

**Projekt iz predmeta Raspoznavanje uzoraka**

**Ak. god. 2012/13**

# **Detekcija i „meko“ raspoznavanje na temelju siluete osoba**

Autori:

Igor Jerković

Petar Alilović

Matija Bagić

Tena Dominko

Daniel Ćuk

Viktor Skolan

Martin Šošić

Matija Šošić

**siječanj, 2013.**

# Sadržaj:

<b>1. Projektni zadatak .....</b>	<b>1</b>
1.1 OPIS PROJEKTOG ZADATAKA .....	1
1.2 PREGLED I OPIS SRODNIH RJEŠENJA .....	1
1.2.1 Granlundov sustav za raspoznavanje slova .....	4
1.3 KONCEPTUALNO RJEŠENJE ZADATAKA .....	6
<b>2. Postupak rješavanja zadatka .....</b>	<b>8</b>
2.1 DOBIVANJE CRNO-BIJELE SILUETE.....	8
2.1.1 Oduzimanje pozadine.....	9
2.1.1.1 Kalibriranje slike.....	9
2.1.1.2 Zaglađivanje slika .....	10
2.1.1.3 Računanje razlike slika.....	10
2.1.1.4 Binarizacija slike .....	10
2.1.1.5 Pročišćavanje siluete .....	11
2.2 GRANLUNDOVI KOEFICIJENTI .....	12
2.2.1 Računanje Fourierovih koeficijenata .....	13
2.2.2 Računanje Granlundovih koeficijenata .....	14
2.3 KLASIFICIRANJE NA OSNOVU PROPORCIJA TIJELA.....	14
2.4 TEMPLATE MACHING .....	16
2.4.1 Template matching s brojem jednakih piksela .....	16
2.4.2 Template matching s PCA .....	17
<b>3. Ispitivanje rješenja.....</b>	<b>20</b>
3.1 ISPITNA BAZA .....	20
3.2 REZULTATI UČENJA I ISPITIVANJA .....	20
3.2.1 Rezultati - Granlundovi koeficijenti.....	20
3.2.2 Rezultati - Klasificiranje na osnovu proporcije tijela.....	22
3.2.3 Rezultati - Template matching.....	24
3.2.4 Rezultati - Template matching s PCA.....	25
3.2.5 Usporedba rezultata .....	26
3.3 ANALIZA REZULTATA.....	26
3.3.1 Granlundovi koeficijenti .....	26
3.3.2 Klasificiranje na osnovu proporcija tijela .....	27
3.3.3 Template matching s PCA .....	27
3.3.4 Template matching.....	28
<b>4. Opis programske implementacije rješenja.....</b>	<b>29</b>

4.1	OPIS KORIŠTENJA .....	30
4.2	OPISI IMPLEMENTACIJA POJEDINIH ALGORITAMA .....	31
<b>5.</b>	<b>Zaključak .....</b>	<b>32</b>
<b>6.</b>	<b>Literatura .....</b>	<b>34</b>

# 1. Projektni zadatak

## 1.1 Opis projektnog zadatka

Prvi dio projektnog zadatka je detekcija osobe i izlučivanje njezine siluete iz slike. Detekcija osobe je dosta zahtjevan zadatak s obzirom da je ulazni podatak statična slika osobe, a ne video sekvenca kretanja osobe. Dodatno, osoba može biti tako odjevena da bude više ili manje uklopljena u pozadinu, što naravno otežava proces same detekcije. Različite mogućnosti izgleda pozadine zahtijevaju dovoljno dobre načine obradbe ulazne slike radi što boljeg dohvaćanja siluete osobe. Silueta osobe je na slici prezentirana skupom bijelih piksela, dok je pozadina crne boje. Silueta odražava figuru osobe i njezine posebnosti vezane uz građu tijela i frizure (posebno se očituje ako osoba ima dugu raspuštenu kosu). Sva ostala svojstva su izgubljena. Upravo zbog te činjenice da se barata sa siluetama koje su binarne (crno - bijele) slike potrebno je dosta eksperimentiranja oko pronalaska algoritma koji će uspjeti obuhvatiti najbolje značajke iz silueta.

Nakon što je dohvaćena silueta osobe prelazi se na fazu raspoznavanja. „Meko“ raspoznavanje odnosi se na pojam nepotpunog, odnosno nepreciznog raspoznavanja. Nepreciznost raspoznavanja nastaje zbog vrlo malo značajki koje su posebne samo za tu osobu, kao što ih ima na primjer kod otiska dlana, slike lica, otiska prstiju i ostalih biometrijskih značajki, koje jedinstveno određuju tu osobu. Kao što je i ljudskom oku vrlo teško raspoznati osobu na ulici kad vidi samo njenu sjenu, tako je i računalu vrlo teško odrediti kojoj osobi pripada ta silueta. Iz navedenih razloga treba problemu pristupiti sa stajališta da je rezultat identifikacije osobe s obzirom na njenu siluetu vrlo zahtjevan problem kojem treba pristupiti iz više aspekata i na više načina.

Što se samog izbora algoritma tiče tu nema nikakvog ograničenja što je u slučaju ovog projekta povoljna situacija jer omogućava isprobavanje različitih metoda i razvoj drugačijih algoritama koji će možda unaprijediti neke postojeće algoritme ili, još bolje, implementirat će se novi algoritmi koji će s obzirom na ulazne slike silueta osoba izlučiti najvažnije značajke koje će omogućiti što bolje raspoznavanje i identifikaciju osobe.

## 1.2 Pregled i opis srodnih rješenja

Većina srodne literature, odnosno članaka i radova na temu prepoznavanja na temelju siluete ne bavi se slikama, to jest statičnim siluetama već videom, odnosno klasifikacijom

objekta u pokretu. Ekstrakcija i analiza siluete tu se uglavnom koristi na isti način kao i u našem projektu, samo kroz veći niz povezanih sličica koji su dio videozapisa.

Promotrimo „*Gait Curves for Human Recognition, Backpack Detection and Silhouette Correction in a Nighttime Environment*“ (DeCann, Ross). Tema članka je razvoj sustava noćnog video nadzora. Sustav prepoznaje humanoidne objekte u pokretu, prati ih te razlikuje. Klasificira ih na temelju hoda (*gait*), za što koristi izlučivanje silueta i poklapanje istih tijekom vremenskog razdoblja, čime razlučuje objekt na statične i dinamičke dijelove. Statični dijelovi predstavljaju centralni dio objekta koji ostaje na istom mjestu kako se objekt kreće kroz videozapis, dok su dinamički dijelovi oni koji osciliraju stalnim periodom gibanja. Pod dinamičke dijelove čovjeka u pokretu tako spadaju ruke, noge, te rubni dijelovi tijela. Sustav također razlikuje strane objekte vezane uz čovjeka (ruksak, torba i slično) prema različitom periodu i smjeru periodičnog gibanja naspram ostatka tijela.

Za ekstrakciju korisnih informacija iz siluete, sustav koristi vrlo jednostavne metode, mjeri visinu siluete kao razliku najvišeg i najnižeg piksela. Pošto visina siluete u videozapisu, zbog različite udaljenosti i položaja objekta naspram kamere, nije uvijek relevantna, mjeri se i relativna visina, kao najveća udaljenost dvaju piksela ekstrahirane siluete.

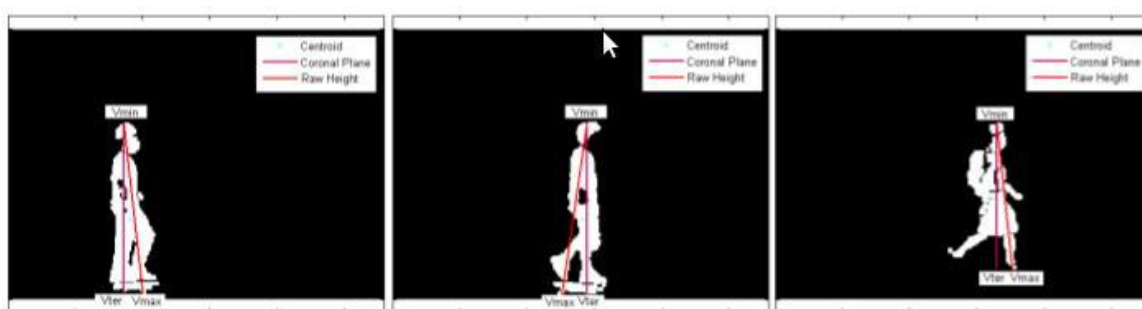


Figure 3. Labeled silhouettes. Note the difference in marking the coronal plane and centroid.

[Slika 1: Slika iz članka \[2\] - označene siluete](#)

Nakon što se u nizu sličica videozapisa analiziraju sve siluete, tako dobivene slike se spoje u jednu, pa sustav gleda oscilacije u svrhu analiziranja hoda i izvlačenja značajki hoda u obliku perioda titranja.

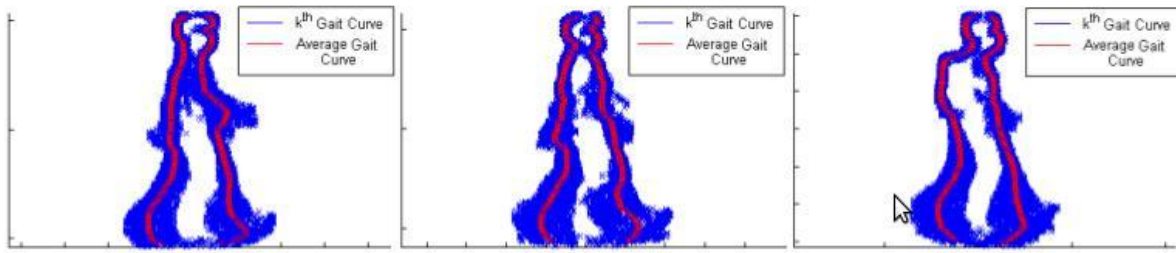


Figure 4. Gait curves for three different subjects. The mean gait curve is included.

Slika 2: Slika iz članka [2] - oscilacije u hodu

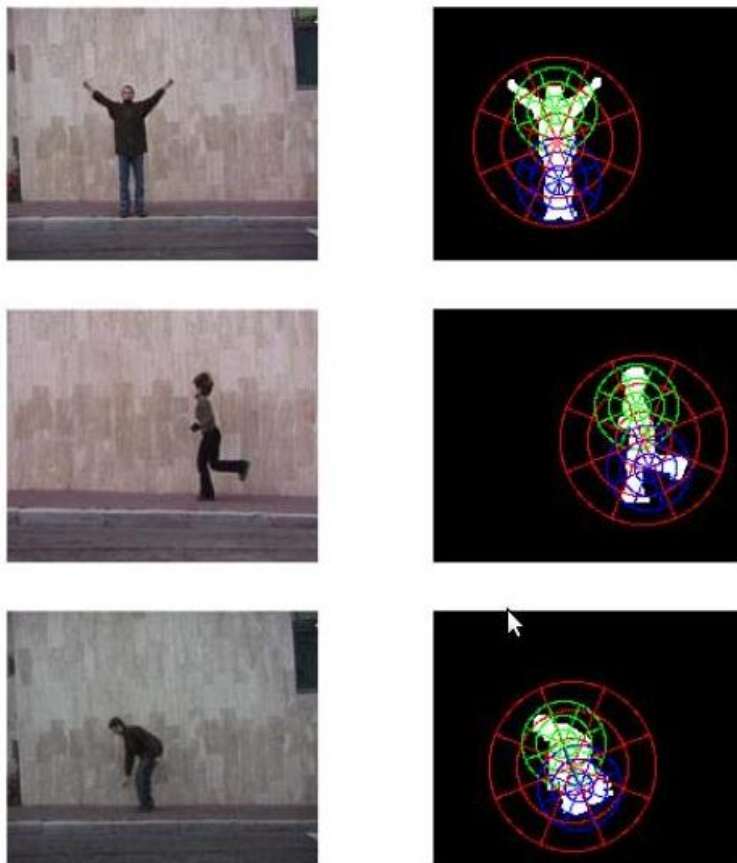
Sljedeći članak kojeg razmatramo je „*Silhouette recognition using high-resolution pursuit*“ (Jaggi, Karl, Mallat, Willsky). Ovaj rad uvodi novi postupak prepoznavanja objekata na slici koristeći siluete nazvan „*High-resolution pursuit*“ (HRP), prevedeno "potraga visoke razlučivosti". Na primjeru prepoznavanja silueta 17 različitih modela vojnih zrakoplova, razvijen je postupak HRP koji, unatoč izuzetnoj jednostavnosti algoritma koji koristi samo osnovne značajke objekta, daje dobre rezultate prepoznavanja.

U prvom koraku ekstrahira se CDP (*Centroid Distance Profile*), kao jednodimenzionalni prikaz siluete objekta koji se računa kao udaljenost točaka rubova objekta od centroida istog. Formula je prikazana na slici, gdje su  $x(n)$  i  $y(n)$  koordinate točke ruba, a  $X_c$  i  $Y_c$  koordinate centroida. Tako dobiven dijagram ne mijenja se u ovisnosti o orijentaciji objekta, jer bilo kakva rotacija samo cirkularno pomiče dijagram, ne mijenjajući njegove točke.

$$f(n) = \sqrt{(x(n) - X_c)^2 + (y(n) - Y_c)^2}$$

HRP također koristi i podatke o kutovima unutar objekta, koje CDP ne sadrži, odnosno izbacuje, da bi kombinacijom CDP-a i podataka o kutovima dobio brz i robustan način prepoznavanja objekata.

Treći i posljednji izabrani srodni članak jest „*Human Action Recognition Using Silhouette Histogram*“, (Hsieh, Huang, Tang). Rad se bavi prepoznavanjem i klasificiranjem ljudskih radnji analizom siluete kroz tri različita sustava polarnih koordinati koji predstavljaju tri dijela ljudske figure – C1, C2 i C3, gdje C1 predstavlja kružnicu koja zaokružuje cijelo tijelo, C2 samo gornji a C3 samo donji dio ljudskoga tijela. Svaki od tih koordinatnih sustava ima različite središnje točke, gdje se za C1 koristi centroid objekta (ekstrahirane siluete), a za C2 i C3 pomaci po osi objekta za pola radijusa C1 prema gore ili dolje.



Slika 3: Slika iz članka [4] - siluete ljudi u tri polarna koordinatna sustava

Svaki od navedena tri polarna sustava računa se iz početnog Kartezijevog sustava u domeni siluete, a koriste se za izradu histograma, nakon što se polarni sustavi podijele na sektore, kao što je vidljivo iz priložene slike. Zatim se, radi klasifikacije ljudskih radnji, analizira osciliranje unutar histograma, koji daje idealno specifične rezultate u odnosu na specifične ljudske radnje.

Iako na području raspoznavanja silueta nema relevantnih rješenja ni članaka, na području raspoznavanja oblika postoje neka. Jedno od takvih rješenja opisano je u Granlundovom članku [1], gdje je potonji autor napravio sustav te predložio nadogradnje istog u svrhu raspoznavanja rukom pisanih velikih tiskanih slova.

### 1.2.1 Granlundov sustav za raspoznavanje slova

Baza koja se koristila tijekom provođenja eksperimenta sadržavala je 175 uzorka, za svako slovo u bazi, od njih 25, postojalo je po 7 rukom napisanih uzoraka. Pošto navedeni pokus datira iz 1972. godine, iz ere kada su još u uporabi bile bušene kartice te je izrada većeg skupa uzoraka bila vrlo vremenski zahtjevna, autor je koristio isti skup i za učenje i za testiranje.



Slika 4: Slika iz članka [1] - Oblici slova korišteni za testiranje

Navedena činjenica rezultirala je vrlo optimističnom procjenom uspješnosti klasificiranja koja u praksi ne bi bila takva. Za svako slovo na bušenoj kartici postupak se provodio jednako u nekoliko koraka:

- bušena kartica sa matricom dimenzije 35 \* 35 učitava se u računalo
- računalo procesira bušenu karticu te izračunava konturu slova
- potom se računaju Fourierovi koeficijenti
- u zadnjem koraku računaju se Granlundovi koeficijenti, po njih sedam za svako slovo ( $d_{11}, d_{22}, d_{21}, d_{12}, d_{31}, d_{13}, d_{44}$ )

Kako sam autor navodi, nije poznato da navedeni koeficijenti predstavljaju najbolje značajke za uzorke nego su izabrani nakon jednostavnog kvalitativnog razmatranja. Nakon učitavanja cijelog skupa za učenje te izračunavanja koeficijenata, idući korak je priprema klasifikatora. Odabrani klasifikator u članku je probabilističke prirode, a klasificira na temelju najmanje udaljenosti. Pretpostavka koja je korištena u članku pretpostavlja da se vrijednosti Granlundovih koeficijenata jednog slova mogu predstaviti dvodimenzionalnom Gaussovom distribucijom

$$f_{d_{mn}}(x, y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2}Q(x, y)\right]$$

gdje



$$Q(x, y) = \frac{1}{1 - \rho^2} \left[ \frac{(x - x_0)^2}{\sigma_1^2} - \frac{2\rho(x - x_0)(y - y_0)}{\sigma_1\sigma_2} + \frac{(y - y_0)^2}{\sigma_2^2} \right].$$

U prikazanoj formuli  $x$  i  $y$  predstavljaju realni imaginarni dio koeficijenta,  $x_0$  i  $y_0$  su srednje vrijednosti dobivene na cijelom skupu za učenje;  $\sigma_1$  i  $\sigma_2$  su varijance varijabli na skupu za učenje, a  $\rho$  je korelacijski koeficijent. Nadalje, u fazi testiranja, računali su se koeficijenti sada nepoznatih slova. Uz pomoć formule računale su se udaljenosti između tih izračunatih vrijednosti i srednje vrijednosti dobivene ranije.

$$f_{d_{mn}}(z) = \frac{1}{\sigma_z \sqrt{2\pi}} \exp \left[ -\frac{1}{2\sigma_z^2} z^2 \right]$$

Uz pretpostavku da su različiti  $d_{mn}$  koeficijenti nezavisni, računa se MLE (Maximum Likelihood Estimator) za svaku klasu. Uzorak se klasificira u najizgledniju klasu, tj. klasu sa najvećom vrijednošću.

$$P(c) = \sum_{m,n} \left[ -\frac{1}{2\sigma_z^2} z^2 \right] - \ln \sigma_{z_{mn}}$$

Rezultati pokusa, kao što je navedeno, vrlo su optimistični. Od 175 slova u skupu, u fazi testiranja samo njih četiri je bilo pogrešno klasificirano.

### 1.3 Konceptualno rješenje zadatka

Za rješavanje zadatka koristi se baza slika koja obuhvaća barem 5 slika svake od 24 osoba u bazi. Prvi korak je detekcija osobe na slici, to jest odvajanje pozadine. Algoritam prima dvije slike (sliku pozadine i sliku iste te pozadine sa osobom na slici) te stvara novu binarnu sliku na kojoj pikseli s vrijednošću 0 predstavljaju pozadinu (crno područje), a pikseli s vrijednošću 1 osobu (bijelo područje). Dobivene slike spremaju se u bazu i koriste u daljnjim koracima.

Sljedeći koraci su izlučivanje i odabir značajki za što se koristi više različitih tehnika. Jedna od njih je računanje Granlundovih koeficijenata. Iz binarne slike dobivene u prethodnom koraku izdvoji se kontura koja predstavlja osobu na slici. Za dobivenu konturu računaju se Granlundovi koeficijenti koji predstavljaju vektor značajki. Značajke se dalje proslijeđuju na dva načina: kao mapa svih vektora značajki koji se sastoje od izračunatih koeficijenata i kao matrica nastala korištenjem algoritma PCA na mapu vektora značajki.

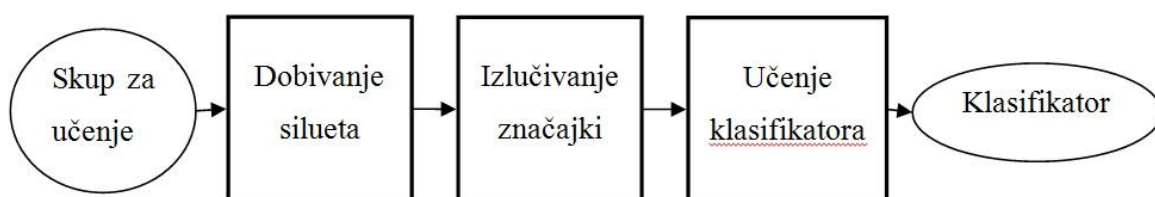
Jedna od obećavajućih metoda pri mekom raspoznavanju je klasificiranje na temelju proporcija tijela, odnosno siluete. Glavne značajke koje se mogu izlučiti iz silueta su svakako visina i širina siluete. Jedna od najvažnijih značajki je visina jer je njome vrlo lako razlikovati

dvije siluete, već i za relativno male razlike značajki. No osim same visine i širine siluete, izlučujemo još i visinu glave, širinu glave, te širinu ramena. Naš sustav radi neovisno o udaljenosti kamere od osobe pri snimanju, zato jer se sve vrijednosti značajki mogu podijeliti s visinom, te tako se barata s omjerima koji su neovisni o udaljenosti kamere ili fotoaparata, što je odlično svojstvo sustava jer se tako ne mora uvijek slikati sa iste udaljenosti kako bismo imali iste rezultate.

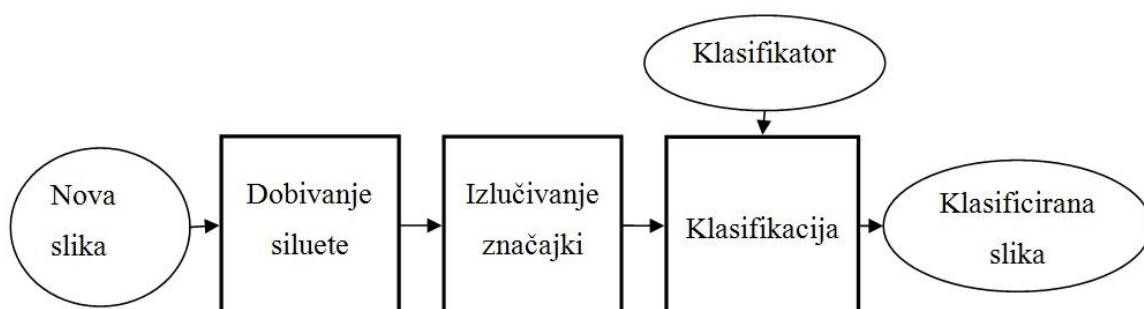
Template matching centrira sve binarne slike silueta te tada traži područje interesa. U ovom problemu je to pravokutnik najmanjih dimenzija u kojeg stane cijela silueta. Tada radi uspoređivanje obrađenom testnom slikom po vrijednostima piksela (broji jednake piksele dviju slika).

Nakon izlučivanja značajki slijedi učenje klasifikatora. Na metodama za koje je to moguće smo isprobali razne klasifikatore (kNN, Random Forest, SVM, Bayes).

Dobiveni klasifikator zatim testiramo na skupu za testiranje kako bismo pronašli najbolji klasifikator.



Slika 5: Učenje klasifikatora



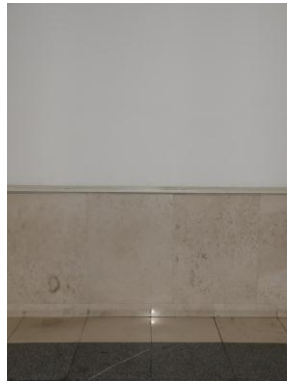
Slika 6: Klasifikacija nove slike

## 2. Postupak rješavanja zadatka

### 2.1 Dobivanje crno-bijele siluete

- **Ulazi:**

- Slika pozadine (dalje u tekstu „slika B“)
- Slika iste te pozadine sa osobom na slici (dalje u tekstu slika „F“)



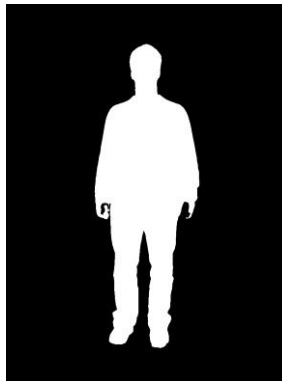
Slika 7: Slika B



Slika 8: Slika F

- **Izlaz:**

- Crno-bijela binarna slika na kojoj je silueta osobe bijele boje a pozadina je crne boje



Slika 9: Dobivena silueta

U ovom koraku iz slike na kojoj se nalazi osoba dobivamo njenu siluetu. Pri tome radi jednostavnosti pretpostavljamo da je sustav statičan, to jest da nam je poznata pozadina. Kada tako pojednostavimo sustav, siluetu osobe možemo dobiti oduzimanjem pozadine (engl. *background subtraction*).

### 2.1.1 Oduzimanje pozadine

Osnovna ideja je vrlo jednostavna: ako imamo sliku pozadine te imamo sliku nekog objekta ispred te pozadine, onda taj objekt možemo izdvojiti sa slike tako da izdvojimo sve piksele u kojima se slike razlikuju. To se uobičajeno radi oduzimanjem slike pozadine od slike s objektom od kuda je i došao naziv postupka.

Nažalost ideja je jednostavna samo dok se radi o idealnim uvjetima. U realnim uvjetima pikseli pozadine na dvije slike neće nikada biti sasvim identični zbog više razloga kao što su sjena koju baca objekt, različito osvjetljenje, mali pomak kamere/fotoaparata.

Kako bismo riješili taj problem, modificirat ćemo postupak tako da postane neosjetljiv na šum, manje promjene u osvjetljenju te mali pomak kamere/fotoaparata.

#### 2.1.1.1 Kalibriranje slike

Kako bismo riješili problem pomaka kamere/fotoaparata uvesti ćemo dodatni korak koji zovemo kalibriranje slike.

Pretpostavimo da je slika F naspram slike B pomaknuta za određeni pomak  $(dx, dy)$  gdje su  $dx$  i  $dy$  cijeli brojevi. Ako nam je taj pomak poznat onda možemo sliku F translirati za  $(-dx, -dy)$  čime dobivamo kalibriranu sliku F s obzirom na sliku B. Naš postupak tako postaje neosjetljiv na pomake kamere/fotoaparata.

Problem koji preostaje riješiti je pronalazak iznosa pomaka  $(dx, dy)$ . Unatoč mogućim razlikama u osvjetljenju među slikama i šumu očekujemo da će razlika između dvije slike biti minimalna kada su slike točno kalibrirane tj. kada sliku F transliramo upravo za  $(-dx, -dy)$ . Sada smo pronalaženje pomaka  $(dx, dy)$  sveli na pronalazak argumenata  $(dx, dy)$  koji minimiziraju razliku između translirane slike F i slike B. Pronalazak najboljih  $(dx, dy)$  smo realizirali tako da računamo razliku između translirane slike F i slike B za svaki par  $(dx, dy)$  gdje  $dx$  i  $dy$  iteriramo od -10 do 10 uz korak od 2 (ukupno 100 kombinacija) te biramo onaj par  $(dx, dy)$  koji daje najmanju razliku. Zatim na isti način još radimo finu kalibraciju u krugu od jednog piksela od pronađenog najboljeg para  $(dx, dy)$  te je to konačni pomak.

Treba napomenuti da razliku između translirane slike F i slike B računamo tako da zbrojimo razlike za svaki par odgovarajućih piksela. Važno je primijetiti da se translacijom slike F pojavljuju nedefinirani pikseli te moramo odlučiti kako ćemo ih uspoređivati s pikselima sa slike B. Bilo bi loše kada bismo pretpostavili da su jednaki pikselima iz slike B jer bi onda optimalan pomak bio upravo jednak ( $\text{širina\_slike}$ ,  $\text{visina\_slike}$ ). Pokazalo se dobrim tu razliku definirati na neku veliku vrijednost jer se tako kažnjavaju velike translacije te izbjegavaju one koje ne doprinose bitno smanjenju razlike.

Nakon što smo pronašli pomak (dx, dy) radimo translaciju slike F za (-dx, -dy). Postavlja se pitanje na koju vrijednost postaviti nedefinirane piksele nastale translacijom slike F. Pošto ionako pretpostavljamo da smo sliku F uspješno kalibrirali sa slikom B onda te nedefinirane piksele postavljamo na iste vrijednosti kao što su odgovarajući pikseli na slici B.

Opisani postupak se pokazao uspješnim, no zbog računske složenosti smo ga ograničili da radi samo za pomake do 10 piksela (što se pokazalo dovoljnim).

### 2.1.1.2 Zaglađivanje slika

Kako bismo donekle usrednjili vrijednosti piksela i ublažili šumove radimo zaglađivanje (engl. *smoothing*) slike po blokovima. Za to koristimo Gaussovu funkciju  $G(x, y) = \frac{1}{2\mu\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ . Prvo zagladimo posebno sliku F i sliku B gdje koristimo blokove veličine k1, zatim radimo razliku slika te potom ponovno zagladimo tu razliku gdje koristimo blokove veličine k2. Kao najbolje vrijednosti pokazale su se k1 = 3 i k2 = 7.

### 2.1.1.3 Računanje razlike slika

Računanjem razlike između slika F i B dobivamo novu sliku R. Dobivena slika R je crno - bijela što znači da joj svaki piksel poprima vrijednosti između 0 i 255.

Vrijednost piksela  $p_R(x, y)$  slike R računamo ovako:

$$p_R(x, y) = \max(255, |p_{F,red}(x, y) - p_{B,red}(x, y)| + |p_{F,green}(x, y) - p_{B,green}(x, y)| + |p_{F,blue}(x, y) - p_{B,blue}(x, y)|)$$

Na dobivenoj slici R sada su pikseli pozadine tamniji, a pikseli objekta svjetliji.

Osim ove metode probali smo i sličnu metodu koja je dala vrlo sličan ali ipak nešto lošiji rezultat:

$$p_R(x, y) = \max(|p_{F,red}(x, y) - p_{B,red}(x, y)|, |p_{F,green}(x, y) - p_{B,green}(x, y)|, |p_{F,blue}(x, y) - p_{B,blue}(x, y)|)$$

### 2.1.1.4 Binarizacija slike

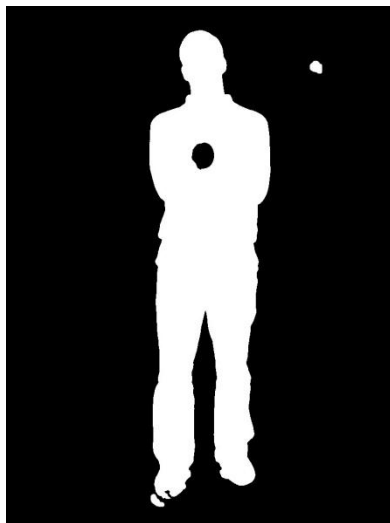
Nakon što smo dobili sliku R gdje pikseli poprimaju vrijednosti od 0 do 255 želimo iz nje dobiti binarnu sliku S gdje pikseli siluete poprimaju vrijednost 1, a pikseli pozadine vrijednost 0. Sliku S dobivamo tako da odabiremo granicu g iznad koje sve piksele postavljamo na 1, a ispod na 0. Tu granicu nazivamo granicom binarizacije. S obzirom na to da je moguće da se slike F i B razlikuju u količini osvjetljenja moguće je da vrijednosti piksela pozadine na slici R nisu vrlo bliske nuli već nešto veće. Kako god, pretpostavljamo da

je promjena osvjetljenja, ako je do nje došlo, otprilike jednaka na cijeloj slici što znači da bi pikseli pozadine na slici R trebali imati međusobno bliske vrijednosti. Također pretpostavljamo da je veći broj slike pokriven pozadinom nego objektom. Poduprti tim pretpostavkama možemo reći da je prosječna vrijednost piksela pozadine na slici R jednaka medijanu svih piksela slike R, i tu ćemo vrijednost koristiti za izračun granice binarizacije slike. Mogli bismo direktno koristiti medijan kao granicu, no tada bismo puno piksela pozadine karakterizirali kao piksele siluete jer pikseli pozadine variraju oko medijana. Zato moramo uvesti dodatnu toleranciju pa granicu binarizacije  $g$  računamo kao  $g = \text{medijan}(R) + \text{thresh}$  gdje je  $\text{thresh}$  parametar tolerancije. Kao optimalna vrijednost tog parametra pokazalo se  $\text{thresh} = 20$ . Postupkom binarizacije sa granicom  $g = \text{medijan}(R) + \text{thresh}$  dobivamo binarnu sliku S koja sadrži siluetu.

### 2.1.1.5 Pročišćavanje siluete

Na slici S pojavljuju se dvije vrste pojava koje želimo izbjeći: rupe unutar siluete i otoci.

Otok je skup piksela siluete koji čine cjelinu, ali su odvojeni od glavne cjeline piksela siluete. Otoci mogu nastati odvajanjem dijelova siluete ili pogrešnim prepoznavanjem dijela pozadine.



Slika 10: Primjer slike sa rupama i otocima

Prvo uklanjamo rupe unutar siluete i to tako da nađemo konture siluete te zatim sve piksele unutar kontura označimo kao piksele siluete.

Zatim uklanjamo otoke tako da identificiramo sve odvojene cjeline piksela siluete te uklonimo sve osim najveće jer za najveću pretpostavljamo da je silueta. Za identificiranje

odvojenih cjelina piksela i računanje njihove veličine koristimo popunjavanje poplavom (engl. flood fill).

Kao konačni rezultat dobivamo binarnu sliku  $S$  gdje su pikseli siluete označeni s 1, a pikseli pozadine s 0. Zajamčeno je da svi pikseli siluete čine jednu cjelinu i da nema rupa unutar siluete.

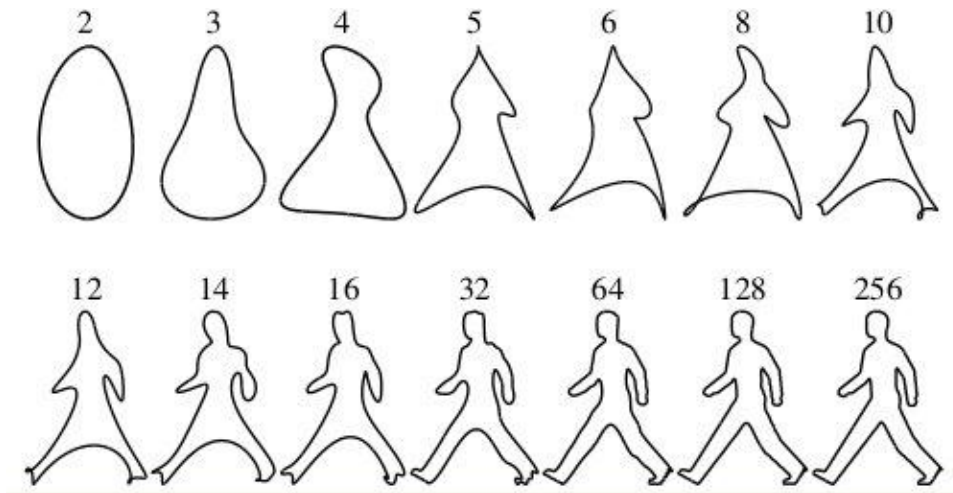
Opisani postupak pokazao se efikasnim na slikama slikanim u zatvorenom prostoru s dvodijelnom pozadinom, dok je za slike na otvorenom, s kompleksnom pozadinom, davao nešto lošije rezultate. Velike probleme također uzrokuju osobe čija je odjeća slična boji pozadine. Naime zbog zaglađivanja i sličnih mjera kojima postupak postaje robustan na šum i promjene u osvjetljenju postupak također teže raspoznaje manje razlike boje pa se često dogodi da odjeću sličnu pozadini označi kao pozadinu pri čemu dobivene siluete postaju u velikoj mjeri neiskoristive.

## 2.2 Granlundovi koeficijenti

Vrlo važan problem u postupku raspoznavanja uzoraka predstavlja pronalaženje optimalnih značajki. Kod problema raspoznavanje različitih oblika na slici općenito postoje dva postupka:

- postupak raspoznavanja oblika na temelju regije koje zauzima
- postupak raspoznavanja oblika na temelju njegove konture

Potonji postupak ekstrahira značajke koje su vezane samo za rub objekta (konturu). Jedan od najpopularnijih načina za raspoznavanje na temelju konture temelji se na upotrebi Fourierovih koeficijenata. Svoju popularnost duguju vrlo dobrom raspoznavanju oblika, jednostavnošću implementacije te jednostavnošću interpretacije. Interpretacija koeficijenata jednostavno se može objasniti uz pomoć slike 8. Slika je dobivena uporabom Fourierove transformacije s različitim brojem koeficijenata te potom rekonstrukcijom konture na temelju istih koeficijenata. Sa slike se vidi kako broj koeficijenata uzetih u obzir utječe na reprezentaciju slike. Dok koeficijenti s nižim indeksom (niske frekvencije) sadrže informacije o općenitom/generalnom obliku konture, koeficijenti s višim indeksom (visoke frekvencije) sadrže informacije o finim detaljima konture.

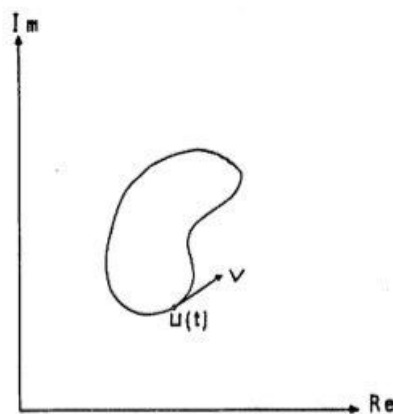


Slika 11: Rekonstrukcija konture na temelju Fourierovih koeficijenata

No, Fourierovi koeficijenti imaju i nedostataka, a to su: ovisnost o rotaciji, translaciji, skaliranju konture i odabiru početne točke kod njihovog računanja. Navedene nedostatke rješava upotreba Granlundovih koeficijenata.

### 2.2.1 Računanje Fourierovih koeficijenata

Kontura u navedenom problemu predstavlja uzorak koji je potrebno klasificirati. Svaka kontura može se shvatiti kao zatvorena krivulja u kompleksnoj ravnini, kao što je prikazano na slici 9.



Slika 12: Krivulja u kompleksnoj ravnini

Kontura se zapisuje jednodimenzionalnom kompleksnom periodičnom funkcijom  $u(t)$  koja crpi vrijednosti iz skupa kompleksnih brojeva:

$$\{(x_t + y_t): t = 0, 1, \dots, T - 1\},$$

gdje je  $(x_t + y_t)$  točka na konturi, a  $T$  broj točaka konture.

Funkciju  $u(t)$  moguće je izraziti Fourierovom transformacijom. Koeficijenti Fourierove diskretne transformacije su tada



$$a_n = \frac{1}{T} \sum_{t=0}^{T-1} d(t) e^{-j2\pi \frac{nt}{T}},$$

a funkcija je jednaka

$$u(t) = \sum_{n=0}^{T-1} a_n e^{j2\pi \frac{nt}{T}}.$$

Vrlo važna činjenica kod računanja Fourierovih koeficijenata predstavlja smjer kretanja po krivulji. Jednom kada je odabran smjer kretanja, isti se smjer mora upotrijebiti i za računanje koeficijenata ostalih kontura.

### 2.2.2 Računanje Granlundovih koeficijenata

Budući da su Fourierovi koeficijenti zavisni, ne samo o obliku konture, već i o izboru početne točke, translaciji, rotaciji i povećanju konture, G. H. Granlund je u svom članku [1] definirao funkcije Fourierovih koeficijenata koje eliminiraju sve te utjecaje:

$$d_{mn} = \frac{a_{1+m}^n a_{T+1-n}^m}{a_1^{m+n}},$$

gdje  $m = 1, 2, 3, \dots$ , a  $n = 2, 3, 4, \dots$

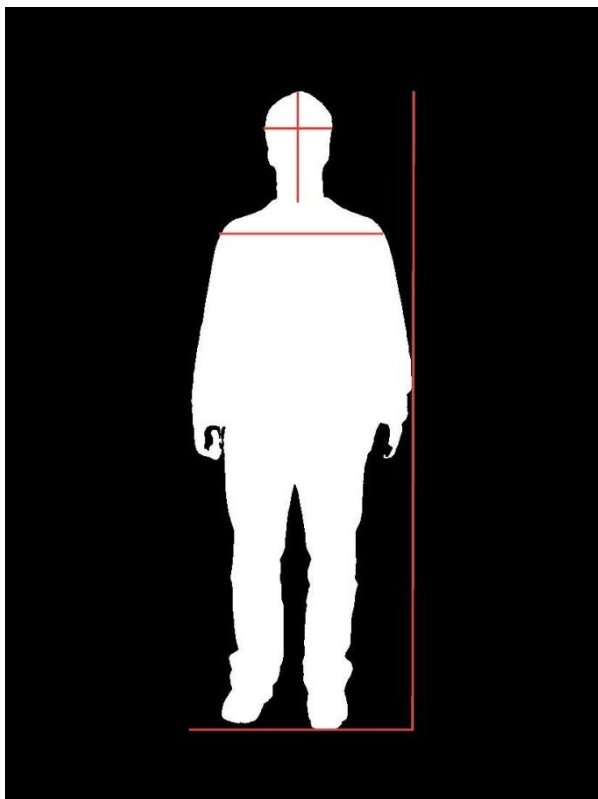
Prikazani koeficijenti predstavljaju značajke u ovom postupku raspoznavanja siluete. Granlundovi koeficijenti ne pate od nedostataka Fourierovih koeficijenata, no ne sadrže informaciju koju je lagano interpretirati. Kao značajka može se koristiti samo jedan ili više Granlundovih koeficijenata. U fazi učenja računaju se Granlundovi koeficijenti za sve slike iz seta za učenje. Korišteni Granlundovi koeficijenti su  $d_{12}, d_{13}, d_{14}, d_{15}, d_{22}, d_{23}, d_{24}, d_{25}, d_{32}, d_{33}, d_{34}, d_{35}, d_{42}, d_{43}, d_{44}, d_{45}, d_{52}, d_{53}, d_{54}, d_{55}$ . Izračunati koeficijenti predstavljaju vektor značajki za svaku sliku i spremaju se u mapu.

Kod klasifikacije se računa udaljenost vrijednosti koeficijenata testne slike od vrijednosti koeficijenata svih slika u mapi i traži se ona slika koja ima najmanju prosječnu udaljenost.

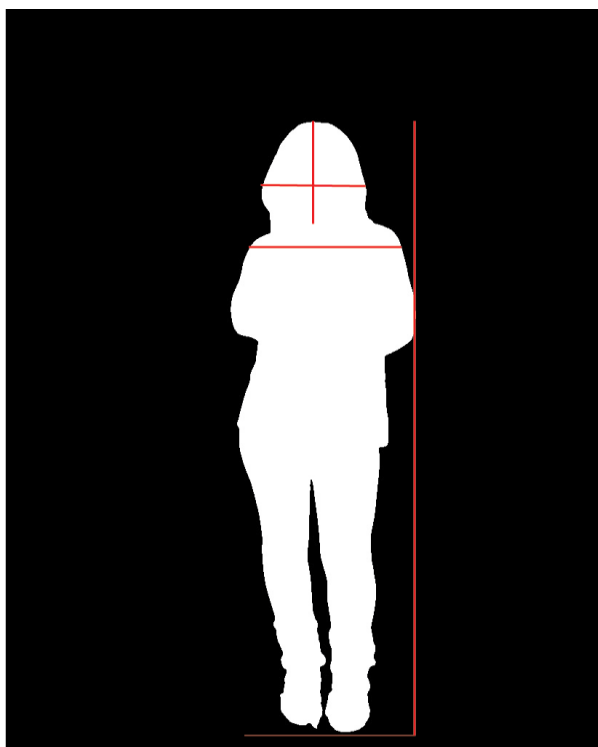
## 2.3 Klasificiranje na osnovu proporcija tijela

Glavne značajke koje se mogu izlučiti iz siluete su svakako visina i širina siluete. Jedna od najvažnijih značajki je visina jer je njome vrlo lako razlikovati dvije siluete, već i za relativno male razlike značajki. No osim same visine i širine siluete, izlučujemo još i visinu glave, širinu glave, te širinu ramena. Naš sustav radi neovisno o udaljenosti kamere od osobe

pri snimanju, zato jer se sve vrijednosti značajki mogu podijeliti s visinom pa se onda barata s omjerima koji su neovisni o udaljenosti kamere ili fotoaparata.



Slika 13: Glavne značajke na muškoj silueti



Slika 14: Glavne značajke na ženskoj silueti

Crvenom bojom su označene značajke izlučene iz siluete - visina siluete, širina siluete, visina i širina glave, te širina ramena. Gledajući gornje dvije slike očito je da postoje znatne razlike u određenim značajkama ovih dviju silueta. Tu činjenicu smo iskoristili pošto je naš skup relativno malen te već i sama značajka visine siluete znatno doprinosi razlikovanju osoba, a uz ostale značajke moguće je relativno dobro klasificiranje.

Samo visinu iz siluete izlučujemo tako što nađemo najviši stupac bijelih piksela, dok za širinu pronalazimo najdulji redak bijelih piksela. Glavu nalazimo krećući s vrha slike siluete dok ne nađemo na bijele piksele, tada pratimo porast u širinama bijelih piksela dok ponovno ne nastane znatan pad pa rast, to jest dok ne nađemo na vrat. Širinu ramena pronalazimo tako da krenemo odande gdje je proces nalaženja visine glave stao - od vrata. Zatim se širimo prema dolje dok promjena u širini ne postane konstantna, odnosno dok ne nađemo na dio tijela koji je ravan.

Pokazalo se da je najbolja značajka od ovih navedenih visina, vjerojatno prije svega jer se radilo na relativno malom skupu podataka pa se visina osoba znatno razlikuje. Rezultati sigurno ne bi bili ni približno dobri samo temeljem visina i širina raznih dijelova tijela siluete mjereno na velikoj populaciji.

Osim toga testirali smo i s ovih pet značajki, ali još podijeljenih sa visinom, dobivši na taj način neovisnost o udaljenosti od kamere pri snimanju siluete.

Izlučene značajke potom koristimo za klasificiranje, te koristimo nekoliko klasifikatora za rješavanje tog problema:

- obično uspoređivanje s kvadratnom greškom pri kažnjavanju
- Bayesova mreža
- stroj s potpunim vektorima (SVM)
- k - najbližih susjeda
- Random Forest

## **2.4 Template matching**

Ova metoda radi na temelju sličnosti dviju slika u vrijednostima određenih značajki slike. Budući da je naša slika siluete crno bijela to je lako izvedivo.

### **2.4.1 Template matching s brojem jednakih piksela**

Metoda prvo centrira sve slike uključujući i testnu sliku te prolazi kroz cijelu testnu sliku i uspoređuje piksele sa slikom iz skupa za učenje. Tako prolazi svim slikama iz skupa za učenje te pronalazi k najbližih slika.

## 2.4.2 Template matching s PCA

Kod slika silueta osoba koje je potrebno raspoznavati u sklopu ovog projekta može se primijetiti velika sličnost između određenih slika. Iz tog razloga metoda raspoznavanja usporedbom elementa s elementom slike nije odgovarajuća. Kod takvih slika javlja se šum kao posljedica nesavršenosti uređaja za snimanje, varijacija u položaju, orijentacije osobe prilikom snimanja i slično. Korištenjem metode analize glavnih komponenti (*principal component analysis* - PCA) izbjegao bi se utjecaj ranije navedenih problema. PCA metoda temelji se na smanjenju informacije koje su sadržane u slici, a da se pri tome zadrže informacije o najvažnijim značajkama. PCA se koristi nad jednodimenzionalnim podacima, a budući da slika sadrži informacije u dvije dimenzije, potrebno ju je prikazati pomoću vektor - stupca, što za sliku dimenzija  $N \times N$  znači prikaz značajki u  $N^2$ -dimenzionalnom prostoru. Metodom PCA traži se prostor (skup ortogonalnih vektora) manjih dimenzija od originalnog prostora u kojem će vektori slika biti postavljeni tako da varijacija podataka bude maksimalna. Matematički rečeno, prikaz slike u novom prostoru dobiva se množenjem ortogonalne matrice  $\mathbf{V}$  i matrice slike  $\mathbf{X}$  prema formuli:

$$\mathbf{Y} = \mathbf{V}\mathbf{X} \quad (1)$$

Unutar matrice  $\mathbf{Y}$  se odabiru samo oni vektori u čijem je smjeru varijanca najveća. Pomoću PCA iz skupa slika za treniranje moguće je dobiti skup najvažnijih značajki unutar skupa za treniranje te skup istih slika predloženih pomoću tih značajki. I ulazna slika koju je potrebno klasificirati mora se transformirati u isti prostor značajki. Skup za treniranje je matrica čiji su stupci slike siluete koje su pretvorene u vektore. Da bi se zadržale sve bitne značajke po kojima se slike razlikuju, potrebno je oduzeti srednju vrijednost svih slika – „prosječnu sliku“.



Slika 15: Rezultat centriranja slike



Slika 16: Rezultat usrednjivanja slike

Nakon toga je potrebno izračunati korelacijsku matricu  $C_X$  koje se računa prema:

$$C_X = E[XX^T] \quad (2)$$

gdje  $\mathbf{X}$  predstavlja matricu sa vektorima slika za treniranje od kojih je oduzeta srednja vrijednost („prosječna slika“).

$$C = \begin{bmatrix} \text{cov}(a_{11}, a_{11}) & \text{cov}(a_{11}, a_{12}) & \cdots & \text{cov}(a_{11}, a_{nn}) \\ \text{cov}(a_{12}, a_{11}) & \text{cov}(a_{12}, a_{12}) & \cdots & \text{cov}(a_{12}, a_{nn}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(a_{nn}, a_{11}) & \text{cov}(a_{nn}, a_{12}) & \cdots & \text{cov}(a_{nn}, a_{nn}) \end{bmatrix}$$

U korelacijskoj matrici  $\mathbf{C}$  svaki pojedini element  $c_{ij}$  predstavlja međusobne korelacije pojedinih elemenata vektora slike i vrijednosti tog elementa slike, dok su elementi na glavnoj dijagonali  $c_{ii}$  varijance tog elementa slike, odnosno njihovih vrijednosti. Nakon što je izračunata matrica korelacije  $\mathbf{C}$  može se pronaći matrica transformacije koja će najbolje projicirati matricu  $\mathbf{X}$  na novi prostor značajki.

Matrica transformacije  $\mathbf{V}$  se dobiva tako da se izračunaju svojstveni vektori korelacijske matrice  $\mathbf{C}$  te poredaju u stupce s obzirom na odgovarajuće svojstvene vrijednosti tako da oni svojstveni vektori s najvećom svojstvenom vrijednošću budu prvi i oni predstavljaju najvažnije značajke u slici. Svaki svojstveni vektor predstavlja jednu siluetu.

Konačno, kako bi se završila faza treniranja potrebno je projicirati vektor slike za treniranje na svojstvene vektore i spremi te projekcije kao vektor značajki. Od tih značajki odabire se prvih nekoliko koje predstavljaju one najvažnije značajke. Time se postiže da se dimenzionalnost značajki smanji na puno manje, ovisno o tome koliko je dimenzija potrebno za što bolju klasifikaciju uzimajući u obzir i vremensko trajanje klasifikacije.

Kako bi mogli uspoređivati ulaznu sliku sa skupom slika za treniranje potrebno je ulaznu sliku (od koje je bila oduzeta srednja vrijednost svih slika) pomnožiti sa spremljenim vektorom značajki svih slika i dobiju se značajke za tu određenu sliku koje se onda uspoređuju s ostalima i određuje se najsličnija slika.

### 3. Ispitivanje rješenja

#### 3.1 Ispitna baza

Za potrebe ovog projekta, odlučili smo izraditi vlastitu bazu slika pošto nismo uspjeli naći postojeću bazu koja bi zadovoljavala naše potrebe.

Baza sadrži 248 slika od ukupno 24 osobe. Slike su uslikane digitalnim fotoaparatom razlučivosti 3648 x 2736 piksela postavljenim na visini od 156 cm i udaljenim 319 cm od osobe. Slike su naknadno reducirane na 30% originalne veličine i takve spremljene u bazu, što rezultira slikama razlučivosti od 821 x 1094 piksela. Takva razlučivost pokazala se dovoljno dobrom za izlučivanje kvalitetne siluete, također i značajno smanjuje ukupnu veličinu baze.

Prilikom izrade baze cilj nam je bio imati slike više različitih poza svake osobe. Svaka osoba slikana je frontalno. Također, prije svake slike osoba je zamoljena da ponovno uđe u kadar kako bismo osigurali prirodnost i nepovezanost uzastopnih snimki iste osobe.

Većina slika napravljena je u zatvorenom prostoru, a dodatno smo za još osam osoba (članovi tima) napravili dio slika na otvorenom kako bismo testirali uspješnost sustava. Ipak, pokazalo se kako tako dobivene siluete nisu zadovoljavajuće kvalitete i stoga te slike nisu korištene u učenju ni ispitivanju sustava. Slike su ostavljene u bazi za svrhe budućeg rada.

Osim samih slika, baza se sastoji i od silueta izlučenih iz njih. Iz svake slike izlučena je silueta i sustav dalje cijelo vrijeme radi samo sa siluetama. Iz nekih slika nije bilo moguće izlučiti siluetu zadovoljavajuće kvalitete, stoga takve slike nisu korištene.

#### 3.2 Rezultati učenja i ispitivanja

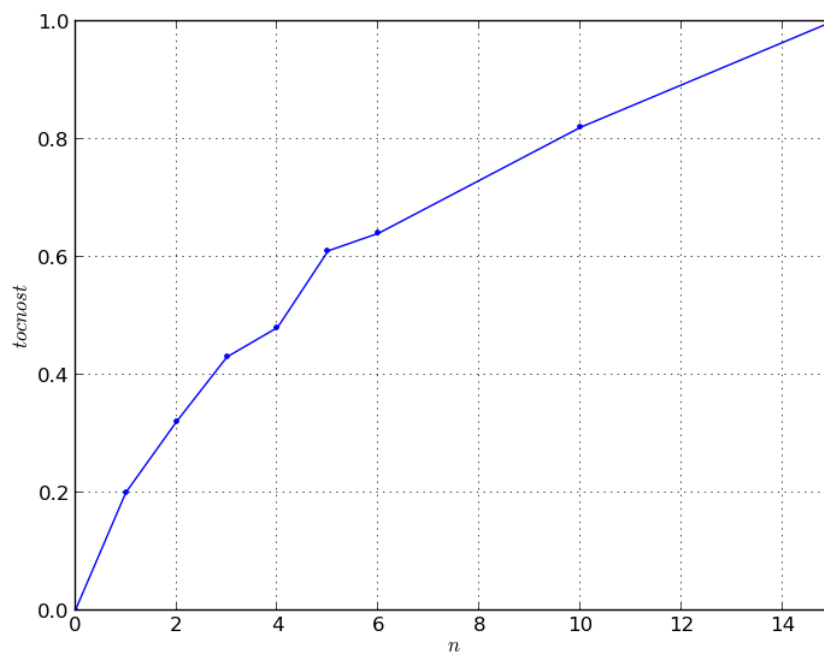
##### 3.2.1 Rezultati - Granlundovi koeficijenti

Uspoređivanje bez PCA s koeficijentima  $d_{12}, d_{13}, d_{14}, d_{15}, d_{22}, d_{23}, d_{24}, d_{25}, d_{32}, d_{33}, d_{34}, d_{35}, d_{42}, d_{43}, d_{44}, d_{45}, d_{52}, d_{53}, d_{54}, d_{55}$ :

$n$	Točnost
1	0.196429
2	0.321429
3	0.428571
4	0.482143
5	0.607143
6	0.642857

10	0.821429
15	1

Tablica 1: Rezultati testiranja metode Granlundovih koeficijenata



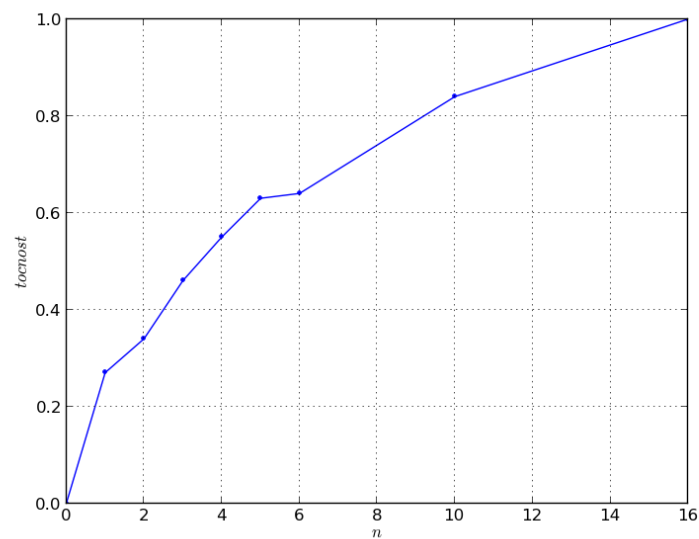
Slika 17: Točnost metode Granlundovih koeficijenata

Uspoređivanje s korištenjem PCA i jednakim koeficijentima:

<i>n</i>	Točnost
1	0.267857
2	0.339286
3	0.464286
4	0.553571
5	0.625
6	0.642857
10	0.839286
16	1

Tablica 2: Rezultati testiranja metode Granlundovih koeficijenata uz korištenje PCA





Slika 18: Točnost metode Granlundovih koeficijenata uz korištenje PCA

Uspoređivanje s odabranim drugačijim koeficijentima ( $d_{12}$ ,  $d_{22}$ ,  $d_{23}$ ,  $d_{24}$ ,  $d_{25}$ ,  $d_{33}$ ,  $d_{43}$ ,  $d_{44}$ ):

- **k - najbližih susjeda**  
Točnost ( $n = 1$ ): 0.178571
- **Bayesov klasifikator**  
Točnost ( $n = 1$ ): 0.107143
- **SVM**  
Točnost ( $n = 1$ ): 0.107143
- **Random Forest**  
Točnost ( $n = 1$ ): 0.160714

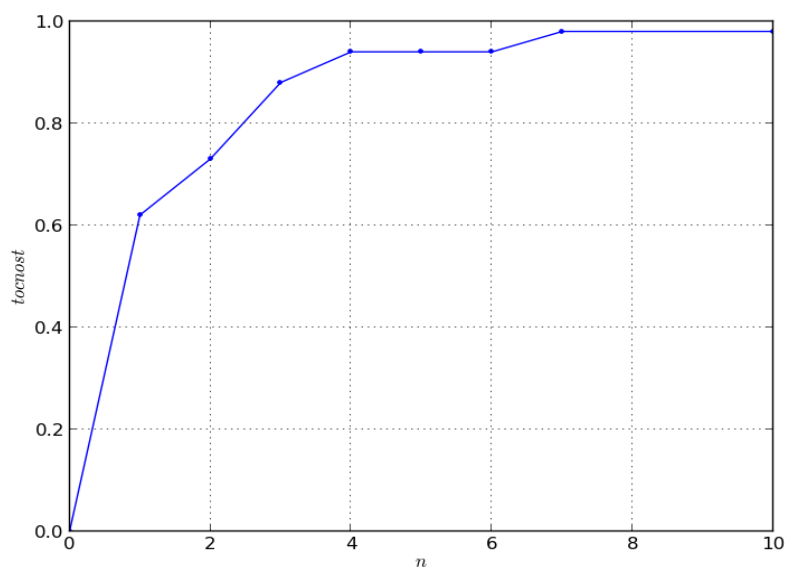
### 3.2.2 Rezultati - Klasificiranje na osnovu proporcije tijela

Uspoređivanje uz kvadratnu grešku kod kažnjavanja:

$n$	Točnost
1	0.625
2	0.732143
3	0.875
4	0.946249
5	0.946249
6	0.946249

7	0.982143
8	0.982143
9	0.982143
10	0.982143

Tablica 3: Rezultati testiranja metode klasifikacije na osnovu proporcije tijela uz kvadratnu pogrešku

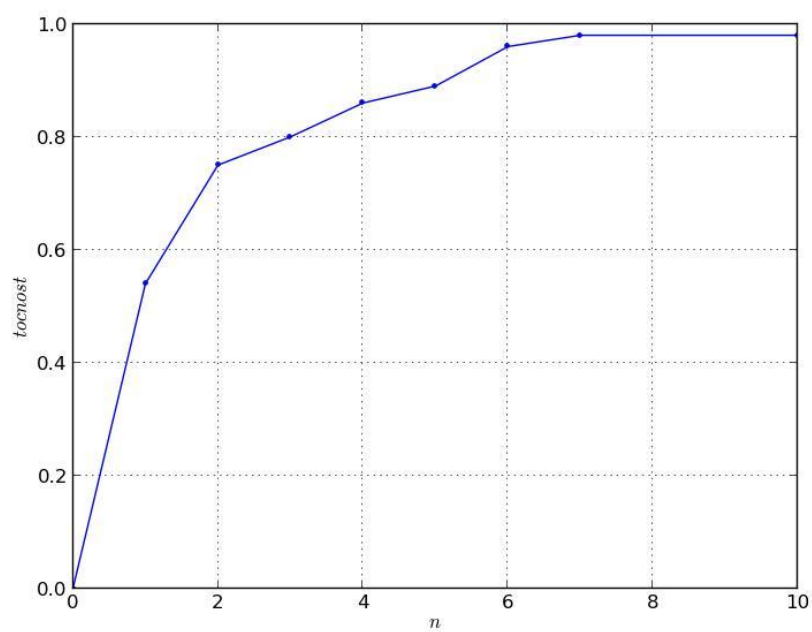


Slika 19: Točnost u ovisnosti o broju najbližnjih slika

Ako se sve značajke podijele s visinom, a i dalje se koristi kvadratna pogreška:

<i>n</i>	Točnost
1	0.535714
2	0.75
3	0.803571
4	0.857143
5	0.892857
6	0.964286
7	0.982143
10	0.982143

Tablica 4: Rezultati testiranja metode klasifikacije na osnovu proporcije tijela uz kvadratnu pogrešku (značajke podijeljene s visinom)



Slika 20: Točnost u ovisnosti o broju najsličnijih slika

- **k - najbližih susjeda**

Točnost ( $n = 1$ ): 0.535714

- **Bayesov klasifikator**

Točnost ( $n = 1$ ): 0.0714286

- **SVM**

Točnost ( $n = 1$ ): 0.0714286

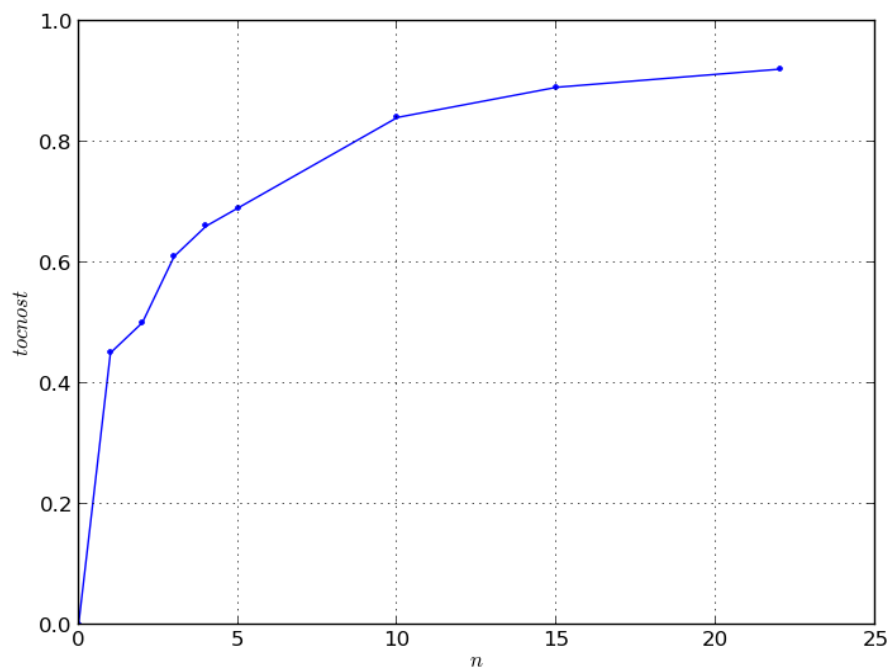
- **Random Forest**

Točnost ( $n = 1$ ): 0.5

### 3.2.3 Rezultati - Template matching

$n$	Točnost
1	0.446429
2	0.5
3	0.607143
4	0.660714
5	0.696429
10	0.839286
15	0.892857
22	0.928571

Tablica 5: Rezultati testiranja metode template matching

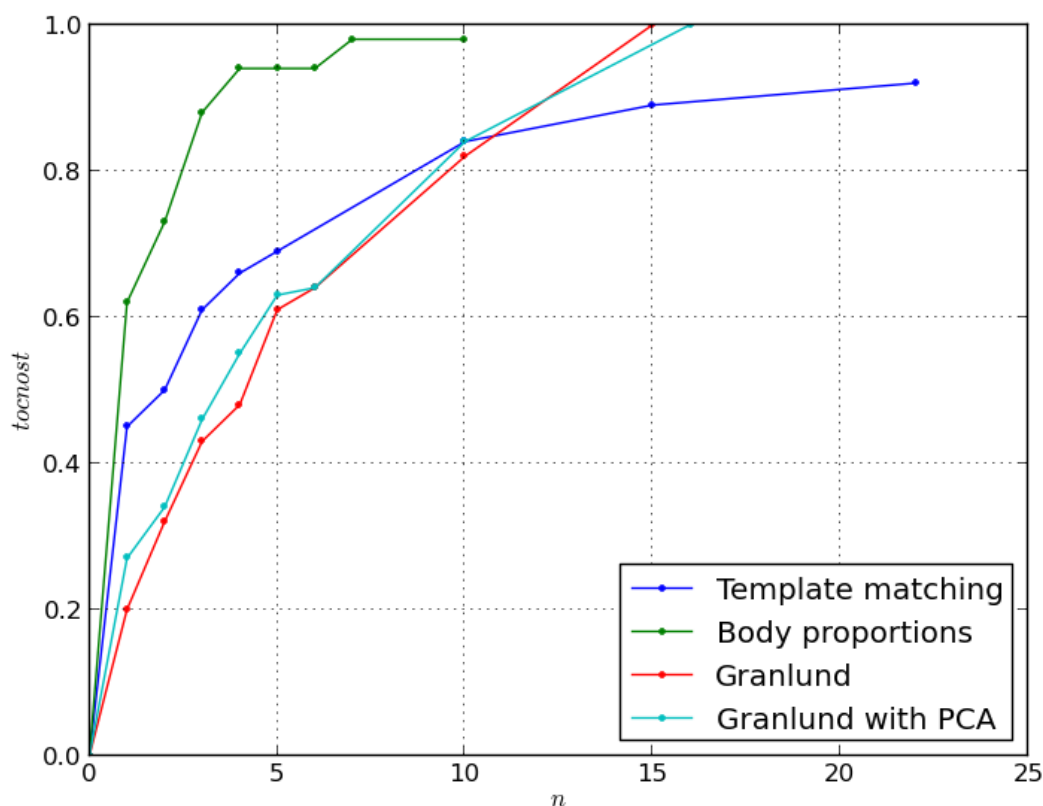


Slika 21: Točnost u ovisnosti o broju najbližnjih slika

### 3.2.4 Rezultati - Template matching s PCA

Rađeno je testiranje na skupu od 44 slika za učenje (4 slike po osobi) i 11 slika za testiranje (1 slika po osobi). Presjek tih skupova je naravno prazan skup. Implementacija za svaku testnu sliku izbacuje najbližnju u skupu slika (silueta) za testiranje. Dobivena uspješnost je 63%.

### 3.2.5 Usporedba rezultata



Slika 22: Usporedba točnosti nekih korištenih metoda

### 3.3 Analiza rezultata

Ukupan skup podataka, odnosno slike iz baze, podijeljen je na skup za učenje i skup za ispitivanje. Ukupno je 56 slika odabrano za testiranje, tako da je za svaku osobu odabran reprezentativan skup od dvije do četiri slike.

Rezultat testiranja koji nas zanima jest točnost koja se može izraziti kao udio točnih klasifikacija u ukupnom broju klasifikacija. S obzirom na zahtjevnu prirodu problema, dopuštamo da se pojedina klasifikacija proglasi točnom ako se klasa kojoj pripada zadani primjer nalazi u prvih  $n$  klasa koje prema izlazu klasifikatora imaju najveću izglednost.

Tako možemo pratiti točnost klasifikatora u ovisnosti o parametru  $n$ , koji određuje koliko ćemo najizglednijih klasa uzimati u obzir pri određivanju točnosti.

#### 3.3.1 Granlundovi koeficijenti

Ako se gleda samo najbolja slika, točnost klasifikacije je 19.64% bez PCA i 26.79% uz PCA, što je s obzirom na ostale isprobane metode loš rezultat. Glavni problem ovog

algoritma je što se apsolutno oslanja na oblik tijela, a u bazi svaka osoba ima više različitih položaja tijela. Zbog toga algoritam uglavnom spaja iste položaje tijela, a ne iste osobe. Granlundovi koeficijenti bili bi dobro rješenje kad bi se baza sastojala od slika osoba u istom položaju. Također, vidi se da korištenje PCA pomaže, ali ne značajno. Metoda je još uvijek lošija od ostalih.

### 3.3.2 Klasificiranje na osnovu proporcija tijela

Najbolji rezultat daje obično uspoređivanje značajki uz kažnjavanje kvadratnim odstupanjem, približno jednake rezultate daje metoda k - najbližih susjeda. Metode kao što su SVM ili Bayes imaju dosta nisku točnost što možemo objasniti činjenicom da u skupu podataka nema normalne razdiobe jer je skup premalen. Te metode bi vjerojatno bile znatno uspješnije da su testirane na slučajnom skupu uzoraka od nekoliko tisuća ljudi, a s druge strane na tako velikom skupu podataka, metode običnog uspoređivanja i k-NN bi imali jako loše rezultate. Pošto je skup podataka dovoljno malen, značajke kao što su visina i širina siluete te dimenzije pojedinih dijelova tijela su već dovoljne za relativno uspješno klasificiranje, odnosno meko raspoznavanje osobe na temelju siluete. Činjenica leži u tome da već i mala odstupanja u visini bivaju dobro razlikovana te metode usporedbe značajki daju dobre rezultate. Tako metoda kažnjavanje kvadratnom greškom daje točnost od 62.5% koristeći samo pet značajki (visina siluete, širina siluete, visina glave, širina glave i širina ramena), a ako gledamo odgovara li određena silueta u 3 najbolje tada točnost raste na 87.5%. Za četiri najbolja raste na dosta sigurnih 94.64%. Metoda k - najbližih susjeda i Random Forest imaju preciznost od oko 50%, dok SVM i Bayesov klasifikator imaju točnost manju od 10%.

Jako je zanimljiva činjenica da ako sve značajke podijelimo s visinom dobivamo gotovo jednako dobre rezultate kao i ako ne koristimo omjere, što je izvrsno jer je tako naš sustav zapravo postao neovisan o udaljenosti s koje je slika odnosno silueta snimljena. Vidimo da za točno klasificiranje uz dijeljenje sa značajkom visine siluete imamo točnost od 53.57% (ovo je i dalje više od svih ostalih metoda, osim k-NN s kojim je približno jednak), dok bez dijeljenja ta točnost iznosi 62.5%. No, već za  $k = 2$ , odnosno  $k \geq 3$  rezultati postaju jako sličnih preciznosti, preko 90%, što je odlično svojstvo sustava.

### 3.3.3 Template matching s PCA

Rezultati su relativno dobri (63 %). Veća točnost bi se mogla dobiti predočavanjem većeg broja slika (silueta) po osobi odnosno povećavanjem skupa za učenje. Povećavanje

skupa za učenje predstavlja problem jer nije moguće dobiti veliki skup slika za jednu osobu jer se gledaju samo siluete te ponavljanjem slične pozicije ne dobivamo ništa (dobivamo vrlo sličnu siluetu).

### **3.3.4 Template matching**

Dobiveni rezultati su za samo jednu najbolju sliku 44% što nije puno. Ali gledajući brzinu i jednostavnost metode to je dovoljno dobar rezultat. Ako gledamo postotak uspješnosti metode u 3 najbolje slike dobivamo uspješnost od 60% što je puno bolje, ali još uvijek nedovoljno dobro.

## 4. Opis programske implementacije rješenja

Programska implementacija napisana je u programskom jeziku C++ uz korištenje OpenCV knjižnice, popularnog javno dostupnog skupa metoda i algoritama za računalni vid i strojno učenje.

Kod je objektno strukturiran, a osnovu implementacije čini apstraktni razred *Classifier* koji sadrži virtualne metode *learn()* i *classify()*, te već implementiranu javnu metodu *test()*.

Pri implementaciji konkretnog klasifikatora dovoljno je samo naslijediti razred *Classifier* i implementirati dvije spomenute virtualne metode. Pozivom metode *test()* izvršava se automatsko testiranje sustava, čime je osigurano jedinstveno testiranje svih isprobanih algoritama.

Metoda *learn()* prima skup za učenje kao parametar i služi kako bi algoritam stvorio strukture podataka potrebne za klasifikaciju. Ovdje se također odvija izlučivanje značajki iz silueta.

Metoda *classify()* prima primjer (siluetu) koju je potrebno klasificirati, a vraća skup klasa za koje je najizglednije da im primjer pripada, zajedno s numeričkom interpretacijom izglednosti za svaku klasu.

Metoda *test()* poziva prvo *learn()* koji inicijalizira klasifikator, a zatim za svaki primjer iz skupa za učenje zove *classify()*. Na kraju se generira HTML izvještaj koji sadrži matricu zabune, rezultat svake pojedine klasifikacije i točnost klasifikatora.



Pred / Real	sil10_ivana	sil11_bagic	sil12_viktor	sil13_tena	sil15	sil17_lucija	sil18	sil19	sil20	sil22_vanda	sil23	sil24	sil2_hrvoje	sil3_igor	sil4_matija	sil5_martin	sil6_petar	sil7_miranda	sil9_zuzic
sil10_ivana	2	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	2	0
sil11_bagic	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sil12_viktor	0	0	1	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0
sil13_tena	0	0	0	2	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
sil15	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sil17_lucija	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
sil18	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
sil19	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0
sil20	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
sil22_vanda	1	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	1	0
sil23	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
sil24	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	1	0	0
sil2_hrvoje	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
sil3_igor	0	0	0	0	0	0	0	1	0	0	0	0	0	2	3	0	1	0	0
sil4_matija	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
sil5_martin	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
sil6_petar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
sil7_miranda	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sil9_zuzic	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2

Correctness (only the best): 25/56 = 0.446429

Correctness (in 5 best): 39/56 = 0.696429

Img ID	Class name	1	2	3	4	5
1	sil10_ivana	sil10_ivana : 0.971057	sil10_ivana : 0.958357	sil22_vanda : 0.926504	sil19 : 0.925712	sil13_tena : 0.925144
2	sil10_ivana	sil22_vanda : 0.920204	sil22_vanda : 0.919236	sil13_tena : 0.914641	sil10_ivana : 0.911659	sil17_lucija : 0.906591
3	sil10_ivana	sil10_ivana : 0.950973	sil23 : 0.921936	sil23 : 0.921431	sil4_matija : 0.906641	sil20 : 0.885725
4	sil11_bagic	sil9_zuzic : 0.904893	sil19 : 0.903365	sil11_bagic : 0.903153	sil4_matija : 0.90219	sil2_hrvoje : 0.897185
5	sil11_bagic	sil11_bagic : 0.914746	sil19 : 0.897637	sil9_zuzic : 0.897598	sil6_petar : 0.893244	sil4_matija : 0.889843
6	sil11_bagic	sil11_bagic : 0.91646	sil12_viktor : 0.903543	sil4_matija : 0.900547	sil9_zuzic : 0.8996	sil11_bagic : 0.898035
7	sil12_viktor	sil4_matija : 0.907389	sil12_viktor : 0.897535	sil9_zuzic : 0.894778	sil20 : 0.887195	sil20 : 0.886779
8	sil12_viktor	sil12_viktor : 0.950251	sil12_viktor : 0.906298	sil4_matija : 0.905631	sil5_martin : 0.895801	sil3_igor : 0.884668
9	sil12_viktor	sil5_martin : 0.909398	sil12_viktor : 0.907881	sil20 : 0.901253	sil20 : 0.899882	sil4_matija : 0.889032
10	sil13_tena	sil13_tena : 0.96303	sil13_tena : 0.94179	sil5_martin : 0.931442	sil19 : 0.928383	sil10_ivana : 0.923537
11	sil13_tena	sil13_tena : 0.943692	sil13_tena : 0.93168	sil10_ivana : 0.924464	sil10_ivana : 0.923917	sil19 : 0.916007
12	sil13_tena	sil15 : 0.91061	sil5_martin : 0.905406	sil22_vanda : 0.903817	sil3_igor : 0.901557	sil13_tena : 0.896713
13	sil15	sil15 : 0.928001	sil3_igor : 0.914147	sil15 : 0.902344	sil3_igor : 0.900523	sil22_vanda : 0.899032
14	sil15	sil15 : 0.93415	sil15 : 0.916308	sil3_igor : 0.90578	sil24 : 0.899856	sil4_matija : 0.893943
15	sil17_lucija	sil17_lucija : 0.892475	sil10_ivana : 0.891572	sil17_lucija : 0.891068	sil10_ivana : 0.882907	sil24 : 0.879487
16	sil17_lucija	sil10_ivana : 0.915406	sil17_lucija : 0.911952	sil13_tena : 0.910765	sil22_vanda : 0.907771	sil10_ivana : 0.906285

Slika 23: Primjer izvještaja

## 4.1 Opis korištenja

Programsko rješenje se sastoji od dva odvojena dijela. Jedan dio je zadužen za izlučivanje siluete, dok se drugi pokreće na bazi izlučenih silueta i testira klasifikatore na temelju konfiguracijske datoteke. Pomoću konfiguracijske datoteke (detaljno je opisana u readme datoteci priloženoj uz izvorni kod), moguće je točno odabrati koje će se slike koristiti za učenje, a koje za testiranje.

U svrhu prezentacije programa, priložena je izvršna datoteka imena „demo“ i odgovarajuća konfiguracijska datoteka. Pri pokretanju korisnik iz naredbenog retka može odabrati jedan od opisanih klasifikatora nakon čega će se program izvršiti i stvoriti izvještaj o njegovoj točnosti.

Također je moguće pokrenuti program koji će izlučiti siluetu iz zadane slike. Implementirano je programsko sučelje koje putem naredbenog retka prima sliku osobe i odgovarajuću sliku pozadine. Izlaz programa je izlučena silueta. Moguće je i programsko

korištenje opisane funkcionalnosti uključivanjem odgovarajućih datoteka u sustav ovisnosti implementacije te pozivanjem metode *silhouette\_extraction()*.

S ovako strukturiranim kodom, moguće je vrlo lako stvoriti novu izvršnu datoteku s prilagođenim ulazima i izlazima, ovisno o specifičnim zahtjevima sustava. Komunikacija s programom odvija se isključivo putem sučelja, što omogućuje i lako dodavanje novih algoritama koji jednostavno implementiraju opisano sučelje klasifikatora.

## 4.2 Opisi implementacija pojedinih algoritama

Izlučivanje značajki kao što su visina i širina siluete, visina i širina glave, te širina ramena dešavaju se u klasi *HWMatching* koja nasljeđuje klasu *Classifier*. Klasa *Classifier* ima već ugrađene metode za automatsko testiranje i iscrtavanje rezultata što znatno olakšava testiranje raznih klasifikatora. Prosljeđivanjem raznih konfiguracijskih varijabli moguće je namjestiti kojim klasifikatorom želimo klasificirati značajke iz skupa uzoraka. Klasifikaciju je moguće odraditi Bayesovim klasifikatorom, kNN klasifikatorom, SVM klasifikatorom, Random Forest klasifikatorom ili običnim uspoređivanjem kvadratne greške značajki koje se na kraju pokazalo i najtočnijim.

Implementacija *Template matching*-a kreće sa centriranjem slike. Pronalazi se centar siluete te područje slike na kojem je silueta, uz uvjet da je to područje što ispunjenije. Nadalje, gleda se samo područje slike koje je dobiveno prethodnim korakom, a ne cijela slika. Takve slike se stavljaju u vektor te se prilikom klasifikacije novih slika one uspoređuju sa slikama u vektoru te se izbacuje k najbližih slika. Uspoređivanje slika radi se brojanjem jednakih piksela kod obje slike.

Implementacija PCA učitava skup slika za učenje u vektor. Nakon toga zahtijeva upis broja komponenti (k) koliko želimo da PCA koristi. Na temelju vektora slika za učenje se izračunavaju jedinstvene vrijednosti te se dobiva vektor k najznačajnijih značajki. Na temelju tih k značajki računa se euklidska udaljenost testne slike sa slikama za učenje te se dobiva najbližnja slika.

## 5. Zaključak

Prilikom implementacije i testiranja mogli su se primijetiti zanimljivi rezultati i promjene kod isprobavanja različitih metoda. U ovom projektu isprobano je dosta metoda za učenje, te dvije metode za testiranje - za template matching s PCA i posebno za ostale metode.

U fazi učenja bilo je različitih pristupa izlučivanju značajki, od centriranja slike odabirom interesnih regija oko same siluete i odbacivanjem ostatka pozadine, preko izračuna Granlundovih koeficijenata, određivanja širine i visine glave osobe, cijele osobe i širine ramena osobe pa do pronalaska srednje slike i prilagođavanja silueta oduzimanjem srednje slike od ostalih i ubacivanjem takvih slika u PCA metodu ili ubacivanjem Granlundovih koeficijenata u PCA umjesto cijele siluete. Korišteni klasifikatori su: Bayesov klasifikator, k - najbližih susjeda, stroj s potpornim vektorima (SVM), *Random Forest*. Ako bi se gledala uspješnost po pojedinim metodama, može se reći da je najuspješnija metoda bila klasificiranje na osnovu proporcije tijela i to s 62.5% (ako se gleda samo jedan najbolji rezultat). Ako bi se gledalo da se osoba nalazi u prve 4 osobe, uspješnost detekcije se povećava i dolazi čak i do 94.64% uspješnosti raspoznavanja. Nakon te metode po uspješnosti dolazi metoda *template matching* s postotkom od 63% ako se gleda samo najsličnija osoba, a 83.92% ako se gleda da se osoba nalazi u skupu od 10 najboljih osoba. Potom slijede sve ostale metode koje obuhvaćaju i PCA i uspoređivanje sličnosti piksel po piksel, Granlundove koeficijente i tako dalje. U tablici 6. Prikazane su sve metode i njihova uspješnost u raspoznavanju kada se gleda samo najsličnija osoba, odnosno kojoj klasi pripada testna silueta. Kod PCA koji uzima kao ulaz cijele slike je korišten manji broj testnih slika i zbog toga je postotak dosta viši, ali također je ovisan i o broju testnih slika i broju značajki koji se odabire. U tablici je prikazan postotak s optimalnim izborom značajki.

Metoda	Klasifikator	Broj testnih slika	Uspješnost
Klasificiranje na osnovu proporcija tijela	Uspoređivanje uz kvadratnu grešku	56	62.5%
Klasificiranje na osnovu proporcija tijela	Najbliži susjed	56	53.57%
Granlundovi koeficijenti	Minimalna udaljenost	56	19.64%
Granlundovi koeficijenti + PCA	Minimalna udaljenost	56	26.79%
PCA/cijele siluete	Najbliži susjed	11	63%

Tablica 6: Prikaz pojedinih metoda i njihove uspješnosti u raspoznavanju

Daljnji razvoj ovog projekta i njegove implementacije obuhvaća razvoj nekog korisničkog sučelja koje će uljepšati i olakšati korištenje implementiranih metoda. Također postoje još dorade vezane uz same metode, na primjer za PCA probati implementirati da ulaz u PCA bude broj bijelih piksela po redcima ili stupcima, probati naći još neke značajke osim samo visine i širine glave i cijele osobe i njezinih ramena, kao što je recimo površina koju zauzima silueta osobe pa dužina nogu i tako dalje. Uglavnom, postoji još dosta različitih pristupa koji bi se mogli upotrijebiti i probati još neki klasifikator koji bi možda bolje odjeljivao uzorke koji se javljaju kod silueta.

Identifikacija osobe na temelju samo njezine siluete je još dosta neistraženo područje i nije još pronađen najbolji način za određivanje najbližnje osobe s vrlo visokim postotkom uspješnosti pa je to područje otvoreno za eksperimentiranje i testiranje. U ovom radu prikazano je nekoliko standardnih metoda korištenjem uobičajenih klasifikatora, a za daljnji razvoj se ostavlja mogućnost isprobavanja novih metoda izlučivanja značajki i učenja te upotreba različitih klasifikatora što bi trebalo još dodatno poboljšati rezultate klasificiranja.

## 6. Literatura

- [1] G. H. Granlund: „Fourier Preprocessing for Hand Print Character Recognition“, Computers, IEEE Transactions on, C-21, veljača 1972., pp 195 - 201  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5008926> (03.01.2013.)
- [2] B. DeCann, A. Ross: "Gait Curves for Human Recognition, Backpack Detection and Silhouette Correction in a Nighttime Environment", Biometric Technology for Human Identification VII., travanj 2010., pp. 76670Q-76670Q-13  
[http://www.cse.msu.edu/~rossarun/pubs/DeCannRossGait\\_SPIE2010.pdf](http://www.cse.msu.edu/~rossarun/pubs/DeCannRossGait_SPIE2010.pdf)  
(20.01.2013.)
- [3] S. Jaggi, W. Clem Karl, S. G. Mallat, A.S.Willsky: „Silhouette recognition using high-resolution pursuit“, Pattern Recognition, Volume 32, Issue 5, svibanj 1999., pp 753–771  
[http://ssg.mit.edu/~willsky/publ\\_pdfs/141\\_pub\\_PR.pdf](http://ssg.mit.edu/~willsky/publ_pdfs/141_pub_PR.pdf) (20.01.2013.)
- [4] C. Hsieh, P.S. Huang, M. Tang: „Human Action Recognition Using Silhouette Histogram“, Instrumentation, Measurement, Computer, Communication and Control, listopad 2011., pp 353 - 356  
<http://crpit.com/confpapers/CRPITV113Hsieh.pdf> (20.01.2013.)
- [5] OpenCV User Guide  
[http://docs.opencv.org/trunk/doc/user\\_guide/user\\_guide.html](http://docs.opencv.org/trunk/doc/user_guide/user_guide.html) (13.12.2013.)
- [6] OpenCV Documentation  
<http://docs.opencv.org/index.html> (13.12.2013.)
- [7] P. Wagner: „Machine Learning with OpenCV2“, veljača 2012.  
<http://www.bytefish.de/pdf/machinelearning.pdf> (08.01.2013.)
- [8] N. Bidyanta: „Pattern Recognition using Principal Component Analysis“, kolovoz 2010.  
<https://sites.google.com/site/binarydigits10/articles/eigenface> (10.01.2013.)