

# Automatska procjena dobi osoba na temelju hoda

---

**Kale, Nika**

**Master's thesis / Diplomski rad**

**2023**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:211:134675>

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

*Download date / Datum preuzimanja:* **2024-05-05**



*Repository / Repozitorij:*

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



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

**Nika Kale**

# **AUTOMATSKA PROCJENA DOBI OSOBA NA TEMELJU HODA**

**DIPLOMSKI RAD**

**Varaždin, 2023.**

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

**Nika Kale**

**Matični broj: 0016133895**

**Studij: Baze podataka i baze znanja**

**AUTOMATSKA PROCJENA DOBI OSOBA NA TEMELJU HODA**

**DIPLOMSKI RAD**

**Mentorica :**

Izv. prof. dr. sc. Petra Grd

**Varaždin, veljača 2023.**

### **Izjava o izvornosti**

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autorica potvrdila prihvaćanjem odredbi u sustavu FOI-radovi*

---

## Sažetak

U ovom radu opisuje se što je to hod kao biometrijska karakteristika, koje su sve značajke hoda i koji su sve načini prepoznavanja hoda. U samom središtu ovog rada je procjena dobi osobe na temelju hoda te analiza različitih primjenjivih metoda. Također, nakon upoznavanja i usvajanja svih bitnih pojmova, zadnje poglavlje sadrži izradu same aplikacije i alate te metode i funkcije koje su bile korištene. Na samom kraju prikazan je izgled aplikacije kroz nekoliko primjera. Svrha ovog rada je približiti pojam hoda kao biometrijske karakteristike kroz zanimljiv primjer izrade aplikacije za procjenu dobi osobe na temelju hoda.

**Ključne riječi:** biometrija; hod; karakteristike; procjena dobi; neuronske mreže; aplikacija; Python

# Sadržaj

|                                                                 |           |
|-----------------------------------------------------------------|-----------|
| <b>1. Uvod</b>                                                  | <b>1</b>  |
| <b>2. Metode i tehnike rada</b>                                 | <b>2</b>  |
| <b>3. Hod kao biometrijska karakteristika</b>                   | <b>3</b>  |
| 3.1. Analiza hoda                                               | 4         |
| 3.2. Značajke biometrije hoda                                   | 6         |
| 3.2.1. Siluete (engl. <i>Silhouettes</i> )                      | 6         |
| 3.2.2. Oblici (engl. <i>Shapes</i> )                            | 7         |
| 3.2.3. Zglobni kutovi (engl. <i>Joint Angles</i> )              | 8         |
| <b>4. Prepoznavanje hoda</b>                                    | <b>9</b>  |
| 4.1. Prepoznavanje hoda na temelju modela                       | 10        |
| 4.2. Prepoznavanje hoda bez modela                              | 13        |
| <b>5. Procjena dobi na temelju biometrijskih karakteristika</b> | <b>16</b> |
| 5.1. Klasifikacija                                              | 17        |
| 5.2. Regresija                                                  | 18        |
| 5.3. Rangiranje                                                 | 19        |
| <b>6. Procjena dobi na temelju hoda</b>                         | <b>21</b> |
| 6.1. Izrada korisničkog sučelja                                 | 21        |
| 6.2. Pred-procesiranje videa                                    | 23        |
| 6.2.1. Gait Energy Image (GEI)                                  | 25        |
| 6.3. Metoda rangiranja-CNN                                      | 26        |
| 6.4. Prikaz finalnog rješenja                                   | 31        |
| <b>7. Zaključak</b>                                             | <b>33</b> |
| <b>Popis literature</b>                                         | <b>36</b> |
| <b>Popis slika</b>                                              | <b>38</b> |
| <b>Popis popis tablica</b>                                      | <b>39</b> |

# 1. Uvod

U suvremeno doba identifikacija osobe pomoću glasa, lica ili hoda je sveprisutna međutim u mnogim primjenama identifikacije uspostavljene biometrije mogu biti nejasne ili nečitljive. Na primjer, lice može biti skriveno ili u niskoj rezoluciji, dlan može biti zasjenjen, a snimka glasa loše kvalitete i nejasna. Međutim, hod se ne može sakriti i upravo to je motivacija za korištenje biometrije i privlači sve veći interes pogotovo jer ne zahtijeva kontakt sa subjektom, kao što je to slučaj s automatskim prepoznavanjem lica i drugim biometrijskim podacima. [1]

Kroz nekoliko godina došlo je do povećanja broja nadzornih sustava, čiji je cilj poboljšanje sigurnosti i zaštite. Ovi sustavi tek trebaju uključiti mogućnosti prepoznavanja, a prepoznavanje hoda moglo bi biti najprikladniji izbor. Psihofiziološke studije su pokazale da ljudi mogu prepoznati prijatelje i spol osobe samo po hodu sa 70-80% točnosti. Također, mnoga istraživanja pokazala su da se hod također može koristiti za otkrivanje identiteta osobe, spola, emocionalnog stanja itd. Prepoznavanje hodom jedna je od najnovijih biometrijskih metoda budući da je njen razvoj započeo tek kada su računalna memorija i brzina obrade postali dovoljni za obradu sekvenci slikovnih podataka s razumnim performansama. Potencijal za prepoznavanje hoda je velik, pa stoga postoji veliki interes za istraživanje računalnog vida u izdvajanju značajki hoda.

Kao meka biometrija, automatska procjena ljudske dobi postala je vrlo važno područje istraživanja jer pruža korisne podatke koji se mogu koristiti u raznim aplikacijama. [2] Sustav za automatsku procjenu dobi može se koristiti za sprječavanje maloljetnika da kupuju alkohol ili cigarete na automatima, može se izvršiti brza intervencija kada je starija osoba u rizičnoj situaciji ili sustav može prilagoditi sadržaj koji se prikazuje korisniku na temelju njegove dobi. Glavni pristup za procjenu ljudske dobi je analizom slika lica, međutim nedavno su pristupi koji se temelje na hodu i ponašanju dobili sve veću pozornost u kontekstu automatske procjene dobi.

## 2. Metode i tehnike rada

Nakon detaljne razrade osnovnih koncepata i upoznavanja s terminologijom u teorijskom dijelu rada, slijedi praktični dio u kojem je opisan proces izrade aplikacije. Praktični dio ovoga rada sastoji se od izrade aplikacije za procjenu dobi, točnije dobne skupine kojoj određeni pojedinac pripada, na temelju hoda. Programski jezik izabran za izradu ove aplikacije je Python zajedno sa slijedećim bitnim bibliotekama:

- **PyQt** - PyQt je jedan od poznatijih „okvira rada“ (engl. Framework) za kreiranje grafičkih korisničkih sučelja.
- **OpenCV** - OpenCV je biblioteka za računalni vid (engl. Computer vision) i strojno učenje (engl. Machine learning) koja sadrži preko 2 500 optimiziranih algoritama.
- **Numpy** - Numpy je osnovna biblioteka za znanstveno računanje u Python-u.
- **TensorFlow** - TensorFlow je open-source biblioteka za strojno učenje i umjetnu inteligenciju.
- **Keras** - Keras je open-source biblioteka koja pruža Python sučelje za umjetne neuronske mreže. Keras djeluje kao sučelje za TensorFlow biblioteku.

### 3. Hod kao biometrijska karakteristika

Hod je aktivno kretanje tijela naizmjeničnim pokretima donjih udova, što rezultira prenošenjem tijela s jednoga mjesta na drugo (lokomocija). Te su kretnje automatske, dok su započinjanje hoda i promjene u hodu (smjer, brzina, zaustavljanje) pod kontrolom volje. [3] Hod možemo definirati i kao stil ili način hodanja gdje se način hodanja kod pojedinca može znatno razlikovati. Na način hoda osobe mogu utjecati spol i konstitucija osobe, životna dob, stanje mišića, uhranjenost, vrsta obuće i slično.

Hod predstavlja složenu prostorno-vremensku biometriju ponašanja, koji iako nije jedinstven za svakog pojedinca, dovoljno je karakterističan da omogući provjeru identiteta. Budući da je hod biometrijska karakteristika ponašanja, on ne mora ostati nepromjenjiv, pogotovo tijekom duljeg vremenskog perioda.

Biometrija hoda nudi niz prednosti koje bi je učinile poželjnim izborom za implementaciju, gdje je potrebno prepoznavanje, procjena ili identifikacija. Takve prednosti uključuju sposobnost otkrivanja na daljinu, mogućnost promatranja na nenametljiv način, činjenicu da se može implementirati korištenjem relativno jeftinih tehnologija i implicitne poteškoće koje bi bile potrebne za reproduciranje hoda druge osobe radi lažiranja biometrijskog sustava. [4] Međutim, iako biometrija hoda ima mnoge prednosti koje druge biometrijske karakteristike nemaju, ona je također podložna brojnim ograničenjima koja su spriječila njezino uključivanje u praktične svrhe. Kao što smo već spomenuli postoji više čimbenika koji mogu utjecati na hod pojedinca, pa tako, između ostalih čimbenika, biometrijski podaci mogu biti prikriveni promjenama u odjeći, ruksacima i vrsti cipela. Također, čimbenici poput stanja promatrane osobe, ozljede ili intoksikacije mogu imati negativan utjecaj na biometriju promatranog hoda.

Studije psihofizike sugeriraju da ljudi mogu prepoznati poznate osobe koristeći se samo njihovim hodom. Takav sustav obično se sastoji od video kamere koja snima slike osobe koja hoda unutar njenog vidnog polja. Odgovarajuće značajke kao što su zglobovi kutovi (engl. *Joint Angles*) ili siluete (engl. *Silhouettes*) izdvojene su iz videa i zatim se koriste za usporedbu s pohranjenim potpisima hoda poznatih pojedinaca. Ako gledamo iz perspektive privatnosti, tehnika prepoznavanja hoda pomoću video snimanja našla se na meti prigovora i kritika. Kako bi se riješila ta ograničenja, metode koje se koriste za biometrijsko snimanje hoda su raznolike i predloženi su noviji pristupi koji ispituju karakteristike (zglobni kutovi, siluete i oblici) za koje je manje vjerojatno da će izazvati prigovore na privatnost. [4] Također je fokus bio na poboljšanju performansi kako bi se ublažili ometajući čimbenici budući da se istraživanje usmjerilo prema proučavanju novih i snažnih tehnika strojnog učenja.

### 3.1. Analiza hoda

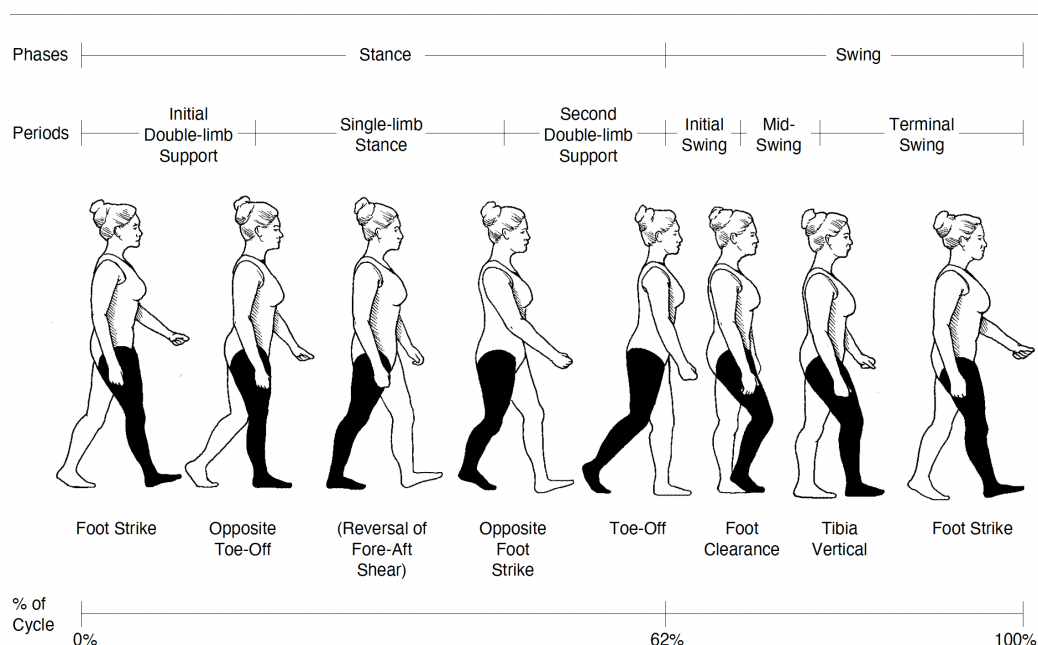
Hodanje se može smatrati periodičnim signalom, odnosno možemo reći da je hod "potpuni ciklus hodanja". Ciklus hoda je vremenski interval između dva uzastopna slučaja inicijalnog kontakta pete s podlogom do ponovnog kontakta s podlogom za isto stopalo. [1]

**Definicija 3.1** Duljina koračanja je linearna udaljenost u ravnini napredovanja između uzastopnih dodirnih točaka istog stopala.

**Definicija 3.2** Duljina koraka je udaljenost između uzastopnih dodirnih točaka suprotnih stopala.

**Definicija 3.3** Korak je kretanje između uzastopnih udaraca petom suprotnih stopala; kompletan ciklus hoda sastoji se od dva koraka.

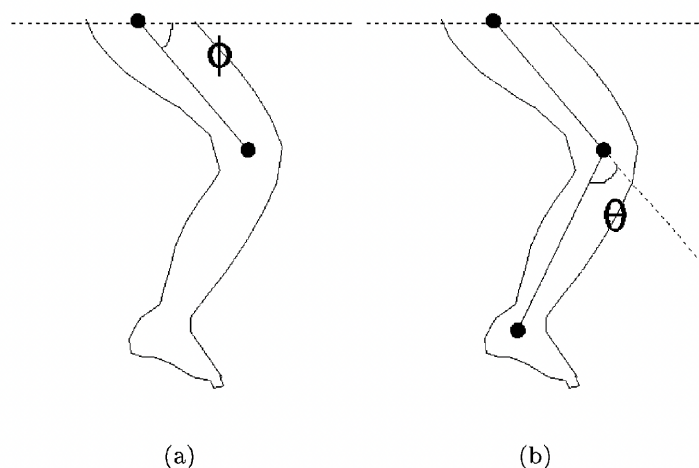
Na slici 1 možemo vidjeti da svaka noga ima dva različita razdoblja; faza stava, kada je stopalo u kontaktu s podom, obično traje oko 62% ciklusa i faza zamaha, kada je stopalo odmaknuto od poda i kreće se naprijed na sljedeći korak, traje posljednjih 38% ciklusa. Faza stava označava početak novog ciklusa kada peta jedne noge udari u pod. Tada se gležanj savija kako bi stopalo bilo ravno na podu, a težina tijela se prenosi na njega. Druga noga zamahuje naprijed dok se peta podiže od tla, tada se težina tijela pomiče na drugo stopalo i potpuno koljeno se savija. Ostatak stopala, koji je sada straga, odiže se od tla čime završava faza stava. [1]



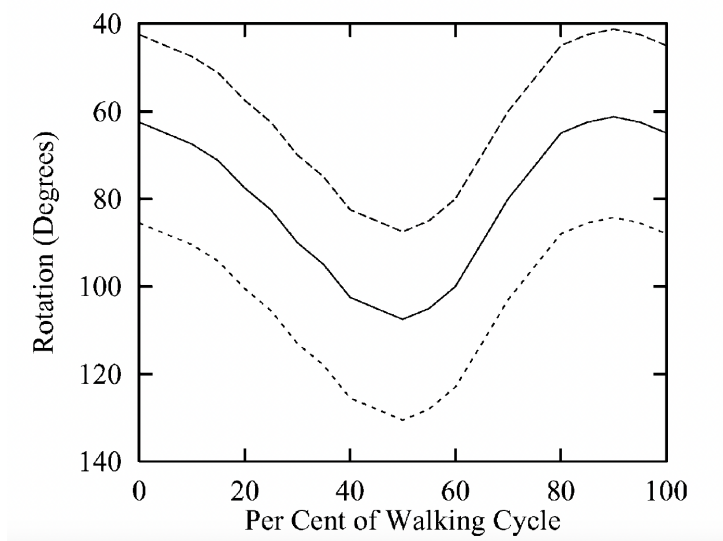
Slika 1: Potpuni ciklus hodanja (Izvor: Chambers i Sutherland 2002)

Početak faze zamaha je kada prsti stopala napuste tlo. Težina se prenosi na drugu nogu i noga se zamahuje prema naprijed kako bi stala pored drugog stopala. Ciklus hoda završava udarcem pete stopala.[1]

Ako uzmemo u obzir sve pokrete u ciklusu hoda, možemo zaključiti da je hod kao takav jedinstven kod svakog pojedinca. Uz navedene pokrete, postoji oko 20 različitih komponenti hoda, od kojih se neke mogu izmjeriti samo iz pogleda na subjekt odozgo. Budući da mnoge komponente koje su utvrđene medicinskim studijama nisu prikladne za sustav koji se temelji na računalnom vidu, komponente su ograničene na obrasce rotacije kuka i koljena. Slika 2 prikazuje kutove rotacije za kuk i koljeno. Normalan obrazac rotacije kuka karakterizira jedno razdoblje ekstenzije i jedno razdoblje fleksije u svakom ciklusu hoda. [6]



Slika 2: Kutovi rotacije kukova (a) i koljena (b). (Izvor: Cunado, Nixon i N.Carter 2003)



Slika 3: Srednji uzorak rotacije kukova. (Izvor: Cunado, Nixon i N.Carter 2003)

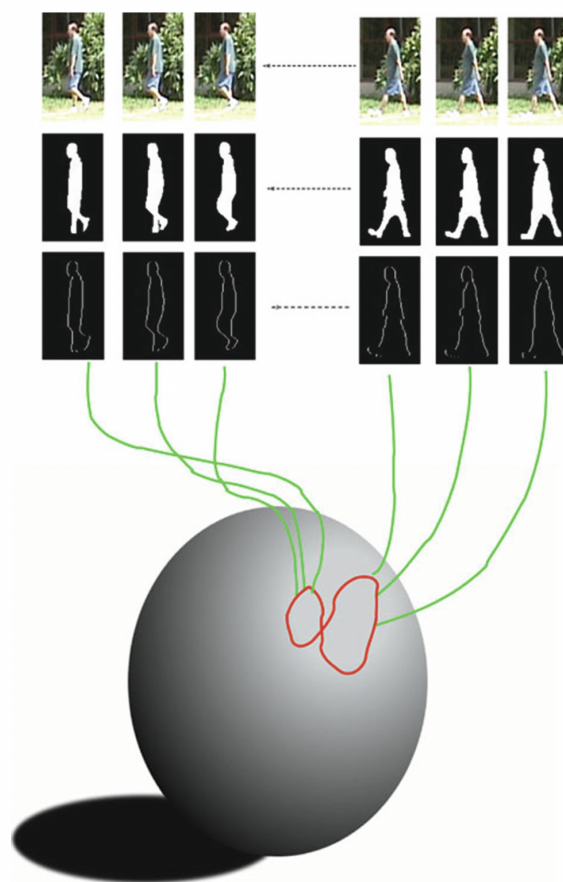
Slika 3 daje prosječni uzorak rotacije, generiran iz baze podataka od 60 subjekata. Gornja i donja točkasta linija označavaju standardnu devijaciju od srednje vrijednosti. U prvoj polovici ciklusa hoda, kuk je u kontinuiranoj ekstenziji dok se trup pomiče prema naprijed preko uda koji ga podupire. U drugoj fazi ciklusa, nakon što se težina prebaci na drugi ud, kuk se počinje savijati u pripremi za fazu zamaha. Ova radnja savijanja ubrzava kuk tako da usmjerava pokretni ud prema naprijed za sljedeći korak. Kut rotacije mjeri se kao kut između linije koja spaja kuk i koljeno i linije koja prolazi kroz točku kuka paralelno s tlom. [6]

## 3.2. Značajke biometrije hoda

Diskriminativne informacije u hodu prisutne su kako u obliku pojedinca tako i u načinu njegova hoda. To znači da biometrijski sustavi koji se temelje na hodu moraju moći modelirati hod kao vremenski niz značajki ili kao dinamički model kako bi izvršili točno prepoznavanje. [7] Također, potrebno je vremenski uskladiti dvije sekvence tako da su kritični događaji tijekom hodanja koji se događaju za vrijeme faze stava ili faze zamaha vremenski točno usklađeni. Potrebno je omogućiti da na izvedbu prepoznavanja značajki ne utječe netočno vremensko usklađivanje između položaja koji se javljaju tijekom hodanja.

### 3.2.1. Siluete (engl. *Silhouettes*)

U većini biometrijskih sustava koji se temelje na hodu, može se pretpostaviti da su kamere statične tijekom kratkog vremenskog razdoblja u kojem hvataju hod pojedinca radi provjere. To omogućuje izradu jednostavnih pozadinskih modela za svaku od ovih kamera. Oduzimanje pozadine zatim identificira skup svih piksela na slici koji pripadaju pojedincu koji se kreće. [7] Na slici 4, u srednjem redu, možemo vidjeti binarne slike oduzete od pozadine u kojima su svi pikseli koji pripadaju pojedincu bijeli, dok je pozadina crna. Ta se binarna slika zatim skalira na jedinstvenu veličinu tako da je izdvojena značajka, odnosno silueta, neovisna o udaljenosti kamere od subjekta.



Slika 4: Značajke silueta i oblik. (Izvor: Li i Jain 2009)

### 3.2.2. Oblici (engl. *Shapes*)

Oblik su sve geometrijske informacije koje ostaju kada se lokacija, mjerilo i rotacijski efekti filtriraju iz objekta. [7] Kendall-ov statistički oblik je rijedak deskriptor oblika koji opisuje konfiguraciju oblika  $k$  orijentiranih točaka u  $m$ -dimenzionalnom prostoru kao  $k \times m$  matrica koja sadrži koordinate orijentiranih točaka. Prostor slike je dvodimenzionalan i stoga je prikladno opisati vektor oblika kao  $k$ -dimenzionalni kompleksni vektor. [7]

Prvi korak nam je dobiti binariziranu siluetu kao što je opisano u prethodnom poglavlju. Ona označava opseg objekta koji se nalazi na slici pomoću kojeg se zatim izdvaja značajka oblika. Ovaj vektor značajki mora biti nepromjenjiv u odnosu na prevođenje i skaliranje jer identitet objekta ne bi trebao ovisiti o udaljenosti objekta od kamere. Dakle, bilo koji vektor značajke koji dobijemo mora biti nepromjenjiv na translaciju i razmjer. Ovime dobivamo pred-oblik objekta u svakom okviru. Pred-oblik možemo definirati kao geometrijski podatak koji ostaje kada se filtriraju učinci lokacije i razmjera.

Neka je konfiguracija skupa od  $k$  orijentiranih točaka dana  $k$ -dimenzionalnim kompleksnim vektorom koji sadrži položaje orijentiranih točaka. Označimo ovu konfiguraciju kao  $X$ . Centrirani pred-oblik dobiva se oduzimanjem srednje vrijednosti od konfiguracije i zatim skaliranjem na normu jedan. [7]

Centrirani pred-oblik je dan pomoću:

$$Z_c = \frac{CX}{\|CX\|}, \quad C = I_k - \frac{1}{k} \mathbf{1}_k \mathbf{1}_k^T$$

gdje je

$$I_k \quad k \times k$$

matrica identiteta, a

$$\mathbf{1}_k$$

je  $k$ -dimenzionalni vektor jedinica. [7]

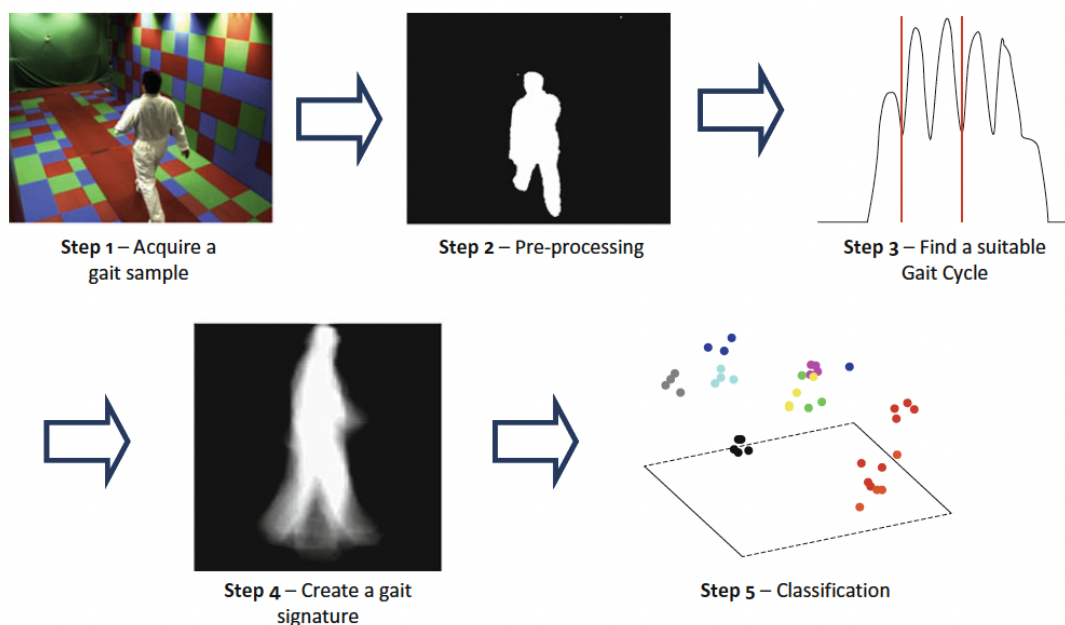
Glavna prednost koju ima korištenje značajke oblika je u tome što se diferencijalna geometrijska svojstva sferne mnogostrukosti u kojoj leže oblici vrlo dobro razumiju, i stoga su odgovarajuće mjere udaljenosti koje mogu objasniti translacijsku, rotacijsku i invarijantnost mjera dobro definirane. Ovu izdvojenu značajku oblika možemo vidjeti na slici 4 u zadnjem redu slike zajedno s grafičkom ilustracijom sferne mnogostrukosti u kojoj leže oblici. Općenito, oblik je vrlo popularna značajka za biometriju koja se temelji na hodu zajedno sa značajkom siluete te nekoliko najsuvremenijih algoritama koristi upravo te značajke kod identifikacije/procjene/-prepoznavanja.

### **3.2.3. Zglobni kutovi (engl. *Joint Angles*)**

U prethodnim poglavljima opisali smo značajke oblika i siluete koje su vrlo bitne kod biometrije hoda. Još jedna vrlo bitna značajka za analizu hoda u medicinskoj i psihofizičkoj zajednici su zglobni kutovi. To su kutovi sklopljeni na svakom od zglobova udova kao što su koljeno, lakat, gležanj, ručni zglob, itd. Prednost korištenja zglobnih kutova kao značajke je činjenica da se automatski postiže nepromjenjivost prikaza dok se koriste zglobni kutovi kao značajke. Unatoč tome, ključni problem s korištenjem zglobnih kutova je činjenica da ih je vrlo teško robusno procijeniti iz nekontroliranih monokularnih video sekvenci. [7]

## 4. Prepoznavanje hoda

Prepoznavanje hoda odnosi se na automatizirane metode koje koriste video ili druge senzorske podatke ljudskog hoda kako bi prepoznale ili identificirale osobu. [7] Na slici 5 prikazan je primjer nekih od osnovnih koraka u sustavu za prepoznavanje hoda.



Slika 5: Opći koraci sustava za prepoznavanje hoda. (Izvor: Matovski, Nixon i Carter 2014)

Prvi korak zahtjeva je prikupljanje podataka pomoću jedne ili više kamera. Ako koristimo samo jednu kameru, prepoznavanje se može provesti pomoću 2D potpisa hoda kao što je Gait Energy Imae (GEI). Međutim, ukoliko koristimo višestruko sinkronizirane kamere broj mogućnosti je veći. Na primjer, jedna od mogućnosti je izrada 3D modela hoda i njegova upotreba za prepoznavanje.

Pred-procesiranje, odnosno oduzimanje pozadine ili segmentacija pozadine, je drugi korak prepoznavanja hoda. Oduzimanjem pozadine može se izmjeriti prirodna scena iza subjekta koji hoda.

Kao što smo već rekli, ljudski hod je periodičan pa se hod može sastojati od više ciklusa hoda. U trećem koraku nastoji se odrediti najprikladniji ciklus kako bi se dovelo do boljih stopa prepoznavanja.

U četvrtom koraku potrebno je pomoću odgovarajuće metode napraviti potpis hoda.

Peti korak je klasifikacija podataka koja se može primijeniti direktno na potpis hoda ili se iz potpisa hoda mogu izdvojiti značajke preko kojih se zatim radi klasifikacija. Postoji izbor tehnika klasifikacije, međutim u najjednostavnijem slučaju može se koristiti klasifikator kao što je k-najbliži susjed (koristeći euklidsku udaljenost).

Pristupi prepoznavanja hoda mogu se podijeliti u dvije glavne skupine: na pristup prepoznavanja na temelju modela i pristup prepoznavanja bez modela.

Tablica 1: Pristupi prepoznavanju hoda

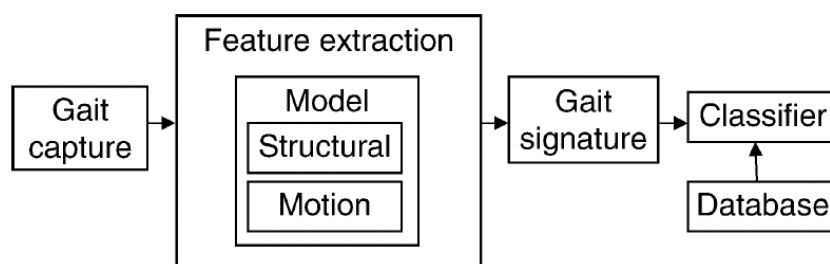
| Bez modela                                                                                                                                       |                                                                                                                                                     | Na temelju modela                                    |                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|-------------------------------------------------------------|
| Oblik u pokretu                                                                                                                                  | Oblik + pokret                                                                                                                                      | Strukturalni                                         | Modelirani                                                  |
| Neumotana silueta; sličnost siluete; relacijska statistika; analiza ključnih okvira; područje; simetrija; modeli raspodjele točaka; ključne poze | Nizovi vlastitog prostora; skriveni Markovljev model; prosječna silueta; trenuci; elipsoidna uklopavanja; kinematičke značajke; stil hoda i sadržaj | Parametri koraka; ljudski parametri; zglobne putanje | Zglobni model; dvojni oscilator; putanje povezanih značajki |

(Izvor: Matovski, Nixon i Carter, 2014)

Pristupi temeljeni na modelu koriste strukturu ljudskog tijela, a pristupi bez modela koriste cijeli obrazac kretanja ljudskog tijela. Koji od pristupa će biti korišten ovisi o uvjetima stjecanja podataka. Pristupi bez modela (temeljeni na izgledu) koriste ulazne slike izravno za kreiranje potpisa hoda bez prilagođavanja modelu. Ovi pristupi mogu izvršiti prepoznavanje pri nižim rezolucijama što ih čini prikladnima za vanjske primjene, gdje subjekt može biti na velikoj udaljenosti od kamere. Pristupi koji se temelje na modelu obično zahtijevaju slike subjekta veće rezolucije kako bi se mogle točno uklopiti u model. [8] U idućim poglavljima ćemo detaljno opisati svaki od navedenih pristupa.

## 4.1. Prepoznavanje hoda na temelju modela

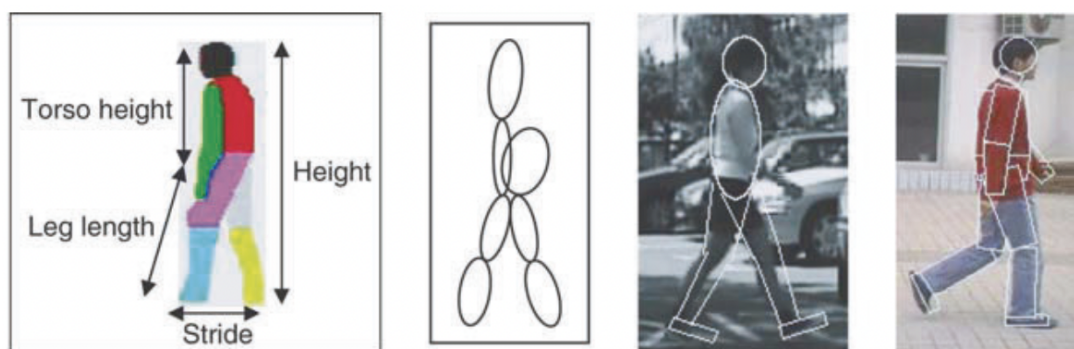
Prepoznavanje hoda temeljeno na modelu odnosi se na identifikaciju pomoću temeljnih matematičkih konstrukta koji predstavljaju diskriminirajuće karakteristike hoda (bilo da su statične ili dinamičke), sa skupom parametara i skupom logičkih i kvantitativnih odnosa između njih. [7] Ovakav sustav se najčešće sastoji od snimanja hoda, izdvajanja značajki, potpisa hoda i klasifikacije (slika 6).



Slika 6: Komponente sustava za prepoznavanje hoda temeljenog na modelu. (Izvor: Li i Jain 2009)

Praktična pitanja koja izazivaju pristup temeljen na modelu mogu se podijeliti u dvije kategorije, koje se odnose na sustav i na osobu. Jedan od izazova povezanih sa sustavima je nepromjenjivost gledišta, dok izazovi povezani s osobom uključuju učinke fizioloških promjena (kao što je starenje, dosljednost hodanja/upisanog u različito vrijeme, mijenja li se naš obrazac hodanja tijekom duljeg vremenskog razdoblja), psihološke promjene (raspoloženje) i vanjski čimbenici (opterećenje, obuća i fizičko okruženje). [7]

U tipičnom pristupu koji se temelji na modelu, često su potrebni strukturalni model i model kretanja koji služe kao osnova za praćenje i izdvajanje značajki (pokretnog čovjeka). Model može biti 2-dimenzionalni ili 3-dimenzionalni, iako je većina trenutnih pristupa 2-dimenzionalna i pokazala je sposobnost postizanja obećavajućih rezultata prepoznavanja na velikim bazama podataka sa preko 100 subjekata. Strukturni model opisuje topologiju ili oblik dijelova ljudskog tijela kao što su glava, torzo, kuk, bedro, koljeno i gležanj pomoću mjera kao što su duljina, širina i položaj. Ovaj model se najčešće sastavlja od primitivnih oblika kao što su cilindar, stožac i štapići ili proizvoljnih oblika koji opisuju rubove tih dijelova tijela. Na slici 7 može se vidjeti nekoliko primjera parametara tijela koji se koriste u strukturnim modelima.



Slika 7: Primjer parametara tijela za strukturni model. (Izvor: Li i Jain 2009)

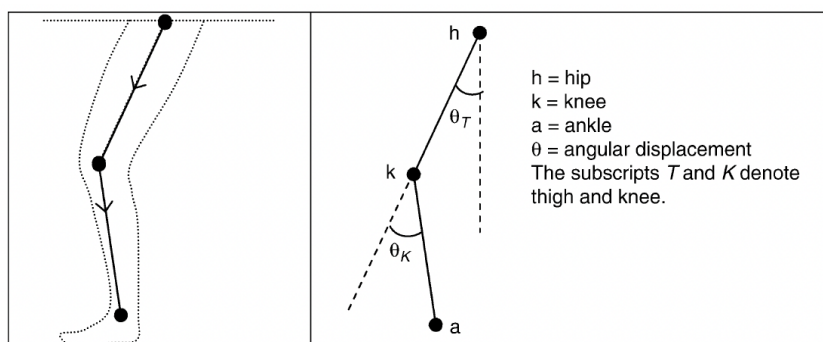
Drugi model koji se može koristiti u pristupu temeljenom na modelu je model kretanja koji opisuje kinematiku ili dinamiku gibanja svakog dijela tijela. Kinematika općenito opisuje kako subjekt mijenja položaj s vremenom ne uzimajući u obzir učinak mase i sile, dok dinamika objašnjava sile koje djeluju na te mase tijela i rezultirajuće gibanje. [7] Prilikom razvijanja modela kretanja potrebno je razumjeti ograničenja hoda kao što su ovisnost o susjednim zglobovima i ograničenje kretanja u smislu opsega i smjera.

U poglavlju o analizi hoda spominjali smo rotaciju kuka i koljena i kako to utječe na jedinstvenost hoda. Cunado, Nixon i Carter [6] opisali su kutno gibanje kuka i koljena Fourier-ovim redom tako da su koristili jednostavan strukturni model kakav je prikazan na slici 7. Iako je model kretanja napravljen za jednu nogu, pretpostavkom da je hod simetričan, drugu nogu možemo modelirati na sličan način s faznim pomakom od pola perioda.

Cunado, Nixon i Carter [6] modelirali su kutno kretanje koljena sa sljedećom jednačbom:

$$\theta = a_0 + 2 \sum_{k=1}^N [b_k \cos k\omega_0 t - c_k \sin k\omega_0 t]$$

gdje je  $N$  ukupan broj frekvencija,  $\omega_0$  osnovna frekvencija, a  $a_0$  pomak (slika 8). Potpis hoda zatim dobijemo množenjem komponente faze i magnitude Fourier-ovog opisa.



Slika 8: Strukturni model noge. (Izvor: Li i Jain 2009)

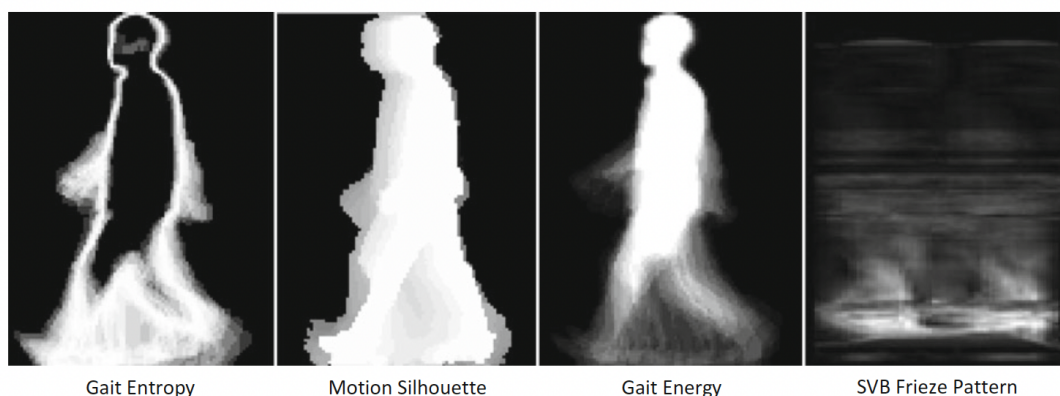
Bitna stavka u pristupu koji se temelji na modelu je ekstrakcija značajki. Time segmentiramo zanimljive dijelove tijela i izdvajamo statičke i/ili dinamičke karakteristike hoda. Inicijalizacijom modela, segmentacijom i praćenjem (procjenom) kretanja čovjeka od jedne slike do druge izdvajamo važne prostorne, vremenske ili prostorno-vremenske signale iz hoda. Osim značajki, vrlo bitno je moći pronaći odgovarajuću sekvencu ciklusa hoda kao što smo već to ranije spominjali. Na primjer, metodom graničnog okvira možemo lako doći do faze stava budući da u fazi stava širina okvira ima najveću vrijednost. Od ostalih metoda koje možemo koristiti tu su još i brojanje piksela ljudske figure, korištenjem binarne maske (slika 9), aproksimacijom vanjskog područja zamaha noge i slično. [7]



Slika 9: Binarna maska ciklusa hoda. (Izvor: Li i Jain 2009)

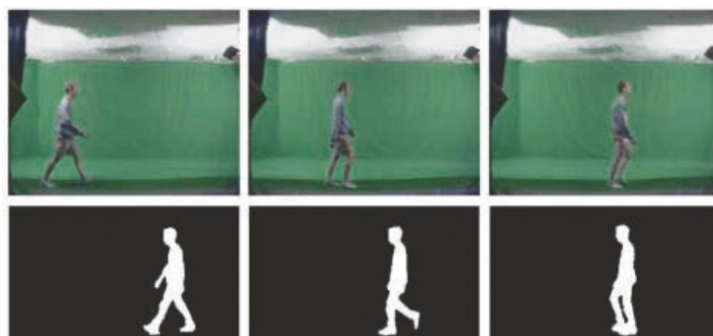
## 4.2. Prepoznavanje hoda bez modela

Prepoznavanje hoda bez modela izdvaja oblik subjekta koji hoda, a zatim izračunava vektor značajki koji opisuje ili obrazac kretanja u hodu ili fizičke dimenzije subjekta. Klasifikator zatim uspoređuje vektor značajki s prethodno dobivenim primjerima za identifikaciju ili provjeru. [7] U nekim literaturama može se pronaći i naziv prepoznavanje na temelju siluete. Postoji više pristupa kojima se može izdvojiti oblik subjekta, a najjednostavniji pristup je Gait Energy Image (GEI) koji formira prosjek silueta tijekom cijelog ciklusa hoda. Iako je GEI jednostavno izračunati, ovaj pristup je vrlo osjetljiv na promjene ljudske siluete. [8] Ostali pristupi, kao i GEI, su prikazani na slici 10.



Slika 10: Primjeri potpisa hoda bez modela. (Izvor: Matovski, Nixon i Carter 2014)

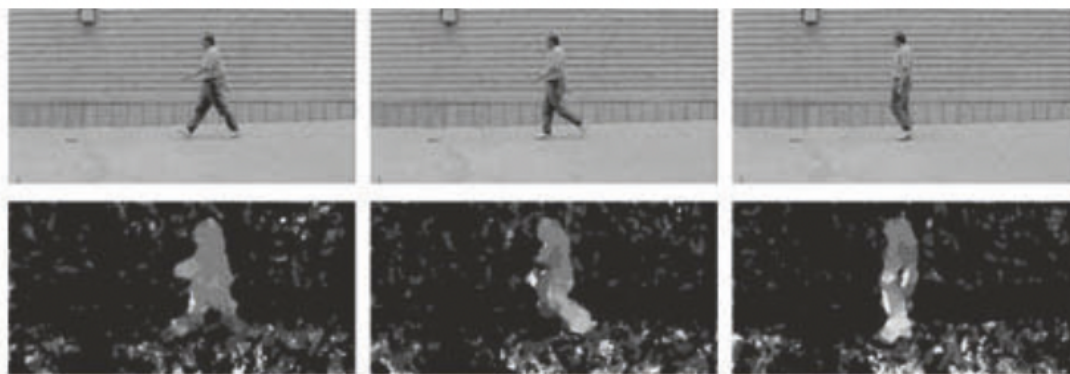
Definicije siluete često znaju biti dvosmislene. Neki siluetu definiraju kao područje koju pokriva figura, dok drugi tvrde da je silueta granica između figure i njezine pozadine. Ako gledamo iz perspektive biometrije hoda, pretpostavit ćemo da se silueta odnosi na područje, a ne na granicu. Kako bi uopće mogli raditi sa siluetom nekog subjekta koji hoda, potrebno je segmentirati piksele slike u prednji plan (subjekta koji se kreće) i skupove piksela pozadine (sve ostalo). Najlakši način za dobivanje pouzdane siluete je "chroma-keying" koji se oslanja na razlike u boji između pozadine i subjekta u prvom planu. Boja pozadine (obično zelena ili plava) odabire se kako bi razlikovanje boja bilo robusnije (slika 11). [7]



Slika 11: Primjer "chroma-keyinga" u analizi hoda. (Izvor: Li i Jain 2009)

Iako se "chroma-keying" smatra najlakšim načinom za dobivanje pouzdane siluete, ovom metodom tj. neobičnom bojom pozadine dajemo subjektu do znanja da je po nadzorom što poništava cijelu "tajnost" biometrije. Druga metoda koju koristimo za dobivanje siluete je metoda oduzimanja pozadine. Oduzimanjem pozadine izbjegavamo potrebu za pozadinom u boji mjerenjem prirodne scene iza subjekta. To podrazumijeva procjenu statističkih svojstava (obično u svjetlini i boji) svakog piksela u jednom ili više okvira videa. Uspoređujući procjenu pozadine sa sljedećim okvirima videa, moguće je klasificirati piksele prednjeg plana kao one koji ne odgovaraju pozadini. [7] U većini slučajeva, procjena i oduzimanje pozadine spajaju se u mrežni sustav koji kontinuirano izračunava razlike u pikselima i zatim ažurira pozadinu za svaki okvir videa. Ova metoda zahtjeva da pozadina i kamera budu nepomični, što bi značilo da je ta metoda idealna u okolnostima gdje se nepomično snima samo jedan te isti kadar.

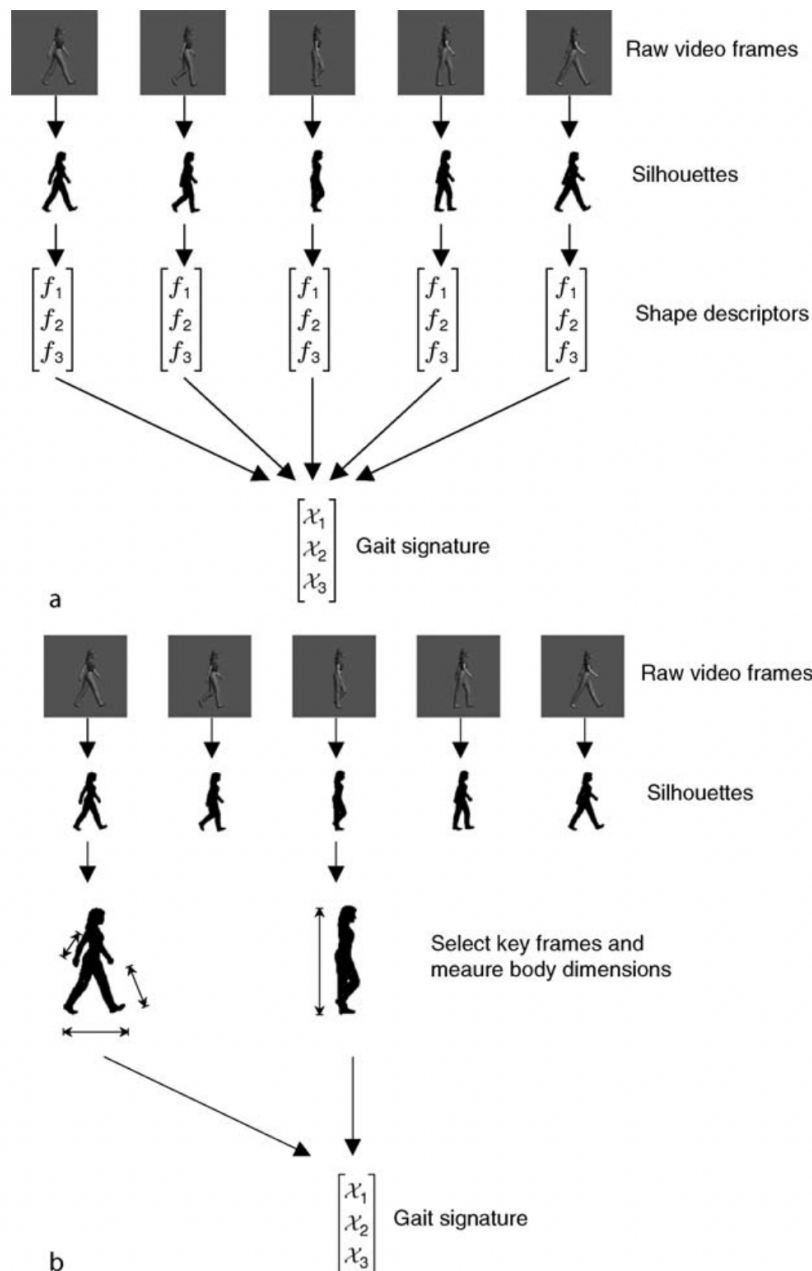
Projekcija gibanja u sceni na ravninu slike kamere naziva se polje gibanja/kretanja. Kada subjekt hoda, segmentacija koja se kreće od sporih ili stacionarnih piksela u polju kretanja će izdvojiti siluetu figure. [7] Ovom metodom nam polje kretanja, osim da pokazuje samo subjekt koji se kreće, pokazuje i koliko se brzo kreću određeni dijelovi tijela. Iako se polje kretanja kao takvo ne može izračunati, može se izračunati aproksimacija polja kretanja, odnosno optički protok, koja je dovoljna za biometrijsko prepoznavanje hoda. Optički protok je djelomično povezan s poljem kretanja, ali nije nužno jednako polju kretanja u svim slučajevima (slika 12).



Slika 12: Primjer optičkog protoka u analizi hoda. (Izvor: Li i Jain 2009)

Kada pogledamo što je sve zajedničko slikama 11, 12, pa čak i 10 možemo vidjeti da je pozicija kamere uvijek iz profila subjekta tj, subjekt uvijek hoda preko vidnog polja, a ne prema ili od kamere. Ovakav pristup snimanja i prikupljanja podataka postoji jer pogledi subjekta sprijeda ili straga pokazuju uglavnom kretanje s jedne na drugu stranu i ne otkrivaju položaj zglobova niti složene obrasce pokreta udova. Snimanje kretanja temeljeno na markerima, npr. Johanssonovi pokretni svjetlosni prikazi, nudi kontrapunkt siluetama koje su manje praktične za biometriju, ali su korisne za dobivanje uvida u perceptivne probleme koji okružuju hod. [7]

Još jedan neophodan korak u prepoznavanju hoda temeljenog na siluetama je pretvorba vremenskog niza silueta u potpis hoda, tj. vektor značajki prikladan za klasifikaciju. Jedan pristup je izdvajanje značajki koje karakteriziraju oblike silueta i njihove varijacije tijekom vremena, kao što je shematski prikazano na slici 13(a).

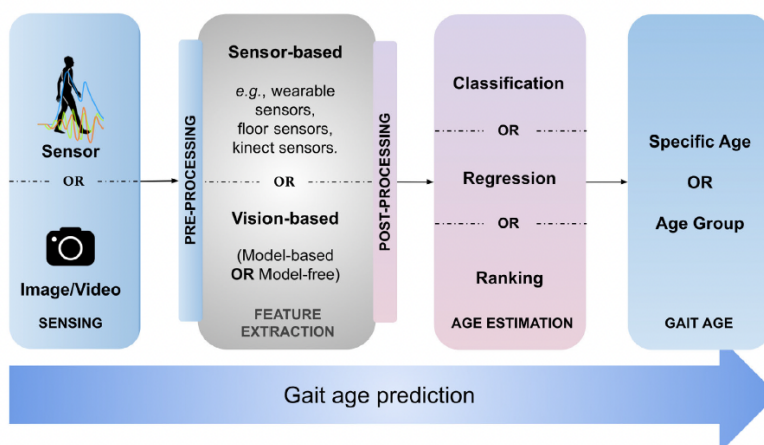


Slika 13: Primjeri procesa dobivanja potpisa hoda. (Izvor: Li i Jain 2009)

Drugi pristup je da se, umjesto oslanjanja na vezu između hoda i strukture tijela za formiranje potpisa hoda, mogu se koristiti vektori značajki koji se izravno odnose na dimenzije tijela kao što je prikazano na slici 13(b). Na primjer, značajke kao što su visina, širina i druge proporcije tijela te karakteristike povezane s kretanjem kao što su duljina koraka i količina zamaha rukama.

## 5. Procjena dobi na temelju biometrijskih karakteristika

Starenje je nepovratan proces, ljudsko lice i hod mijenjaju se s vremenom što značajno utječe na izgled osobe. Prepoznati koliko godina određena osoba ima često je veliki izazov. Upravo se iz tog razloga u istraživačkoj zajednici računalnog vida i prepoznavanja uzoraka veliki interes pridaje automatskoj procjeni dobi. U posljednjih nekoliko godina predloženi su brojni pristupi za proučavanje problema prepoznavanja dobnih skupina na temelju RGB slika. Postoje dvije kategorije pristupa: Cranio-Facial pristupi i bihevioralni pristupi. [9] Iako su bihevioralni pristupi vrlo obećavajući i daju dobre performanse kada je osoba daleko od kamere, lice je i dalje područje koje se najviše mijenja starenjem. Posljednjih godina primijenjene su mnoge metode za automatsku procjenu dobi osobe. U okviru procjene dobi razlikujemo tri pristupa: regresiju, klasifikaciju i duboko učenje(rangiranje) (slika 14).



Slika 14: Proces procjene dobi. (Izvor: Aderinola, Connie, Ong i dr. 2021)

Modeli klasifikacije uključuju  $k$  najbližih susjeda (engl. *k Nearest Neighbors*), višeslojne perceptrone (engl. *Multilayer Perceptrons*) i najčešće korištene strojeve potpornih vektora (SVM) (engl. *Support Vector Machines*). Za metode regresije u literaturama najviše se spominju kvadratna regresija (engl. *Quadratic Regression*), regresija potpornoga vektora (SVR) (engl. *Support Vector Regression*) i regresor s više instanci (engl. *multi-instance regressor*). U novije vrijeme, tehnike dubokog učenja kao što su konvolucijske neuronske mreže (CNN) (engl. *Convolutional Neural Networks*) primijenjene su na procjenu ljudske dobi kako bi se naučile značajke starenja izravno iz velikih podataka o određenim biometrijskim karakteristikama kao što su hod i lice. [11] Rezultati mnogih istraživanja pokazuju da obrasci starenja, koji su dobiveni pomoću dubokog učenja, dovode do značajnog poboljšanja performansi na odabranim skupovima podataka kao i na fotografijama. Iako su dva tri pristupa generalno dobra za korištenje, klasifikacija u potpunosti zanemaruje redne informacije u dobnim oznakama, a regresija je previše pojednostavljuje na linearni model, dok je obrazac ljudskog starenja općenito nelinearan. Općenito, kada ljudi predviđaju nečiju dob, obično je lakše odrediti je li osoba starija od određene dobi nego izravno dati točnu dob. [11]

## 5.1. Klasifikacija

Trenutno najpopularnija i najviše korištena metoda klasifikacije je SVM metoda (*engl. Support Vector Machines*). SVM je algoritam za nadzirano učenje koji se temelji na ideji pronalaženja hiperravnine koja najbolje odvaja značajke u različite domene. [12] Točke koje su najbliže hiperravnini nazivaju se točkama nosivih vektora, a udaljenosti vektora od hiperravnine nazivaju se rubovima. Što su SV točke udaljenije od hiperravnine, to je veća vjerojatnost ispravnog klasificiranja točaka u njihovoj odgovarajućoj regiji ili klasama. SV točke su vrlo kritične u određivanju hiperravnine jer ako se položaj vektora promijeni, mijenja se položaj hiperravnine. [13] U primjeru procjene dobi na temelju biometrijskih karakteristika SVM se koristi tako da analizira numerička svojstva različitih značajki slika i zatim organizira podatke u kategorije.

Jedan od razloga zašto je SVM tako popularan je trik s jezgrom (*engl. the Kernel Trick*). To je način izračunavanja točkastog produkta dvaju vektora  $x$  i  $y$  u nekom (vrlo visokodimenzionalnom) prostoru značajki, zbog čega se funkcije jezgre ponekad nazivaju "generalizirani točkasti produkt". Primjena trika jezgre znači samo zamijeniti točkasti umnožak dva vektora funkcijom jezgre. [13] Postoji više vrsta jezgre, međutim nama najinteresantnija je Gaussova jezgra. To je funkcija čija vrijednost ovisi o udaljenosti od ishodišta ili neke točke, a definiramo ju na sljedeći način:  $K(X_1, X_2) = e^{-\gamma \|X_1 - X_2\|^2}$ , gdje je  $\|X_1 - X_2\|$  Euklidska udaljenost između  $X_1$  i  $X_2$ . Koristeći udaljenost u izvornom prostoru izračunavamo točkasti produkt (sličnost)  $X_1$  i  $X_2$ . [13] Na slici 15 prikazan je način korištenja SVM algoritma, odnosno Gaussove jezgre.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Algorithm 2: Radial Basis Function (RBF) Kernel</b></p> <p><b>Input:</b> Feature vector for all of training images, feature vector for Testing images</p> <p>M: n-classes, attributes</p> <p><b>Output:</b> Class test</p> <p><b>Begin</b></p> <p><b>Step 1:</b> Read feature vector for all of the training images, with M attributes and feature vector for all Testing images.</p> <p><b>Step 2:</b> Obtain the weight <math>w</math> and kernel <math>K</math> by using the classical adaptive scaling SVM.</p> <p><b>Step 3:</b> Now, initialize entire data into a dataset.</p> <p><b>Step 4:</b> To rank the features use Random Forest Algorithm.</p> <p><b>Step 5:</b> Remove the less important features from the dataset and update it so that the size of the dataset can be reduced.</p> <p><b>Step 6:</b> Apply SVM with RBF kernel in order to reduce features.</p> <p>Minimize <math>(\frac{1}{2} \ w\ ^2 + C \sum_{i=1}^m \xi_i)</math></p> <p>Subject to: <math>Y_i(W^T x_i + b) \geq 1 - \xi_i</math>, <math>\xi_i \geq 0</math>, <math>i = 1, 2, \dots, m</math></p> <p>Where <math>Y_i</math> is the class label of support vector <math>x_i</math>,</p> <p><math>w</math> is a weight vector,</p> <p><math>b</math> is bias and variables</p> <p><math>\xi</math> is positive slack which is necessary to allow miss classification</p> <p><b>Step 7:</b> Consider that parameter <math>C</math> seeks to decision error when searching for the maximum marginal hyper plane.</p> <p>Maximize <math>[\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \alpha_i y_i y_j K(x_i, x_j)]</math></p> <p>Subject to: <math>0 \leq \alpha_i \leq C</math>, <math>1 \leq i \leq l</math>, <math>\sum_{i=1}^l \alpha_i y_i = 0</math>, where indicates the overall inputs. Where <math>x</math> is a training sample and <math>k</math> is the radial basis function is given by <math>K(x_i, x_j) = \exp(-\gamma \ x_j - x_i\ ^2)</math></p> <p><b>Step 8:</b> Choose the best parameter for create as a classification model.</p> <p><b>End</b></p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Slika 15: Primjer SVM korištenja (Izvor: Madhavi, Bhuvana Sree, Shriya i dr. 2021)

## 5.2. Regresija

Metoda kvadratne regresije (*engl. Quadratic Regression*) je metoda regresije koja je najviše zastupljena. Ako uzmemo u obzir izdvojene značajke za, na primjer, svaku sliku lica, regresijska funkcija se često koristi za karakterizaciju odnosa između izdvojenih značajki  $y$  i dobnih oznaka  $L$ ,  $L = f(y)$ . [14]

Funkciju kvadratne regresije možemo definirati kao

$$\hat{L} = \omega_0 + w_1^T y + w_2^T y^2$$

gdje je  $\hat{L}$  procjena starosti,  $\omega_0$  je pomak,  $y$  i  $y^2$  su ekstrahirani vektor obilježja i njegov kvadrat, a  $w_1$  i  $w_2$  su težinski vektori. Parametre modela može se optimizirati tako što se minimizira razlika između stvarne dobi pojedinaca,  $L$  i dobi procijenjene korištenjem funkcije  $\hat{L}$  tj.  $\|L - \hat{L}\|$ . [14]

Međutim, iako metoda kvadratne regresije ima mnoge prednosti, kod primjene za procjenu dobi ima i nekoliko mana:

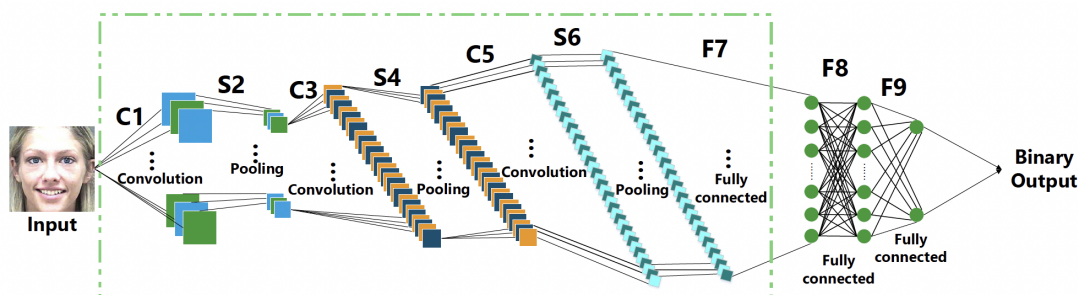
- Starenje je složen problem nelinearne regresije, posebno za veliki raspon godina, npr. 0-90. Jednostavna kvadratna funkcija možda neće ispravno modelirati složeni proces starenja.
- Procjena najmanjeg kvadrata osjetljiva je na odstupanja koja dolaze od netočnih oznaka u prikupljanju velike baze podataka slika.
- Kriterij procjene najmanjeg kvadrata samo minimizira empirijski rizik koji se možda neće dobro generalizirati za neviđene primjere, posebno s malim brojem primjera koji se treniraju. To je tipično za procjenu dobi zbog poteškoća u prikupljanju slika dobi i raznolikosti uzoraka dobi zbog različitih životnih uvjeta, kozmetike, spolnih razlika i oblika lica. [14]

Upravo zbog navedenih mana, umjesto metode kvadratne funkcije, usvaja se metoda regresije potpornih vektora, SVR (*engl. Support Vector Regression*).

Osnovna ideja SVR-a je pronaći funkciju  $f(y)$  koja ima najveće odstupanje  $\varepsilon$  od stvarno dobivenoga cilja  $z_i$  za podatke o treniranju  $y_i$ , a istovremeno je što je moguće ravnija. Drugim riječima, ne zanimaju nas pogreške sve dok su manje od  $\varepsilon$ . Ovo svojstvo određuje da je SVR manje osjetljiv na greške nego kvadratna funkcija gubitka. Kada bi direktno uspoređivali metodu kvadratne funkcije i metodu regresije potpornih vektora (SVR), možemo zaključiti da je SVR robusnija od metode kvadratne funkcije kada je riječ o rješavanju odstupanja.

### 5.3. Rangiranje

Metoda rangiranja-CNN sastoji se od dvije faze: pred-vježbe sa slikama lica i finog podešavanja s dobno označenim licima. Na početku imamo osnovnu mrežu koja je unaprijed uvježbana s neograničenim slikama lica kako bi se naučila nelinearna transformacija ulaznih uzoraka koja bilježi njihovu glavnu varijaciju. Iz osnovne mreže zatim treniramo skup osnovnih binarnih CNN-ova s rednim oznakama starosti. Konkretno, kategoriziramo uzorke u dvije skupine: s rednim oznakama više ili niže od određene dobi, a zatim ih koristimo za treniranje odgovarajućeg binarnog CNN-a. Potpuno povezani slojevi u binarnom CNN-u najprije izravnavaju značajke dobivene u prethodnim slojevima, a zatim ih povezuju s binarnim predviđanjem. Težine se ažuriraju kroz SGD usporedbom predviđanja s danom oznakom. Naposljetku, svi binarni izlazi se agregiraju kako bi se napravilo konačno predviđanje dobi. [11]



Slika 16: Arhitektura osnovnog binarnog CNN-a. (Izvor: Chen, Zhang i Dong 2018)

Na slici 16 je prikazana osnovna arhitektura binarnog CNN-a na primjeru biometrijske karakteristike lica. Klasična arhitektura sastoji se od tri konvoluciona sloja i sloja za poduzorkovanje, te tri potpuno povezana sloja. Kao što je vidljivo na slici, započinje se s C1 koji je prvi konvolucijski sloj s mapama značajki povezanih sa susjednim područjem  $5 \times 5$  u ulazu. Postoji 96 filtara primijenjenih na svaki od 3 kanala (RGB) ulaza, nakon čega slijedi ReLU (engl. *Rectified Linear Unit*). S2 je prvi sloj pod-uzorkovanja s mapama značajki povezanih s odgovarajućim mapama značajki u C1. Nakon S2 slijedi lokalna normalizacija odgovora (LRN) koja može pomoći u generalizaciji. C3 radi na sličan način kao C1 s 256 filtara u 96 kanala i veličinom filtera  $5 \times 5$ , nakon čega slijedi ReLU. Sloj S4 funkcionira slično kao S2, a slijedi ga LRN. C5 je treći konvolucijski sloj s 384 filtara u 256 kanala i manjom veličinom filtera  $3 \times 3$ , nakon čega slijedi treći maksimalni sloj udruživanja S6. [11] Sa F7 krećemo u prvi od tri potpuno povezana sloja u kojem su mape značajki spljoštene u vektor značajki. Postoji 512 neurona u F7, nakon čega slijedi ReLU i ispadajući sloj. F8 je drugi potpuno povezani sloj s 512 neurona koji prima izlaz od F7 nakon čega slijedi ReLU i još jedan ispadajući sloj. F9 je treći potpuno povezani sloj i izračunava vjerojatnost da ulaz  $x$  (tj. izlaz nakon F8) pripada klasi  $i$  pomoću logističke funkcije. Logistička funkcija se koristi budući da je izlaz osnovnog CNN-a binaran. [11] Optimalni parametri modela mreže obično se uče minimiziranjem funkcije gubitaka. Koristimo negativnu logaritamsku vjerojatnost kao funkciju gubitka i minimiziramo je pomoću SGD-a.

Svakim slojem kako prolazimo kroz binarni CNN generira se skup mapa značajki. Budući da model na slici ima 3 konvoluciona sloja i 3 sloja za povezivanje, može se generirati

ukupno 6 skupova mapa značajki (CONV1, POOL1, CONV2, POOL2, CONV3 i POOL3), gdje je CONV1 skup mapa značajki nakon prvoj sloja C1, POOL1 nakon sloja S2 i tako dalje. Svakim skupom mapa značajki koncentracija je na različita područja ulaznog lica od kojih neka ističu oči i usta, dok se druga fokusiraju na konture lica. Nakon konačnog sloja udruživanja S6, izlazni POOL3 s 384 mape značajki bio bi spljošten u F7 kao vektor koji predstavlja lice prije procjene dobi. [11]

Ako pretpostavimo da je  $x_i$  vektor značajki koji predstavlja  $i$ -ti uzorak i da je  $y_i \in 1 \dots K$  odgovarajuća ordinalna oznaka. Kako bismo mogli trenirati  $k$ -ti binarni CNN moramo podijeliti cijeli skup podataka  $D$  u dva podskupa s dobnim vrijednostima višim, nižim ili jednakim od  $k$ ,

$$D_k^+ = (x_i, +1) | y_i > k, \quad D_k^- = (x_i, -1) | y_i \leq k$$

Pogrešku binarnog rangiranja  $\varepsilon(x_i)$  definiramo  $\varepsilon(x_i) = [f_k(x_i) > 0][y_i \leq k] + [f_k(x_i) \leq 0][y_i > k]$ , gdje je  $f_k(x_i)$  izlaz osnovne mreže, a  $[\cdot]$  označava operator testa istinitosti koji je 1 ako je unutarnji uvjet istinit, a u suprotnom je 0.

S obzirom na nepoznati ulaz  $x_i$ , prvo koristimo osnovne mreže za donošenje skupa binarnih odluka, a zatim ih agregiramo kako bismo napravili konačno predviđanje dobi  $r(x_i)$ ,

$$r(x_i) = 1 + \sum_{k=1}^{K-1} [f_k(x_i) > 0]$$

Može se pokazati da je konačna pogreška rangiranja ograničena maksimumom binarnih pogrešaka rangiranja. To jest, CNN-ovi rezultati rangiranja mogu se poboljšati optimizacijom osnovnih mreža.

---

**Algorithm 1** Algorithm of Ranking-CNN

---

```

1: procedure TRAINING PROCEDURE
2:   pretrain Base CNN
3:   top:
4:   for  $k = 1$  to  $K-1$  do
5:      $e_k \leftarrow k_{th}$  Basic CNN
6:   end for
7:    $k' \leftarrow \text{sort } e_k$ 
8:   for  $k' = 1$  to  $K-1$  do
9:      $D_k^+ = \{(x_i, +1) | y_i > k'\}$ 
10:     $D_k^- = \{(x_i, -1) | y_i \leq k'\}$ 
11:    fine-tune  $k'_{th}$  Basic CNN  $\leftarrow e_{k'}$ 
12:   end for
13:   if not converged
14:     goto top
15:   end if
16: procedure TESTING PROCEDURE
17:   for  $k = 1$  to  $K-1$  do
18:      $f_k(x_i) \leftarrow k_{th}$  Basic CNN
19:   end for
20:   final prediction  $r(x_i) \leftarrow 1 + \sum_{k=1}^{K-1} [f_k(x_i) > 0]$ 

```

---

Slika 17: Algoritam rangiranja-CNN. (Izvor: Chen, Zhang i Dong 2018)

Općenito, korištenje CNN metode za procjenu dobi pruža okvir za istovremeno učenje značajki i dobno rangiranje na temelju slika. Također, oznake dobi prirodno su ordinalne, a rangiranje-CNN može zadržati relativni redni odnos među različitim dobnim skupinama.

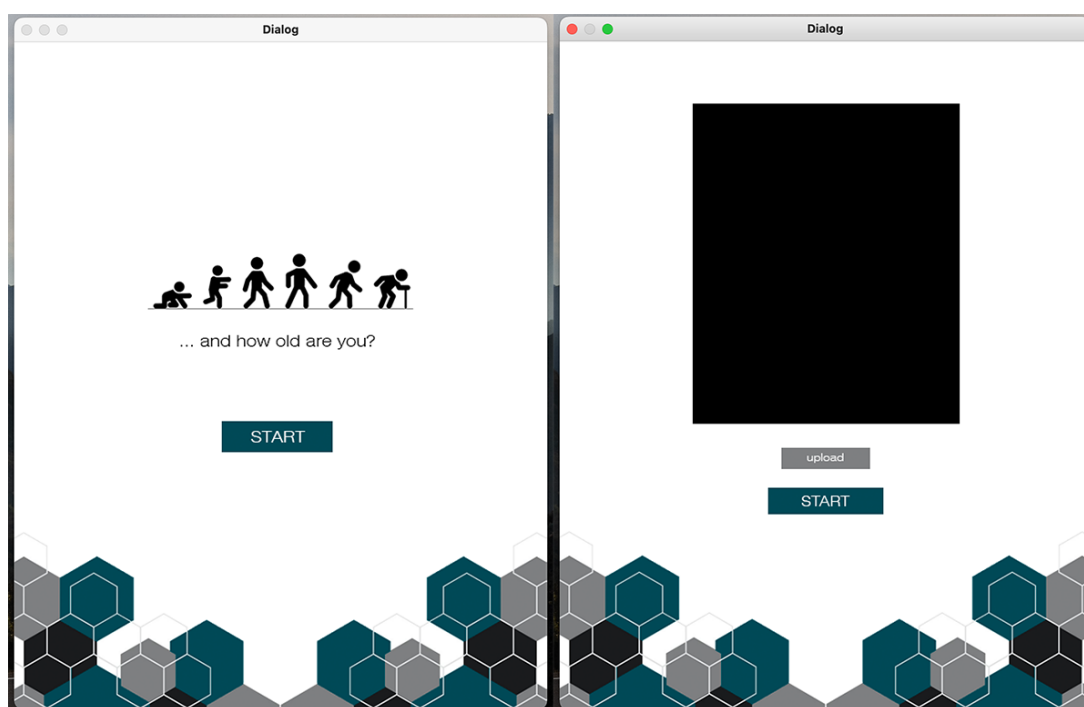
## 6. Procjena dobi na temelju hoda

Kao što smo već spomenuli, praktični dio ovoga rada sastoji od izrade aplikacije za procjenu dobne skupine osobe na temelju hoda. U nastavku ovog rada opisan je postupak izrade, navode se metode koje su se koristile te je na kraju prikazan finalni izgled i rješenje ove aplikacije.

### 6.1. Izrada korisničkog sučelja

Prva faza u izradi naše aplikacije je izrada grafičkog korisničkog sučelja. Za izradu je korišten PyQt5, točnije QtDesigner. QtDesigner je pristupačan jer nakon što se ručno nacrtaju sučelje koje se želi, jednostavnim naredbom kroz terminal se napravljeni dizajn pretvara u Python kod. Također, kako bi sučelje izgledalo atraktivnije, korišten je Adobe Photoshop za izradu pozadine.

Cilj izrade grafičkog korisničkog sučelja za ovu aplikaciju bio je da s jednostavnim i pristupačnim dizajnom izdvojimo najbitniju stvar, a to je njena funkcionalnost. Na slici 18 može se vidjeti kako aplikacija izgleda. Slika lijevo prikazuje početni prozor koji korisnik prvo vidi kada otvori aplikaciju, a zatim pritiskom na gumb "Start" otvara se prozor desno na slici.



Slika 18: Grafičko korisničko sučelje.

Kako bi povezali ova dva prozora, stvorili smo klasu *MainCover* u kojoj smo definirali funkciju *move* u kojoj smo importirali klasu *MainPrva* koja inicijalizira otvaranje drugog prozora.

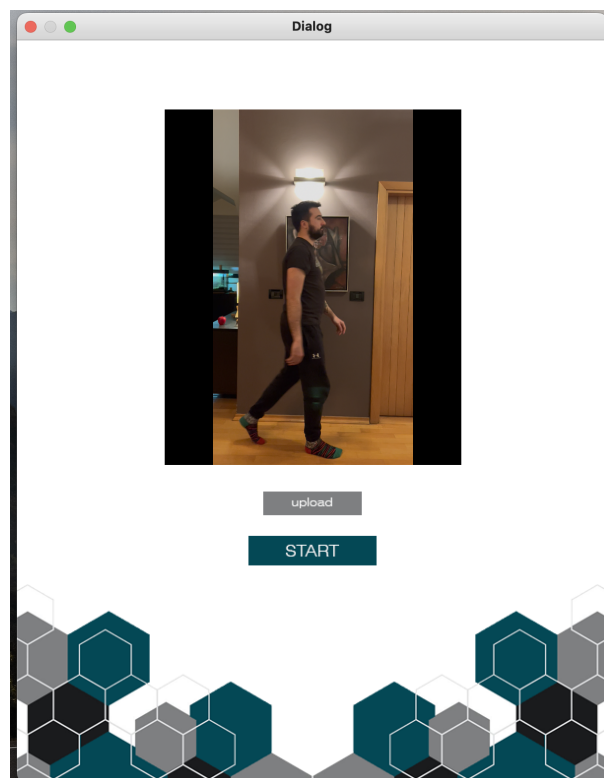
Prije nego što smo mogli pred-procesirati video, bilo je potrebno taj video učitati kako bi ga mogli kroz cijeli kod dalje koristiti. U QtDesigneru smo nacrtali pravokutni *Widget* kako bi u njega mogli spremiti učitani video. Budući da je video drugačiji od slike, nismo ga mogli samo direktno učitati već smo morali koristiti *QMediaPlayer*. Jednom kada smo postavili video zapis u predefinirani *mediaPlayer* objekt, jednostavno smo ga samo proslijedili u naš *self.widget*.

## Programski kod

```
self.mediaPlayer = QMediaPlayer(None, QMediaPlayer.VideoSurface)
self.mediaPlayer.setVideoOutput(self.widget)

def openFile(self):
    self.filename, _ = QtWidgets.QFileDialog.getOpenFileName(self, "Open_Video",
        os.getcwd(), ("Video_Files_(*.avi*.mp4)"))
    if self.filename != '':
        self.mediaPlayer.setMedia(QMediaContent(QUrl.fromLocalFile(self.filename)))
        self.mediaPlayer.play()
```

Kako izgleda kada je video uspješno učitano može se vidjeti na slici 19.



Slika 19: Učitani video.

## 6.2. Pred-procesiranje videa

Druga faza bila je priprema videa kako bi se podaci dobiveni u videu mogli koristiti za daljnju obradu podataka. Pristup koji je bio korišten je da se video podijelio na "frame-ove" kako bi se iz odabranoga broja tih frame-ova mogao dobiti dio ciklusa hoda.

### Programski kod

```
if (video.isOpened() == False):
    print("Error_opening_video_file!")

while (video.isOpened()):
    ret, frame = video.read()
    if ret == True:
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

Budući da smo odabrali pristup prepoznavanja hoda bez modela, idući korak bio je izvući siluete iz videa. Kako ne bi imali ponovljenog i nepotrebnog koda, svaki puta kada se video podijelio na novi frame također je i iz tog frame-a izvukao i siluetu.

### Programski kod

```
results = selfie_segmentation.process(image)

mask = results.segmentation_mask

masked = (mask * 255).astype('uint8')

framed = imutils.resize(masked, width=180)
```

Sam postupak prepoznavanja siluete je ubrzalo korištenje Google-ovog MediaPipe-a. MediaPipe je okvir rješenja za strojno učenje koji je razvio Google i ima ugrađene razne unaprijed obučene modele pa tako i model uklanjanja pozadine. Uklanjanje pozadine u osnovi vraća segmentiranu masku prednjeg plana sa slike odnosno subjekta, a ostale stvari su pozadina. Na slici 20 može se vidjeti silueta izvučena iz jednog frame-a našeg videa. Također, kako bi se kasnije mogli koristiti frame-ovi odnosno slike koje smo zapisali, morali smo pretvoriti dobivenu masku u nama odgovarajuće dimenzije zato ćemo dalje prosljeđivati *framed*, a ne *mask*, odnosno *masked*.



Slika 20: Silueta

Zadnji korak u pred-procesiranju videa bio je odabrati koje frame-ove da spremamo za daljnju obradu podataka.

### Programski kod

```
video = cv2.VideoCapture(self.filename)
num_frames = video.get(cv2.CAP_PROP_FRAME_COUNT)
f = num_frames - (num_frames - 1)
middle = num_frames // 2
```

Budući da je nama najbitniji srednji dio videa, gdje se točno vidi cijeli odabrani ciklus hoda, te nam potpuno crni frame-ovi samo narušavaju kasnije rezultat, bilo je potrebno odrediti interval unutar kojeg ćemo spremati frame-ove. U varijablu smo spremili koliko frame-ova učitani video ima kako bi mogli najlakše odrediti sredinu.

### Programski kod

```
if (i >= (f + 22) and i <= (middle)):
    cv2.imwrite('/Users/nikakale/Desktop/Frames/Frame/Frame' + str(i) + '.jpg', framed)
```

Jednom kada smo odredili gdje je sredina, i metodom pokušaja i pogrešaka ustanovili da je najbolje zanemariti prvih 20-tak frame-ova, dobili smo niz frame-ova s kojim ćemo kasnije dalje raditi (slika 21).



Slika 21: Spremljeni frame-ovi.

## 6.2.1. Gait Energy Image (GEI)

Već smo ranije u radu spominjali GEI, odnosno Gait Energy Image. GEI definiramo kao prosjek svih silueta u jednom ciklusu hoda i upravo zato smo morali spremati više frame-ova iz našeg videa. Također, GEI predstavljaju većinu mogućih informacija o sekvencama slika, a definirana je na sljedeći način:

$$G(x, y) = \frac{1}{N} \sum_{t=1}^N B_t(x, y)$$

gdje je N broj okvira u jednom kompletnom ciklusu hoda osobe koja hoda, t je broj okvira u slijedu (trenutak vremena), a x i y su vrijednosti u koordinati 2D slike [15].

### Programski kod

```
def mass_center(img, is_round=True):
    Y = img.mean(axis=1)
    X = img.mean(axis=0)

    Y_ = np.sum(np.matmul(np.arange(Y.shape[0]), Y)) / np.sum(Y)
    X_ = np.sum(np.matmul(np.arange(X.shape[0]), X)) / np.sum(X)

    if is_round:
        return int(round(X_)), int(round(Y_))
    return X_, Y_

def image_extract(img, newsize):
    x_s = np.where(img.mean(axis=0) != 0)[0].min()
    x_e = np.where(img.mean(axis=0) != 0)[0].max()

    y_s = np.where(img.mean(axis=1) != 0)[0].min()
    y_e = np.where(img.mean(axis=1) != 0)[0].max()

    x_c, _ = mass_center(img)
    x_s = x_c - newsize[1] // 2
    x_e = x_c + newsize[1] // 2
    img = img[y_s:y_e, x_s if x_s > 0 else 0:x_e if x_e < img.shape[1] else
            img.shape[1]]
    return resize(img, newsize)

images = [image_extract(i, (128, 88)) for i in images]

gei = np.mean(images, axis=0)
gei_model = np.expand_dims(gei, axis=0)

gei_png = (gei * 255).astype('uint8')
cv2.imwrite('/Users/nikakale/Desktop/Frames/GEI.png', gei_png)
```



Slika 22: Dobiveni GEI.

Na slici 22 možemo vidjeti konačni rezultat koji smo pokušavali dobiti nakon cijelog predprocesiranja videa, a to je GEI. Idući korak nam je proslijediti GEI sliku u istrenirani model kako bi dobili konačno rješenje, odnosno dobni rang kojoj osoba pripada.

### 6.3. Metoda rangiranja-CNN

Metoda rangiranja-CNN je metoda koju smo koristili za predviđanje dobi, odnosno dobno-g rang-a osobe. Već smo ranije u ovome radu objasnili što je to točno metoda rangiranja-CNN kao i što je duboko učenje (*engl. Deep Learning*). Budući da treniranje neuronskih mreža zahtjeva veću količinu podataka prvo nam je bila potrebna baza od nekoliko tisuća podataka. U ovom radu koristili smo *OULP-Age* bazu, odnosno "*The OU-ISIR Gait Database, Large Population Dataset with Age*" [16] koja u sebi sadrži 63 846 podataka. Podaci se sastoje od GEI slika svih subjekata, te tablice u kojoj su nazivu slike dodijeljene godine u kojima je subjekt na slici, a rang godina je od 2 do 90 godina.

Prvi korak kako bi mogli koristiti navedene podatke bio je kreirati listu sa svim slikama, te drugu listu sa dodijeljenim godinama i spremiti u npz datoteku. Odredili smo 7 dobnih skupina: '0–2', '3–7', '8–12', '13–19', '20–36', '37–65', '65+' te godine spremili u listu pod index-om kojoj dobnjoj skupini pripada.

#### Programski kod

```
for a in all_images:
    split_a = a.split(".")
    num_img = split_a[0]

    for x, y, z in data:
        if(x.__contains__(num_img)):
            images.append(cv2.imread(path + 'GEI/' + a))
            age = int(y)

            if age>=0 and age<=2:
                ages.append(0)
            elif age>=3 and age<=7:
                ages.append(1)
```

```

elif age>=8 and age<=12:
    ages.append(2)
elif age>=13 and age<=19:
    ages.append(3)
elif age>=20 and age<=36:
    ages.append(4)
elif age>=37 and age<=65:
    ages.append(5)
elif age>=66:
    ages.append(6)

```

```
np.savez_compressed('compressed.npz', images=images, ages=ages)
```

Jednom kada smo kreirali npz datoteku pripremili smo sve potrebne podatke koje ćemo trenirati u našem modelu. Idući korak je učitati datoteku u naš model i dodijeliti listi X vrijednosti slika, a listi y vrijednosti godina. Također, kako bi mogli raditi sve s matricama, bilo je potrebno vrijednosti godina, odnosno listu y, pretvoriti u matrice. Nadalje, provodimo test razdvajanja podataka i preoblikovanja X u ispravan format kako bismo ga unijeli u model u daljnjim koracima i stvaramo objekt za povećanje podataka pomoću ImageDataGeneratora.

## Programski kod

```

loaded = np.load('compressed.npz')

X = loaded['images']
y = loaded['ages']

y = np_utils.to_categorical(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,
                                                    random_state=42)

X_train = np.array(X_train).reshape(44692, 128, 88, 3)
X_test = np.array(X_test).reshape(19154, 128, 88, 3)

train_datagen = ImageDataGenerator(rescale=1./255,
                                    horizontal_flip=True,
                                    shear_range=0.2,
                                    zoom_range=0.2,
                                    height_shift_range=0.1,
                                    width_shift_range=0.1,
                                    rotation_range=15)

test_datagen = ImageDataGenerator(rescale=1./255)

```

Idući korak bio je kreiranje i kompiliranje našeg modela. Na slici 23 možemo vidjeti rezime našeg modela po kojem ćemo dalje raditi.

## Programski kod

```
final_cnn = Sequential()

final_cnn.add(Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=(128,
    88, 3)))    # 3rd dim = 1 for grayscale images.
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=64, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=128, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=256, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(GlobalAveragePooling2D())

final_cnn.add(Dense(132, activation='relu'))

final_cnn.add(Dense(7, activation='softmax'))

final_cnn.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['
    accuracy'])

final_cnn.summary()
```

| Layer (type)                                      | Output Shape        | Param # |
|---------------------------------------------------|---------------------|---------|
| conv2d (Conv2D)                                   | (None, 126, 86, 32) | 896     |
| average_pooling2d (AveragePooling2D)              | (None, 63, 43, 32)  | 0       |
| conv2d_1 (Conv2D)                                 | (None, 61, 41, 64)  | 18496   |
| average_pooling2d_1 (AveragePooling2D)            | (None, 30, 20, 64)  | 0       |
| conv2d_2 (Conv2D)                                 | (None, 28, 18, 128) | 73856   |
| average_pooling2d_2 (AveragePooling2D)            | (None, 14, 9, 128)  | 0       |
| conv2d_3 (Conv2D)                                 | (None, 12, 7, 256)  | 295168  |
| average_pooling2d_3 (AveragePooling2D)            | (None, 6, 3, 256)   | 0       |
| global_average_pooling2d (GlobalAveragePooling2D) | (None, 256)         | 0       |
| dense (Dense)                                     | (None, 132)         | 33924   |
| dense_1 (Dense)                                   | (None, 7)           | 931     |
| Total params: 423,271                             |                     |         |
| Trainable params: 423,271                         |                     |         |
| Non-trainable params: 0                           |                     |         |

Slika 23: Model

Zadnji korak prije pokretanja modela bio je da se pohranjuje samo najbolji model, točnije model koji ima najbolju točnost odnosno u našem slučaju najbolji *val-acc*.

## Programski kod

```
metric = 'val_acc'
checkpoint = ModelCheckpoint(filepath="final_cnn_model_checkpoint.h5",
                             monitor=metric,
                             save_best_only=True,
                             save_weights_only=False,
                             verbose=2
                             )
```

Treniramo model koristeći CNN model tako da nam je definirani *batch size* = 512, a *epoch* = 70.

## Programski kod

```
history = final_cnn.fit(train_data,
                        validation_data=test_data,
                        epochs=epochs,
                        callbacks=[checkpoint],
                        shuffle=False
                        )
```

Na slici 24 možemo vidjeti početak treniranja modela koji započinje s epochom 1 od 70 i vidimo da je točnost modela na samom početku tek 0.0137.

```
Epoch 1/70
1/88 [.....] - ETA: 8:15 - batch: 0.0000e+00 - size: 512.0000 - loss: 1.9454 - acc: 0.0137
2/88 [.....] - ETA: 3:55 - batch: 0.5000 - size: 512.0000 - loss: 1.8992 - acc: 0.1895
3/88 [>.....] - ETA: 3:55 - batch: 1.0000 - size: 512.0000 - loss: 1.8387 - acc: 0.2487
4/88 [>.....] - ETA: 3:53 - batch: 1.5000 - size: 512.0000 - loss: 1.8102 - acc: 0.2686
5/88 [>.....] - ETA: 3:54 - batch: 2.0000 - size: 512.0000 - loss: 1.7830 - acc: 0.2887
6/88 [=>.....] - ETA: 3:49 - batch: 2.5000 - size: 512.0000 - loss: 1.7615 - acc: 0.2956
7/88 [=>.....] - ETA: 3:46 - batch: 3.0000 - size: 512.0000 - loss: 1.7434 - acc: 0.3039
8/88 [=>.....] - ETA: 3:43 - batch: 3.5000 - size: 512.0000 - loss: 1.7264 - acc: 0.3145
9/88 [==>.....] - ETA: 3:40 - batch: 4.0000 - size: 512.0000 - loss: 1.7120 - acc: 0.3151
10/88 [==>.....] - ETA: 3:38 - batch: 4.5000 - size: 512.0000 - loss: 1.6978 - acc: 0.3195
```

Slika 24: Početak treniranja modela

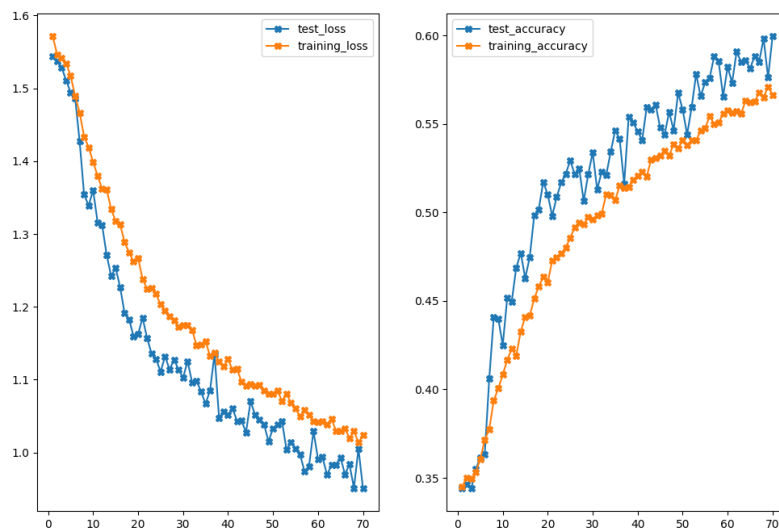
Ukupno trajanje treniranja modela bilo je 10 sati, te je najbolji model spremljen sa točnošću od 0.59961 što je u našem slučaju zadovoljavajuće budući da je to najbolja točnost koju smo uspjeli postići kroz treniranje nekoliko modela. Na slici 25 može se vidjeti završetak treniranja našeg modela.

```
70/88 [====>.....] - ETA: 2:11 - batch: 34.5000 - size: 512.0000 - loss: 1.0245 - acc: 0.5661
71/88 [====>.....] - ETA: 2:07 - batch: 35.0000 - size: 512.0000 - loss: 1.0250 - acc: 0.5664
72/88 [====>.....] - ETA: 2:02 - batch: 35.5000 - size: 512.0000 - loss: 1.0242 - acc: 0.5669
73/88 [====>.....] - ETA: 1:58 - batch: 36.0000 - size: 512.0000 - loss: 1.0248 - acc: 0.5665
74/88 [====>.....] - ETA: 1:53 - batch: 36.5000 - size: 512.0000 - loss: 1.0245 - acc: 0.5669
75/88 [====>.....] - ETA: 1:48 - batch: 37.0000 - size: 512.0000 - loss: 1.0243 - acc: 0.5667
76/88 [====>.....] - ETA: 1:42 - batch: 37.5000 - size: 512.0000 - loss: 1.0242 - acc: 0.5666
77/88 [====>.....] - ETA: 1:35 - batch: 38.0000 - size: 512.0000 - loss: 1.0253 - acc: 0.5663
78/88 [====>.....] - ETA: 1:28 - batch: 38.5000 - size: 512.0000 - loss: 1.0251 - acc: 0.5663
79/88 [====>.....] - ETA: 1:21 - batch: 39.0000 - size: 512.0000 - loss: 1.0258 - acc: 0.5657
80/88 [====>.....] - ETA: 1:13 - batch: 39.5000 - size: 512.0000 - loss: 1.0255 - acc: 0.5656
81/88 [====>.....] - ETA: 1:05 - batch: 40.0000 - size: 512.0000 - loss: 1.0253 - acc: 0.5654
82/88 [====>.....] - ETA: 57s - batch: 40.5000 - size: 512.0000 - loss: 1.0254 - acc: 0.5654
83/88 [====>.....] - ETA: 48s - batch: 41.0000 - size: 512.0000 - loss: 1.0250 - acc: 0.5657
84/88 [====>.....] - ETA: 39s - batch: 41.5000 - size: 512.0000 - loss: 1.0246 - acc: 0.5657
85/88 [====>.....] - ETA: 30s - batch: 42.0000 - size: 512.0000 - loss: 1.0239 - acc: 0.5661
86/88 [====>.....] - ETA: 20s - batch: 42.5000 - size: 512.0000 - loss: 1.0241 - acc: 0.5663
87/88 [====>.....] - ETA: 10s - batch: 43.0000 - size: 512.0000 - loss: 1.0238 - acc: 0.5664
88/88 [====>.....] - ETA: 0s - batch: 43.5000 - size: 507.8636 - loss: 1.0238 - acc: 0.5665
Epoch 70: val_acc improved from 0.59826 to 0.59961, saving model to final_cnn_model_checkpoint.h5

88/88 [====>.....] - 1019s 12s/step - batch: 43.5000 - size: 507.8636 - loss: 1.0238 - acc: 0.5665 - val_loss: 0.9508 - val_acc: 0.5996
```

Slika 25: Kraj treniranja modela

Na kraju smo kreirali graf radi lakše vizualizacije samog procesa.



Slika 26: Vizualizacija procesa treniranja.

## 6.4. Prikaz finalnog rješenja

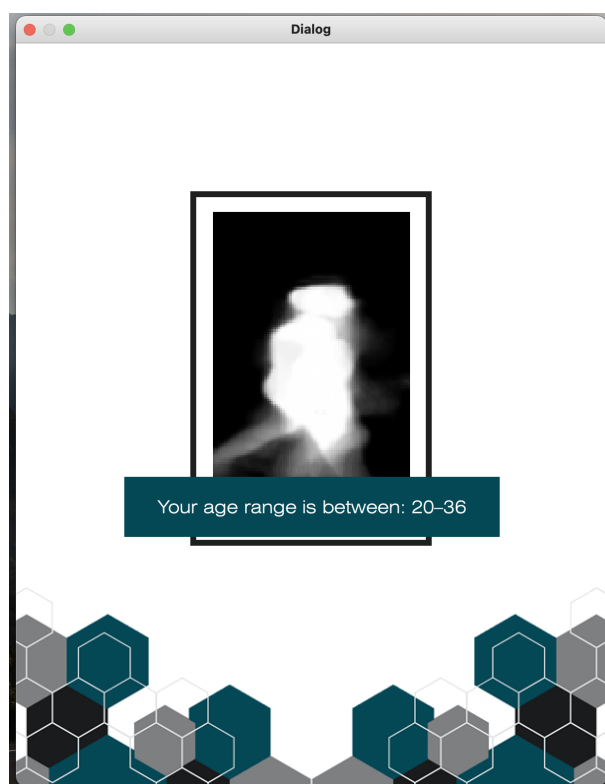
Završna faza naše aplikacije je predviđanje i ispis dobne skupine kojoj osoba pripada. Kako bi mogli koristiti naš model bilo ga je potrebno uvesti u kod, a kako smo si već unaprijed priredili dobar format podataka, nije ostalo ništa drugo osim iskoristiti funkciju *predict* na našem dobivenom GEI-u.

### Programski kod

```
model = load_model('/Users/nikakale/Desktop/final_cnn_model_checkpoint.h5')

age = model.predict(gei_model)
result = ranges[np.argmax(age)]
self.label_2.setText("Your_age_range_is_between:_ " + result)
```

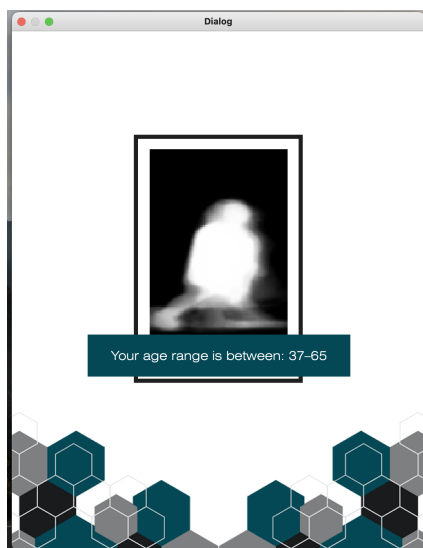
Rezultat se zapisuje u *label* te nam dobiveno rješenje prikazuje kao na slici 27. Možemo zaključiti da je dobna skupina dobro procijenjena budući da osoba na videu ima 24 godine.



Slika 27: Krajnja procijenjena dobna skupina osobe.

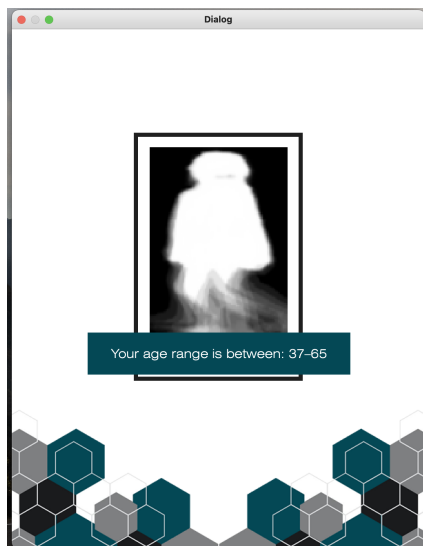
U nastavku su prikazani još neki primjeri kako naša aplikacija radi:

- Dobro procijenjena dob - osoba ima 54 godine.



Slika 28: Krajnja procijenjena dobna skupina osobe.

- Loše procijenjena dob - osoba ima 70 godina.



Slika 29: Krajnja procijenjena dobna skupina osobe.

## 7. Zaključak

U prvom dijelu ovoga rada govorilo se o hodu, kako ga definiramo, različitim fazama hoda i bitnim značajkama hoda. Opširno smo opisali prepoznavanje hoda zajedno s detaljnim opisom i prikazom različitih načina i metoda kako se može primjenjivati. Još jedna od bitnih tema koje smo se dotaknuli bila je procjena dobi općenito na temelju biometrijskih karakteristika kako bi dobili dobar uvod u praktični dio ovoga rada. U glavnom fokusu ovoga rada bila je izrada same aplikacije koja procjenjuje dob osobe na temelju hoda te se nastojalo čitatelju predložiti zanimljivost nastanka ove aplikacije. Ova aplikacija, kao i svaka druga, ima prostora za unaprjeđenje i poboljšanje te se može još dodatno razvijati. Na primjer, može se postići bolja točnost na svim primjerima tako da se pomoću neuralne mreže određuje najbolji ciklus hoda ili da se omogući bolja detekcija iz raznih kutova snimanja i slično. Osobno smatram da je biometrija jedna od zanimljivih grana u znanosti i nadam se da sam svojim radom uspjela približiti i zainteresirati one koji se još nisu susreli s njom.

# Popis literature

- [1] M. S. Nixon, J. N. Carter, D. Cunado, P. S. Huang i S. V. Stevenage, „Automatic gait recognition,” *In: Jain, A.K., Bolle, R., Pankanti, S. (eds) Biometrics. Springer, Boston, MA.*, str. 232–249, 1996. DOI: 10.1007/0-306-47044-6\_1.
- [2] N. Mansouri, M. A. Issa i Y. B. Jemaa, „Gait features fusion for efficient automatic age classification,” *IET Comput. Vis.*, 2018, Vol. 12 Iss. 1, str. 69–75, 2017. DOI: 10.1049/iet-cvi.2017.0055.
- [3] „hod,” Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža. (2021.), adresa: <https://www.enciklopedija.hr/natuknica.aspx?id=25844> (pogledano 27. 12. 2022.).
- [4] J. E. Mason, I. Traoré i I. Woungang, *Machine Learning Techniques for Gait Biometric Recognition*. Springer, 2016.
- [5] H. G. Chambers i D. H. Sutherland, „A Practical Guide to Gait Analysis,” *The Journal of the American Academy of Orthopaedic Surgeons*, str. 222–231, 2002.
- [6] D. Cunado, M. S. Nixon i J. N. Carter, „Automatic extraction and description of human gait models for recognition purposes,” *Computer Vision and Image Understanding*, str. 1–41, 2003.
- [7] S. Z. Li i A. Jain, *Encyclopedia of Biometrics*. Springer, 2009.
- [8] D. S. Matovski, M. S. Nixon i J. N. Carter, „Gait Recognition,” *In: Ikeuchi, K. (eds) Computer Vision.*, str. 309–318, 2014. DOI: 10.1007/978-0-387-31439-6\_375.
- [9] N. Mansouri, „Automatic Age Estimation: A Survey,” *Computación y Sistemas*, Vol. 24, No. 2, str. 883–895, 2020. DOI: 10.13053/CyS-24-2-3317.
- [10] T. B. Aderinola, T. Connie, T. S. Ong, W.-C. Yau i A. B. J. Teoh, „Learning Age From Gait: A Survey,” *IEEE Access*, sv. 9, str. 100352–100368, 2021. DOI: 10.1109/ACCESS.2021.3095477.
- [11] S. Chen, C. Zhang i M. Dong, „Deep Age Estimation: From Classification to Ranking,” *IEEE Transactions on Multimedia*, sv. 20, br. 8, str. 2209–2222, 2018. DOI: 10.1109/TMM.2017.2786869.
- [12] A. Madhavi, G. Bhuvana Sree, V. Shriya, B. Shanmukh i T. Harshitha, „Human Age Estimation Using Support Vector Machine,” *Machine Learning Technologies and Applications*, C. K. Mai, A. B. Reddy i K. S. Raju, ur., Singapore: Springer Singapore, 2021., str. 273–286, ISBN: 978-981-33-4046-6.

- [13] „SUPPORT VECTOR MACHINES(SVM).” (2018.), adresa: <https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589> (pogledano 21. 2. 2023.).
- [14] G. Guo, Y. Fu, C. R. Dyer i T. S. Huang, „Image-Based Human Age Estimation by Manifold Learning and Locally Adjusted Robust Regression,” *IEEE Transactions on Image Processing*, sv. 17, br. 7, str. 1178–1188, 2008. DOI: 10.1109/TIP.2008.924280.
- [15] F. Imran, S. Bakchy, A. Gorski i M. Islam, „Human Gait Analysis using Gait Energy Image,” siječanj 2017.
- [16] C. Xu, Y. Makihara, G. Ogi, X. Li, Y. Yagi i J. Lu, „The OU-ISIR Gait Database Comprising the Large Population Dataset with Age and Performance Evaluation of Age Estimation,” *IPSJ Trans. on Computer Vision and Applications*, sv. 9, br. 24, str. 1–14, 2017.
- [17] T. U. Islam, L. K. Awasthi i U. Garg, „Gender and Age Estimation from Gait: A Review,” *D. Gupta et al. (eds.), International Conference on Innovative Computing and Communications, Advances in Intelligent Systems and Computing 1166*, str. 947–962, 2021. DOI: 10.1007/978-981-15-5148-2\_82.
- [18] B. Abirami, T. S. Subashini i V. Mahavaishnavi, „Automatic age-group estimation from gait energy images,” *Materials Today: Proceedings*, 2020. DOI: 10.1016/j.matpr.2020.08.298.
- [19] J. E. Boyd i J. J. Little, „Biometric Gait Recognition,” *M. Tistarelli, J. Bigun, and E. Grosso (Eds.): Biometrics School 2003, LNCS 3161*, str. 19–42, 2005.
- [20] Y. Makihara, M. Okumura, H. Iwama i Y. Yagi, „Gait-based Age Estimation using a Whole-generation Gait Database,” 2011. DOI: 10.1109/IJCB.2011.6117531.
- [21] X. Li, Y. Makihara, C. Xu, Y. Yagi i M. Ren, „Gait-based human age estimation using age group-dependent manifold learning and regression,” *Multimed Tools Appl (2018)*, str. 28 333–28 354, 2018. DOI: 10.1007/s11042-018-6049-7.
- [22] M. Nabila, A. I. Mohammed i B. J. Yousra, „Gait-based human age classification using a silhouette model,” *IET Biom., 2018, Vol. 7 Iss. 2*, str. 116–124, 2017. DOI: 10.1049/iet-bmt.2016.0176.
- [23] S. Zaghebani, N. Boujneh i M. S. Bouhlef, „Age estimation using deep learning,” *Computers & Electrical Engineering*, sv. 68, str. 337–347, 2018., ISSN: 0045-7906. DOI: <https://doi.org/10.1016/j.compeleceng.2018.04.012>. adresa: <https://www.sciencedirect.com/science/article/pii/S0045790617334298>.
- [24] C. Xu, Y. Makihara, R. Liao i dr., „Real-Time Gait-Based Age Estimation and Gender Classification from a Single Image,” *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021., str. 3459–3469. DOI: 10.1109/WACV48630.2021.00350.
- [25] „Gait Recognition System: Deep Dive into This Future Tech.” (2021.), adresa: <https://recfaces.com/articles/what-is-gait-recognition> (pogledano 27. 12. 2022.).
- [26] „Gait Recognition Systems Can Identify You with Your Manner of Walking.” (2022.), adresa: <https://www.bayometric.com/gait-recognition-identify-with-manner/> (pogledano 27. 12. 2022.).

- [27] „Age Detection using CNN with Keras.” (2022.), **adresa:** <https://medium.com/mlearning-ai/age-detection-using-cnn-with-keras-with-source-code-easiest-way-easy-implementation-57c107b23bc4> (**pogledano 10.2.2022.**).

# Popis slika

|     |                                                                                                  |    |
|-----|--------------------------------------------------------------------------------------------------|----|
| 1.  | Potpuni ciklus hodanja (Izvor: Chambers i Sutherland 2002) . . . . .                             | 4  |
| 2.  | Kutovi rotacije kukova (a) i koljena (b). (Izvor: Cunado, Nixon i N.Carter 2003) . .             | 5  |
| 3.  | Srednji uzorak rotacije kukova. (Izvor: Cunado, Nixon i N.Carter 2003) . . . . .                 | 5  |
| 4.  | Značajke silueta i oblik. (Izvor: Li i Jain 2009) . . . . .                                      | 6  |
| 5.  | Opći koraci sustava za prepoznavanje hoda. (Izvor: Matovski, Nixon i Carter 2014)                | 9  |
| 6.  | Komponente sustava za prepoznavanje hoda temeljenog na modelu. (Izvor: Li i Jain 2009) . . . . . | 10 |
| 7.  | Primjer parametara tijela za strukturni model. (Izvor: Li i Jain 2009) . . . . .                 | 11 |
| 8.  | Strukturni model noge. (Izvor: Li i Jain 2009) . . . . .                                         | 12 |
| 9.  | Binarna maska ciklusa hoda. (Izvor: Li i Jain 2009) . . . . .                                    | 12 |
| 10. | Primjeri potpisa hoda bez modela. (Izvor: Matovski, Nixon i Carter 2014) . . . . .               | 13 |
| 11. | Primjer "chroma-keyinga" u analizi hoda. (Izvor: Li i Jain 2009) . . . . .                       | 13 |
| 12. | Primjer optičkog protoka u analizi hoda. (Izvor: Li i Jain 2009) . . . . .                       | 14 |
| 13. | Primjeri procesa dobivanja potpisa hoda. (Izvor: Li i Jain 2009) . . . . .                       | 15 |
| 14. | Proces procjene dobi. (Izvor: Aderinola, Connie, Ong i dr. 2021) . . . . .                       | 16 |
| 15. | Primjer SVM korištenja (Izvor: Madhavi, Bhuvana Sree, Shriya i dr. 2021) . . . . .               | 17 |
| 16. | Arhitektura osnovnog binarnog CNN-a. (Izvor: Chen, Zhang i Dong 2018) . . . . .                  | 19 |
| 17. | Algoritam rangiranja-CNN. (Izvor: Chen, Zhang i Dong 2018) . . . . .                             | 20 |
| 18. | Grafičko korisničko sučelje. . . . .                                                             | 21 |
| 19. | Učitani video. . . . .                                                                           | 22 |
| 20. | Silueta . . . . .                                                                                | 23 |
| 21. | Spremljeni frame-ovi. . . . .                                                                    | 24 |
| 22. | Dobiveni GEI. . . . .                                                                            | 26 |

|     |                                                   |    |
|-----|---------------------------------------------------|----|
| 23. | Model . . . . .                                   | 28 |
| 24. | Početak treniranja modela . . . . .               | 29 |
| 25. | Kraj treniranja modela . . . . .                  | 30 |
| 26. | Vizualizacija procesa treniranja. . . . .         | 30 |
| 27. | Krajnja procijenjena dobna skupina osobe. . . . . | 31 |
| 28. | Krajnja procijenjena dobna skupina osobe. . . . . | 32 |
| 29. | Krajnja procijenjena dobna skupina osobe. . . . . | 32 |

# Popis tablica

|                                          |    |
|------------------------------------------|----|
| 1. Pristupi prepoznavanju hoda . . . . . | 10 |
|------------------------------------------|----|