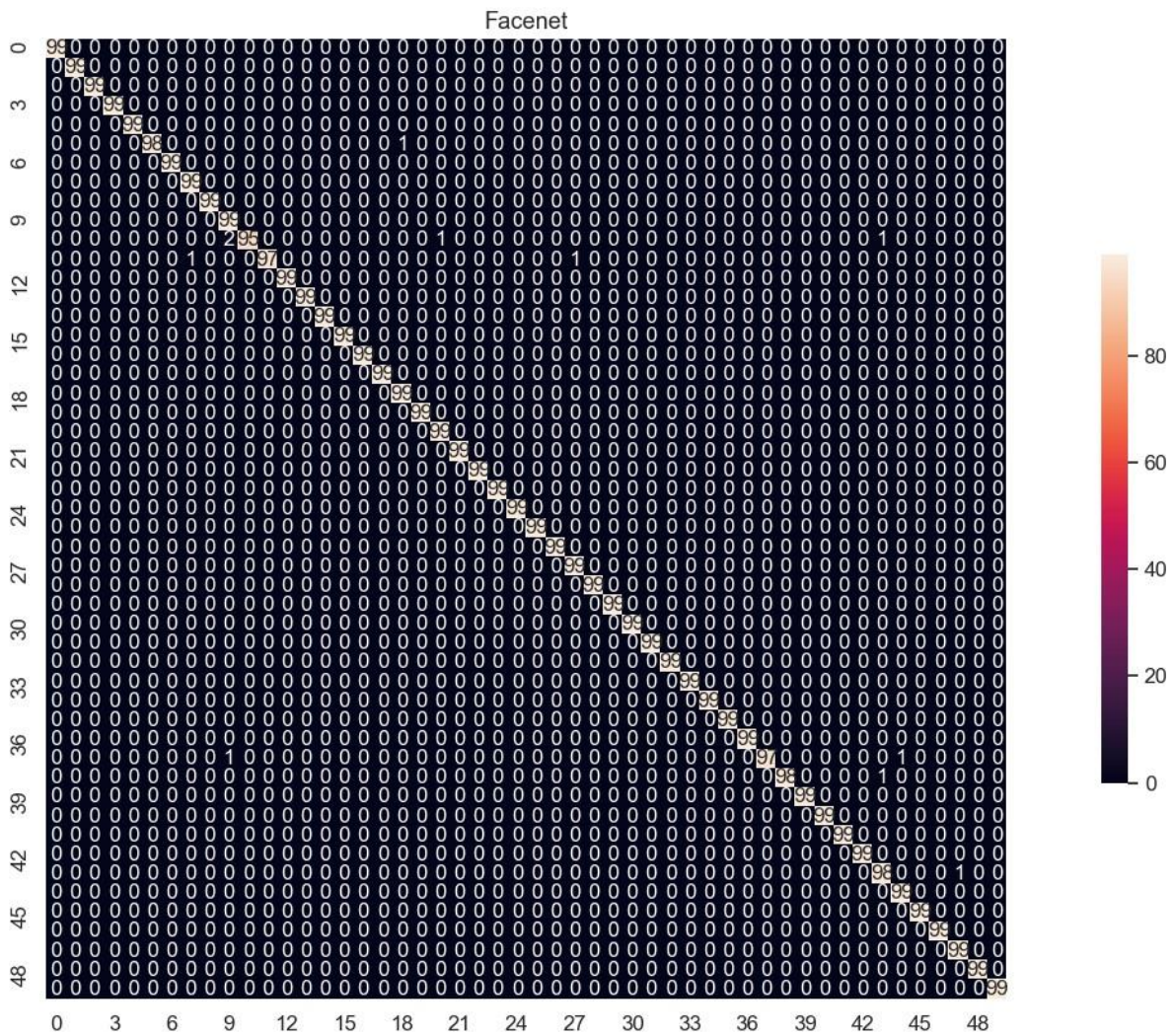
Facenet izlazni vektor značajki ima veličinu 128 za razliku od vgg16 modela koji ima veličinu 512. U tablici 4.16. vidljiv je izvještaj klasifikacije za facenet model upotrebljavajući jednu trening sliku. Na slici 4.9. prikazana je matrica zbunjenosti za facenet.

**Tab 4.16.** *Izvještaj klasifikacije za facenet s 1 trening slikom*

KLASA	PRECIZNOST	ODZIV	F1-MJERA
Al Pacino	1	1	1
Angelina Jolie	1	1	1
Anne Hathaway	1	1	1
Barack Obama	1	1	1
Bill Gates	1	1	1
Brad Pitt	1	0.99	0.99
Chris Hemsworth	1	1	1
Chris Pratt	0.99	1	0.99
Christian Bale	1	1	1
Colin Farrell	0.97	1	0.99
Danny DeVito	1	0.96	0.98
David Beckham	1	0.98	0.99
Dwayne Johnson	1	1	1
Elon Musk	1	1	1
Emilia Clarke	1	1	1
Francesco Totti	1	1	1
Gabriel Macht	1	1	1
George Clooney	1	1	1
Hugh Jackman	0.99	1	0.99
Jackie Chan	1	1	1
Jason Momoa	0.99	1	0.99
Jason Statham	1	1	1
Jennifer Lawrence	1	1	1
Jennifer Lopez	1	1	1
Jerry Seinfeld	1	1	1



Justin Timberlake	1	1	1
Keanu Reeves	1	1	1
Kevin Costner	0.99	1	0.99
Kevin Spacey	1	1	1
Kit Harington	1	1	1
Magnus Carlsen	1	1	1
Megan Fox	1	1	1
Meghan Markle	1	1	1
Morgan Freeman	1	1	1
Natalie Portman	1	1	1
Nicole Scherzinger	1	1	1
Penelope Cruz	1	1	1
Robert De Niro	1	0.98	0.99
Robin Williams	1	0.99	0.99
Rowan Atkinson	1	1	1
Ryan Reynolds	1	1	1
Salma Hayek	1	1	1
Shakira	1	1	1
Steve Jobs	0.98	0.99	0.98
Sylvester Stallone	0.99	1	0.99
Terry Crews	1	1	1
Tom Cruise	1	1	1
Tom Hanks	0.99	1	0.99
Tom Holland	1	1	1
Will Smith	1	1	1
<b>srednja vrijednost</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>Ukupan broj slika</b>			<b>4950</b>
<b>TOČNOST</b>			<b>0.99778</b>



**Sl.4.9.** Matrica zbunjenosti za facenet s 1 slikom za trening

Kao što se može vidjeti iz izvještaja, facenet model uz SVM klasifikator ima točnost od nevjerovatnih 99,778% što je izvanredno. Iz matrice zbunjenosti se vidi da su sve klase klasificirane točno u preko 95% slučajeva.

## 5. ZAKLJUČAK

U ovom radu testirani su algoritmi prepoznavanja lica. Za treniranje i testiranje klasifikatora bilo je potrebno prikupiti skup podataka te ga razdvojiti na trening i test skup. Prikupljeni skup podataka bilo je potrebno prilagoditi za prepoznavanje lica, što znači detektirati, izrezati, poravnati i centrirati lica te spremi za upotrebu u algoritmima prepoznavanja lica. Za izdvajanje značajki i smanjenje dimenzionalnosti u prvom testu upotrebljavala se analiza glavnih komponenta (PCA), dok je za drugi test upotrebljen facenet model za izdvajanje značajki. U prvom i drugom testu bilo je 6 klasa. Trening skup je imao 75 slika, dok je testni skup imao 25 slika. Zaključeno je da su logistička regresija, analiza linearnih diskriminanti (LDA) i metoda potpornih vektora (SVM) bolji za klasifikaciju u prepoznavanju lica od algoritma k najbližih susjeda. Za facenet model klasifikator nije bio važan i rezultat je bio savršen jer su značajke lica bile vrlo dobro izdvojene te je broj klasa bio manji. U trećem testu broj klasa je povećan na 50 te je facenet model testiran s istim brojem trening slika i testnih slika. Rezultat je također bio 100%. Za posljednji test rabljena je metoda *one-shot learning*, gdje se SVM klasifikator testirao s dvjema naučenim mrežama, facenet i vgg16. Vgg16 model imao je točnost od odličnih 94,6%, dok je facenet model imao točnost od nevjerojatnih 99,8%. Točnosti ovih modela može se pripisati velikoj količini slika lica na kojima su bili trenirani. I facenet i vgg16 modeli su naučeni na preko 3 milijuna slika te nije čudno što su toliko dobri u izdvajanju najvažnijih značajki pojedinog lica.



## LITERATURA

- [1] Bledsoe W. W., Report on a Large-Scale Experiment, Technical Report PRI 22, Panoramic Research, Inc., 1966.
- [2] Goldstein A. J., Harmon L. D., Lesk A. B., Identification of human faces. In *Proc. IEEE*, vol. 59, p.748, 1971.
- [3] Sirovich L. and Kirby M., Low-Dimensional Procedure for the Characterization of Human Faces, *Journal of Optical Soc. of Am.*, vol. 4, pp. 519-524, 1987.
- [4] Turk M. and Pentland A., Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
- [5] Povijest prepoznavanja lica, <https://readwrite.com/2020/03/12/history-of-facial-recognition-technology-and-its-bright-future/> (pristupljeno: travanj 2020.)
- [6] facenet: Face recognition using Tensorflow, <https://github.com/davidsandberg/facenet> (pristupljeno: siječanj 2020.)
- [7] keras-vggface biblioteka, <https://github.com/rcmalli/keras-vggface> (pristupljeno: svibanj 2020.)
- [8] M. Vranješ, R. Grbić, Predavanja iz kolegija Strojno učenje u sustavima autonomnih i umreženih vozila, studij Automobilsko računarstvo i komunikacije, ak.god. 2018/2019., [https://loomen.carnet.hr/pluginfile.php/1600840/mod\\_resource/content/3/01%20Uvod%20u%20strojno%20u%C4%8Denje.pdf](https://loomen.carnet.hr/pluginfile.php/1600840/mod_resource/content/3/01%20Uvod%20u%20strojno%20u%C4%8Denje.pdf) (pristupljeno: travanj 2020.)
- [9] K. Kyungnam, Face Recognition using Principle Component Analysis, <http://staff.ustc.edu.cn/~zwp/teach/MVA/pcaface.pdf> (pristupljeno: svibanj 2020.)
- [10] A. Pant, Introduction to logistic regression, <https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148> (pristupljeno: svibanj 2020.)
- [11] Sigmoid function, Wikipedia, [https://en.wikipedia.org/wiki/Sigmoid\\_function](https://en.wikipedia.org/wiki/Sigmoid_function) (pristupljeno: svibanj 2020.)

- [12] R. Gotesman, Mathematical insights into classification using linear discriminant analysis, <https://towardsdatascience.com/mathematical-insights-into-classification-using-linear-discriminant-analysis-9c822ad2fce2> (pristupljeno: svibanj 2020.)
- [13] Analiza linearnih diskriminanti, Wikipedia, [https://en.wikipedia.org/wiki/Linear\\_discriminant\\_analysis](https://en.wikipedia.org/wiki/Linear_discriminant_analysis) (pristupljeno: lipanj 2020.)
- [14] R. Gandhi, Support Vector Machine – Introduction to Machine Learning Algorithms, <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47> (pristupljeno: svibanj 2020.)
- [15] A. Navlani, KNN Classification using Scikit-learn, <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn> (pristupljeno: svibanj 2020.)
- [16] S. Saha, A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (pristupljeno: svibanj 2020.)
- [17] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, <https://arxiv.org/abs/1503.03832> (pristupljeno: veljača 2020.)
- [18] `search_bing_api.py` skripta za preuzimanje slika, [https://github.com/aekruijssen/Facial-Recognition/blob/master/search\\_bing\\_api.py](https://github.com/aekruijssen/Facial-Recognition/blob/master/search_bing_api.py) (pristupljeno: siječanj 2020.)
- [19] U. Soni, Removing Duplicate Images through Python, <https://medium.com/@urvisoni/removing-duplicate-images-through-python-23c5fdc7479e> (pristupljeno: veljača 2020.)
- [20] P. Viola, M. Jones, „*Rapid Object Detection using a Boosted Cascade of Simple Features*“, Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Sjedinjene Američke Države, 2001.

- [21] S. Rimac-Drlje, M. Vranješ, R. Grbić, predavanja iz kolegija Digitalna obrada slike i videa za autonomna vozila, studij Automobilsko računarstvo i komunikacije, ak.god. 2018/2019., <https://loomen.carnet.hr/course/view.php?id=9844> (pristupljeno: siječanj 2020.)
- [22] Cascade Classifier, [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html) (pristupljeno: travanj 2020.)
- [23] Haarcascades, <https://github.com/opencv/opencv/tree/master/data/haarcascades> (pristupljeno: veljača 2020.)
- [24] Tensorface: `shape` predictor, [https://github.com/AKSHAYUBHAT/TensorFace/blob/master/openface/models/dlib/shape\\_predictor\\_68\\_face\\_landmarks.dat](https://github.com/AKSHAYUBHAT/TensorFace/blob/master/openface/models/dlib/shape_predictor_68_face_landmarks.dat) (pristupljeno: siječanj 2020.)
- [25] M. Krishnan, Understanding the Classification report through sklearn, <https://muthu.co/understanding-the-classification-report-in-sklearn/> (pristupljeno: travanj 2020.)

## SAŽETAK

U ovom radu ispitana su četiri algoritma strojnog učenja koja su se upotrebljavala za klasifikaciju. Prvotno je pohranjen skup podataka koji je zatim obrađen na način da su detektirana lica, izrezana, poravnata te pohranjena u podatak koji se kasnije učitava radi lakšeg izvršavanja eksperimenata. U prvom eksperimentu se za izdvajanje značajki i smanjenje dimenzionalnosti upotrebljavala analiza glavnih komponentata (PCA), dok je za drugi eksperiment upotrebljen facenet model za izdvajanje značajki. Zaključeno je da je facenet superiorniji u izdvajanju značajki od PCA. U trećem eksperimentu je također rabljen facenet model, ali s povećanim brojem klasa u odnosu na prvi i drugi eksperiment. U posljednjem, četvrtom eksperimentu, uspoređena je vgg16 naučena neuronska mreža s facenet naučenim modelom. Vgg16 model imao je točnost od 94,6%, dok je facenet model imao točnost od nevjerovatnih 99,8%. Zaključeno je da su neuronske mreže daleko učinkovitije u izdvajanju značajki te sam klasifikacijski algoritam ne igra veliku ulogu u točnoj klasifikaciji, ukoliko su značajke dobro izdvojene.

**Ključne riječi:** algoritmi strojnog učenja za prepoznavanje lica, analiza glavnih komponentata, K najbližih susjeda, analiza linearnih diskriminanti, logistička regresija, metoda potpornih vektora, facenet, vgg16

## **ABSTRACT**

### **MACHINE LEARNING ALGORITHMS IN FACE RECOGNITION**

In this paper, four machine learning algorithms have been tested for classification. Initially, the dataset was processed in such a way that the detected faces were cut, aligned and stored in a file for later use. In the first experiment, principal component analysis was used for feature extraction and dimensionality reduction while for the second experiment, the pretrained facenet model was used. From the experiments, it was concluded that facenet is superior to PCA in feature extraction. In the third experiment, facenet model was used, but with a greater number of classes than in the first two experiments. In the last, fourth experiment, the vgg16 pretrained neural network was compared to the facenet pretrained model. Vgg16 model had an accuracy of 94,6% while the facenet model had an amazing accuracy of 99,8%. In conclusion, neural networks are far more efficient in feature extraction and the classification algorithm itself doesn't play a big role in correctly classifying the face, if the features are well extracted.

**Key words:** machine learning algorithms in face recognition, principal component analysis, K nearest neighbors, linear discriminant analysis, logistic regression, support vectore machine, facenet, vgg16

## **ŽIVOTOPIS**

Benjamin Puljko rođen je 30.09.1995. godine u Osijeku. U Veliškovcima završava osnovnu školu Ivana Kukuljevića i 2010. upisuje opću gimnaziju u Valpovu. Nakon gimnazije upisuje preddiplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Nakon završenog preddiplomskog studija 2017. godine upisuje diplomski studij – Automobilsko računarstvo i komunikacije.

Potpis:

---

## **PRILOZI**

- Programski kod
- Skup podataka