

**Projekt iz predmeta Raspoznavanje uzoraka**

**Ak. god. 2015/16**

# **Detekcija i lokalizacija lica na temelju generalizirane Houghove transformacije**

Autori:

Mislav Larva

Tomislav Marinković

Josip Milić

Petar Pavlović

Domagoj Peregrin

Domagoj Vukadin

**siječanj, 2016.**

# Sadržaj:

<b>1. Projektni zadatak .....</b>	<b>3</b>
1.1 Opis projektnog zadatka .....	3
1.2 Pregled i opis srodnih rješenja .....	3
a. Konceptualno rješenje zadatka .....	5
<b>2. Postupak rješavanja zadatka .....</b>	<b>6</b>
2.1 Houghova transformacija .....	6
2.2 Postupak Generalizirane Houghove transformacije .....	7
2.2.1 Poseban slučaj GHT-a .....	8
2.2.2 Opći slučaj GHT-a .....	9
2.2.3 Prednosti, nedostaci i moguća poboljšanja GHT-a.....	11
<b>3. Ispitivanje rješenja .....</b>	<b>12</b>
3.1 Ispitna baza .....	12
3.1.1 Referentne slike .....	12
3.1.2 Ispitne slike .....	14
3.2 Rezultati učenja i ispitivanja .....	15
3.3 Analiza rezultata .....	21
<b>4. Opis programske implementacije rješenja .....</b>	<b>26</b>
4.1 Opis programske implementacije .....	26
4.2 Korištenje programske implementacije .....	28
<b>5. Zaključak .....</b>	<b>29</b>
<b>6. Literatura.....</b>	<b>30</b>

# 1. Projektni zadatak

## 1.1 Opis projektnog zadatka

U ovom radu razmatra se detekcija i lokalizacija lica na slici temeljem generalizirane Houghove transformacije. Za sami rad važno je razlikovati pojmove “detekcija” i “lokalizacija”. Detekcija lica je određivanje postoji li lice na nekoj slici ili ne postoji. Lokalizacija je pronalazak lica ako smo ustanovili da lice postoji u slici, odnosno definiranje u kojem dijelu slike se ono nalazi. Detekcija je vrlo važna u različitim primjenama gdje se treba identificirati ili verificirati ljudsko lice.

Dok je ljudima jako jednostavno detektirati postoji li lice na slici i gdje se točno nalazi, za računala to predstavlja velik problem. Stoga se još od prošlog stoljeća razvijaju različiti algoritmi koji omogućavaju da računalo također može pod nekim određenim uvjetima detektirati i lokalizirati lica na slikama, ali važno je naglasiti da još uvijek nema generalnog algoritma koji bi davao savršeno točne rezultate.

Houghova transformacija je tehnika izlučivanja značajki koja se koristi u analizi slike, u računalnom vidu. Prvotno je razvijana kako bi se analitički definirali oblici poput linije, kružnice i slično. U ovom radu bavit ćemo se generaliziranom Houghovom transformacijom koja nudi mogućnost definiranja i kompleksnijih oblika. Na ulazu sustava je digitalna fotografija lica, a na izlazu rezultat detekcije (uspješna ili neuspješna) i lokalizacije (pretpostavljena točka lica i vjerojatnost lokalizacije).

## 1.2 Pregled i opis srodnih rješenja

Detekcija lica jedna je od čestih tema u računalnom vidu, stoga već postoje razni pokušaji implementiranja takvog sustava, pogotovo u zadnja dva desetljeća. Jedno takvo rješenje ponudio je Sachin Sudhakar Farfade s dvoje kolega u svom radu [4]. U radu je opisan sustav kojem nije potreban model za orijentaciju, već detektira lica u različitim orijentacijama koristeći model temeljen na dubokim konvolucijskim neuronskim mrežama. Za razliku od sličnih metoda temeljenih na dubinskom učenju, nije potrebna ni segmentacija slike, regresija za granični obrub ili SVM klasifikator.

Ključni temelji tog rada su:

- moć dubokih konvolucijskih mreža za klasificiranje i izlučivanje značajki za učenje klasifikatora koji detektira lica
- minimiziranje računalne složenosti pojednostavljivanjem arhitekture detektora

Također je moguće detektirati lica iz različitih kuteva. Eksperimenti su pokazali slične ili još bolje rezultate u odnosu na metode koje su složenije i koje zahtijevaju referentne slike i značajke.

Viola i Jones u svom radu [6] prezentirali su implementaciju detekcije lica u realnom vremenu s vrlo niskom računalnom složenosti. Koristi se kaskadni detektor koji se sastoji od nekoliko jednostavnih do složenih klasifikatora za lica. Detektori kao takvi nude poprilično velik broj mogućnosti i širok spektar nadogradnje. Štoviše, kaskadni detektori primjenjuju se u mnogim uređajima kao što su pametni mobiteli i digitalne kamere. Problem kaskadnih detektora je što ne mogu dobro detektirati lica iz različitih kuteva i lica koja su dijelom pokrivena.

Analiza svojstvenih komponenata ( *engl. Principal component Analysis*, u nastavku PCA) koristi se kao alat za analiziranje podataka i kao predikcijski model. PCA možemo tumačiti kao metodu koja opisuje unutarnju strukturu podataka na način koji najbolje opisuje varijancu (varijanca može predstavljati najbitnije značajke, a rezultat metode su svojstveni vektori pa se može odrediti smjer raspršenja podataka).

PCA smanjuje dimenzionalnost izlučivanjem najmanjeg broja najbitnijih komponenata. Time se rješavamo redundancije i čuvamo varijancu u manjem broju koeficijenata. Ulazni skup pretvara se u karakteristične vektore (tzv. *eigenface* vektore<sup>1</sup>), zatim uzima K najboljih vektora od M ukupno, koliko ima ulaza. Uz uvjet da je K uvijek manji ili jednak broju M dobivamo K karakterističnih vektora. Time prepoznavanje ovisi o K a ne o M. Prepoznavanje se obrađuje kao težinska suma karakterističnih vektora. Usput je smanjen utjecaj šuma jer su odbačeni karakteristični vektori sa šumom.

---

<sup>1</sup> <http://www.scholarpedia.org/article/Eigenfaces>

## 1.3 Konceptualno rješenje zadatka

Problem detekcije i lokalizacije lica na slici rješavamo generaliziranom Houghovom transformacijom (*engl. Generalised Hough Transform*, u nastavku GHT).

Postoje dva slučaja GHT-a:

1. poseban slučaj:
  - transformacija s fiksnom orijentacijom i veličinom, kut = 0 i skaliranje = 1
2. opći slučaj:
  - transformacija s mijenjanjem kutova i veličina iz skupa raznih kutova i skaliranja

S obzirom da ne možemo očekivati da ćemo u praksi na slikama imati oblike fiksne orijentacije i veličine, koristimo opći slučaj generalizirane Houghove transformacije koji je u mogućnosti detektirati oblike koji imaju različitu orijentaciju (rotaciju  $\theta$ ) i veličinu (uniformno skaliranje  $s$ ) [3].

Postupak se sastoji od sljedećih koraka:

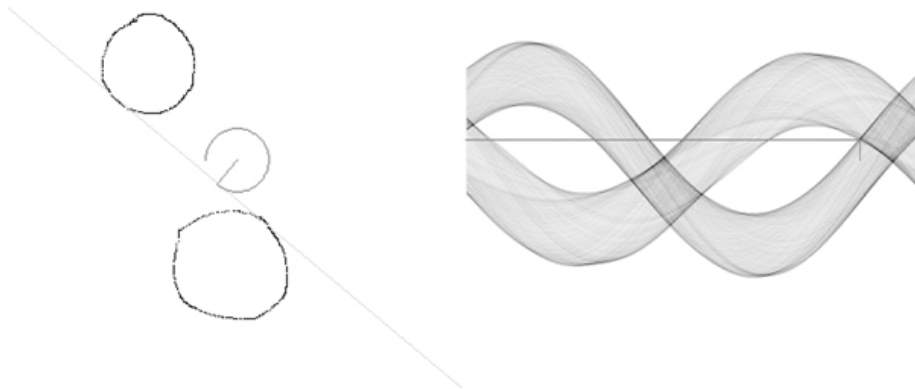
1. Pretprocesiranje
  - a. zadanom skupu referentnih slika dodijelimo referentne točke (specificiramo ciljani objekt)
  - b. iz zadanog skupa referentnih slika gradimo R-tablicu (referentne točke spremljene kao funkcija kuta  $\phi$ )
2. Detekcija i lokalizacija
  - a. na temelju R-tablice računamo pretpostavljene referentne točke te se provodi glasanje za njih
  - b. moguće lokacije kontura objekta dane su lokalnim maksimumima koji su dobiveni glasanjem
  - c. ukoliko je ta vrijednost veća od zadanog praga  $T$ , kontura objekta nalazi se kod pretpostavljene referentne točke

## 2. Postupak rješavanja zadatka

### 2.1 Houghova transformacija

Houghova transformacija se inicijalno razvila kao sredstvo za detekciju analitički poznato definiranih likova (linije, kružnice, elipse itd.). Transformacija preslikava ulaznu sliku u Houghov prostor parametara. U tom prostoru definiramo tzv. "bin"-ove. Metoda Houghove transformacije bazira se na sistemu glasanja, te će razni oblici u ulaznom prostoru glasati za razne "bin"-ove. Na primjer, linije iz ulaznog prostora se preslikavaju u točke u Houghovom prostoru i obrnuto.

Problem ovakvog pristupa je neograničenost Houghovog prostora. Vertikalne će se linije preslikati u beskonačno daleke točke. Rješenje nalazimo u prijelazu na polarne koordinate, gdje svaka točka dodaje sinusoidu u prostoru parametara. Prisutnost šuma uzrokovat će teže nalaženje vrha u prostoru parametara, međutim mala je vjerojatnost da će višestruki šum glasati za isti model. Naravno, jači šum prouzročit će teže nalaženje vrha te kažemo da je Houghova transformacija djelomično otporna na šum [2].



Slika 1. Ulazni prostor (lijevo) i Houghov prostor parametara u polarnim koordinatama (desno)

Prednosti Houghove transformacije:

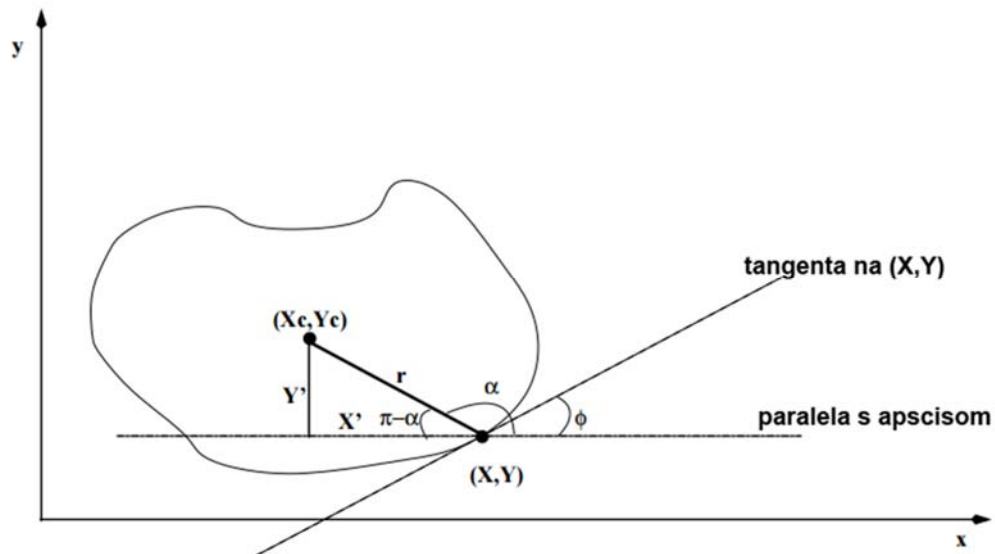
- dobro radi s nelokaliziranim podacima
- nalazi višestruku pojavu modela u jednom prolazu
- djelomično otporan na šum

Nedostatci Houghove transformacije:

- složenost pretrage eksponencijalno ovisi o broju parametara modela
- neciljani oblici mogu izazvati značajan broj glasova u prostoru parametara

## 2.2 Postupak Generalizirane Houghove transformacije

Generaliziranjem Houghove transformacije nastala je Generalizirana Houghova transformacija (*engl. Generalised Hough Transform*, u nastavku GHT) s kojom je moguće detektirati kompleksne oblike koji nemaju analitički poznatu formu [1].



Slika 2. Tangenta na koordinatu točke oblika i njezin gradijentni kut  $\phi$

Razlikujemo dva slučaja GHT-a:

1. Poseban slučaj - transformacija s fiksnom orijentacijom (kut = 0) i veličinom (skaliranje = 1)
2. Opći slučaj - transformacija s mijenjanjem kutova i veličina

## 2.2.1 Poseban slučaj GHT-a

Ovakvu transformaciju još nazivamo i specijalnom jer se podrazumijeva da oblik kojeg želimo detektirati ima jednaku orijentaciju i veličinu kao i oblik koji nam služi kao referentni oblik. U praksi takav slučaj nije vjerojatan no može biti pokazatelj kako transformacija funkcionira. Također, manjim preinakama jednostavne GHT dobivamo složenu GHT.

$$x = x_c + x' \rightarrow x_c = x - x'$$

$$y = y_c + y' \rightarrow y_c = y - y'$$

$$\cos(\pi - \alpha) = \frac{y'}{r} \rightarrow y' = r \cos(\pi - \alpha) = -r \sin(\alpha)$$

$$\sin(\pi - \alpha) = \frac{x'}{r} \rightarrow x' = r \sin(\pi - \alpha) = -r \cos(\alpha)$$

Iz čega slijedi:

$$x_c = x + r \cos(\alpha), y_c = y + r \sin(\alpha)$$

### Faza 1/2: Pretprocesiranje

- 1) Odabir referentne točke  $(x_c, y_c)$  (obično centar referentne slike).
- 2) Crtanje linije od referentne točke do točke ruba referentne slike.
- 3) Izračun kuta  $\phi$  (prikazan na gornjoj slici).
- 4) Izrada R-tablice (obično rječnik) koja sadrži za svaki kut  $\phi$  niz parova udaljenosti do ruba  $r$  i kuta  $\alpha$ :

$$\phi_1 : (r_1^1, \alpha_1^1), (r_2^1, \alpha_2^1), \dots$$

$$\phi_2 : (r_1^2, \alpha_1^2), (r_2^2, \alpha_2^2), \dots$$

$$\dots$$
$$\phi_n : (r_1^n, \alpha_1^n), (r_2^n, \alpha_2^n), \dots$$

Pomoću R-tablice koja je nastala obradom referentne slike, moguć je izračun pretpostavljene lokacije referentne točke na slici koju analiziramo odnosno u kojoj tražimo referentni oblik.



## Faza 2/2: Detekcija

- 1) Detekcija se vrši nad slikom koja nije referentna. Stvori se matrica jednakih dimenzija kao ta slika. Svaka točka u toj matrici  $P$  se u početku smatra kao pretpostavljena referentna točka detekcije odnosno matrica se gleda kao graf i vrijednost svake koordinate/elementa je u početku nula.
- 2) Kako bi proces detekcije bio jednostavniji, promatraju se samo točke slike koje su na jasnim rubovima (mjesto gdje je velika razlika između vrijednosti piksela) unutar slike. Za svaku takvu točku  $(x, y)$ :
  - a. Koristeći gradijentni kut  $\theta$  od  $(x, y)$ , dohvatiti sve  $(r, \alpha)$  parove za taj kut.
  - b. Za svaki  $(r, \alpha)$  par potrebno je izračunati pretpostavljenu referentnu točku:
$$x_c = x + r \cos(\alpha)$$
$$y_c = y + r \sin(\alpha)$$
  - c. U koordinati  $(x_c, y_c)$  u  $P$  povećava se vrijednost za 1:
$$P[x_c][y_c]++$$
- 3) Veća vrijednost točke  $P[x_c][y_c]$  znači veća pretpostavka da upravo u toj točki se nalazi referentna točka traženog oblika odnosno u tom dijelu slike se nalazi traženi oblik. Obično se postavi neki prag  $T$  i gledaju se samo točke koje imaju vrijednost  $> T$ , ostale točke su nepouzdana.

### 2.2.2 Opći slučaj GHT-a

Pretpostavka da je traženi oblik podložen određenoj rotaciji  $\theta$  i skaliranju  $s$ :

$$(x', y') \rightarrow (x'', y'')$$

$$x'' = s(x' \cos(\theta) - y' \sin(\theta))$$

$$y'' = s(x' \sin(\theta) + y' \cos(\theta))$$

Zamjenom  $x'$  s  $x''$  i  $y'$  s  $y''$  dobivamo:

$$x_c = x - x'' \rightarrow x_c = x - s(x' \cos(\theta) - y' \sin(\theta))$$

$$y_c = y - y'' \rightarrow y_c = y - s(x' \sin(\theta) + y' \cos(\theta))$$

Postupak detekcije oblika koji to uzima u obzir je time robusniji i može se koristiti u praktične svrhe.

## Faza 1/2: Pretprocesiranje

Pretprocesiranje općeg slučaja jednako je pretprocesiranju posebnog slučaja.

## Faza 2/2: Detekcija

- 1) U posebnom slučaju akumulacija se vrši nad pojedinom koordinatom, odnosno povećava se vrijednost pojedine koordinate matrice. U općem slučaju mora se uzeti u obzir kut rotacije i skaliranje, stoga se dodaju dvije nove dimenzije za svaku koordinatu koja poprima oblik  $(x_c, y_c, \theta, s)$ . Svaka takva koordinata/element ima početnu vrijednost nula. Ako se želi izbjeći proširivanje koordinate, može se ostaviti  $(x_c, y_c)$  oblik i stvoriti posebnu matricu za svaku kombinaciju kuta  $\theta$  i skaliranja  $s$ . Npr. ako želimo koristiti kutove  $\theta_1, \theta_2, \dots$  i skaliranja  $s_1, s_2, \dots$ , potrebno je napraviti posebnu matricu za  $(\theta_1, s_1), (\theta_1, s_2), (\theta_1, s_3), \dots, (\theta_m, s_n)$  i zatim vršiti akumulacije nad elementima svih tih ukupno  $m * n$  matrica. Takav način je prikladan ako se žele pokazati vrijednosti za svaki  $(\theta, s)$  slučaj.
- 2) Kako bi proces detekcije bio jednostavniji, promatraju se samo točke slike koje su na jasnim rubovima (mjesto gdje je velika razlika između vrijednosti piksela) unutar slike. Za svaku takvu točku  $(x, y)$ :

- a. Koristeći gradijentni kut  $\phi$  od  $(x, y)$ , dohvatiti sve  $(r, \alpha)$  parove za taj kut.
- b. Za svaki  $(r, \alpha)$  par potrebno je izračunati pretpostavljenu referentnu točku:

$$\begin{aligned}x' &= r \cos(\alpha) \\y' &= r \sin(\alpha) \\for(\theta &= \theta_1; \theta \leq \theta_m; \theta++) \\for(s &= s_1; s \leq s_n; s++) \\x_c &= x - s(x' \cos(\theta) - y' \sin(\theta)) \\y_c &= y - s(x' \sin(\theta) + y' \cos(\theta)) \\P[x_c][y_c][\theta][s] &++\end{aligned}$$

Veća vrijednost točke  $P[x_c][y_c][\theta][s]$  znači veća pretpostavka da upravo u toj točki se nalazi referentna točka traženog oblika odnosno u tom dijelu slike za određeni kut i određeno skaliranje nalazi se traženi oblik. Obično se postavi neki prag  $T$  i gledaju se samo točke koje imaju vrijednost  $> T$ , ostale točke su nepouzdana.

### 2.2.3 Prednosti, nedostaci i moguća poboljšanja GHT-a

#### Prednosti Generalizirane Houghove transformacije:

- GHT se može koristiti za detekciju proizvoljnih oblika.
- Robusna je za djelomično prikazane ili malo deformirane oblike.
- Robusna je za prisutnost drugih oblika (linije, krivulje itd.) koje nemaju veze s traženim oblikom.
- Tolerira prisutnost šuma.
- Može detektirati višestruko pojavljivanje traženog oblika u istoj slici.

#### Nedostaci Generalizirane Houghove transformacije:

- Zahtijeva zauzeće veće količine memorije.
- Procesorski intenzivna, no postupak se može paralelizirati.

#### Moguća poboljšanja

- Hijerarhijske reprezentacije: prvi pronalazak koristeći Houghovu rezolucijsku matricu i zatim selektivno proširivanje dijelova matrice koji imaju veće vrijednosti. [5]
- Projekcije: umjesto višedimenzionalnih matrica koristiti nekoliko projekcija koje koriste koordinate (npr.  $(x_c, y_c)$ ,  $(y_c, \theta)$ ,  $(\theta, s)$ ,  $(s, x_c)$ ) i zatim pronaći konzistentne 'vrhove'. [7] [8]

## 3. Ispitivanje rješenja

### 3.1 Ispitna baza

Postupak koristi dvije vrste slika:

1. Referentne slike
  - slike koje sadrže samo lica pomoću koje se gradi R-tablica
2. Ispitne slike
  - slike koje sadrže lice (ili lica) koje se želi detektirati i lokalizirati

#### 3.1.1 Referentne slike

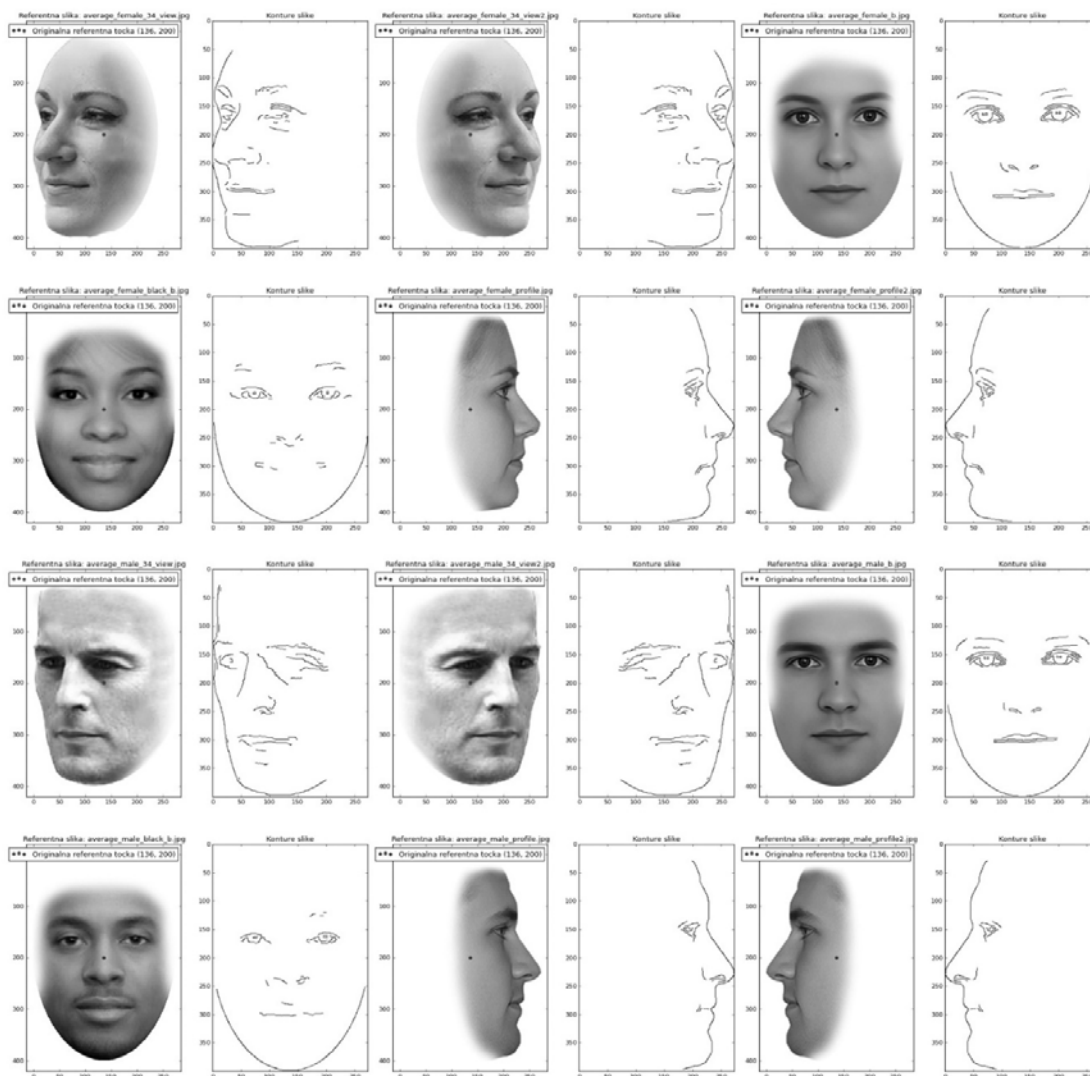
Kako bi postupak uspješno prepoznao na slici (novo) lice potrebna mu je R-tablica gradijentnih kutova i vektora nastala obradom referentnih slika različitih lica u raznim pozama. U ovom projektu za demonstraciju postupka koristile su se slike lica različitih poza tri muškarca i tri žene.

Korištene poze su sljedeće: krupni kadar, iz profila i  $\frac{3}{4}$  pogled. Robusnost postupka se, osim prethodnog, povećala i korištenjem slike lica krupnog kadra i iz profila koje su nastale usrednjavanjem mnogo lica<sup>2</sup>. Pretpostavka je da će takve slike pridonijeti boljoj detekciji i lokalizaciji nepoznatih lica.

Za stvaranje R-tablice iz svake slike lica potrebno je bilo dobiti samo konture lica. To se činilo pretvorbom slike u crno-bijelu verziju i potom uporabom *canny* metode. Rezultat toga je *true/false* matrica istih dimenzija kao slika lica koja umjesto vrijednosti piksela sadrži informaciju nalazi li se ili ne nalazi u toj točki rub. Prikaz crno-bijelih verzija korištenih referentnih slika i njihovih kontura je prikazan na Slici 3. Slike su se ručno obradile na način da se istakne samo lice kako postupak ne bi bio ometan nebitnim dijelovima (kosa, vrat). Također, rubovi koje smatramo nebitnim su zaglađeni kako ih *canny* metoda ne bi 'detektirala' kao konture lica što je bitno u slikama lica slikanih iz profila.

---

<sup>2</sup> Projekt BeautyCheck, Prirodoslovni fakultet, Sveučilište u Regensburgu, Njemačka: [http://www.uni-regensburg.de/Fakultaeten/phil\\_Fak\\_II/Psychologie/Psy\\_II/beautycheck/english/durchschnittsgesichter/durchschnittsgesichter.htm](http://www.uni-regensburg.de/Fakultaeten/phil_Fak_II/Psychologie/Psy_II/beautycheck/english/durchschnittsgesichter/durchschnittsgesichter.htm)



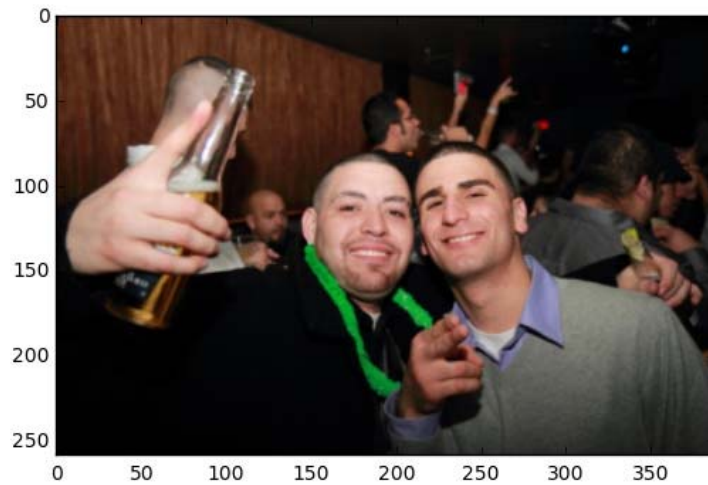
Slika 3. Korištene referentne slike i njihove konture

Konture lica su se zatim koristile u postupku GHT-a odnosno u njegovoj 1. fazi 'Preprocesiranje'.

### 3.1.2 Ispitne slike

Kako bi se postupak ispitao potrebno je bilo koristiti bazu slika lica s različitim položajima, pozama i veličinama. U tu svrhu odabrana je ispitna baza *Annotated Faces in the Wild* (AFW)<sup>3</sup>

Pomoću nje se vršila i evaluacija uspješnosti detekcije i lokalizacije slika korištenog postupka. Primjer slike je prikazan na Slici 4. i korišten u opisu (među)rezultata postupka.



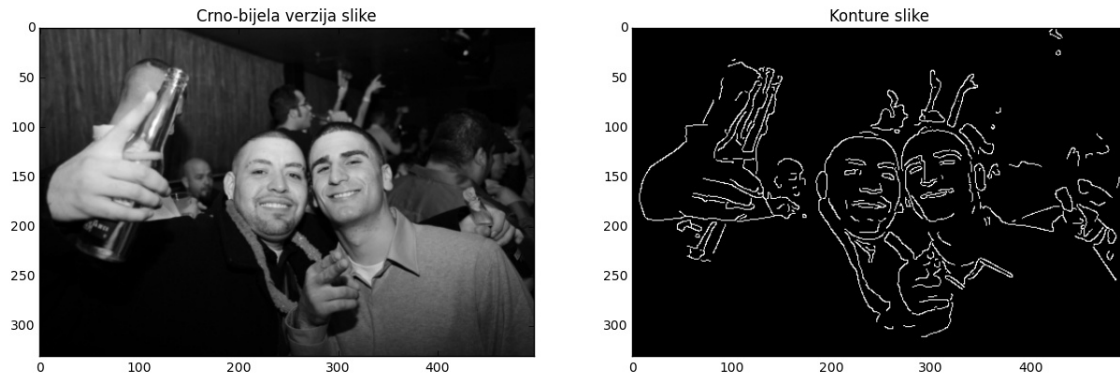
Slika 4. Primjer ispitne slike (4237203680.jpg)

---

<sup>3</sup> Face Detection, Pose Estimation and Landmark Localization in the Wild, 2012., <https://www.ics.uci.edu/~xzhu/face/>

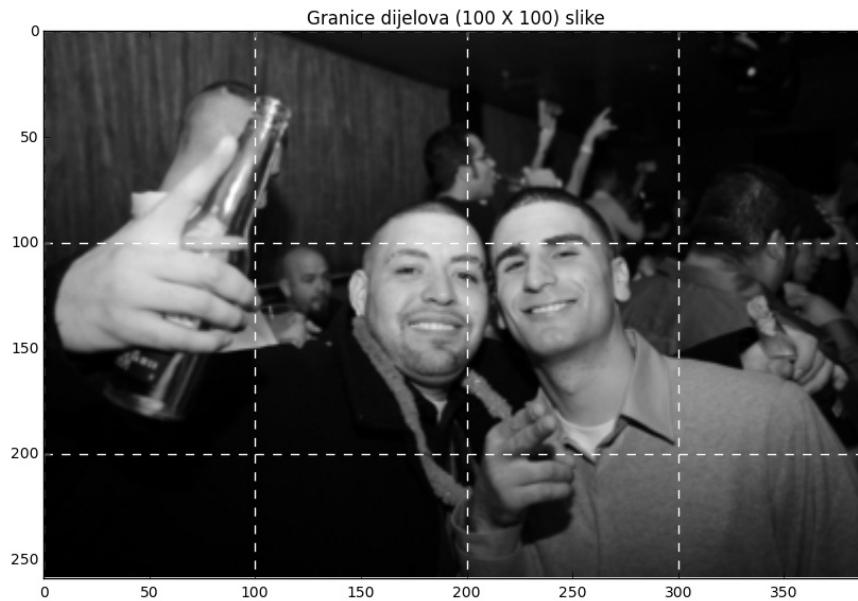
## 3.2 Rezultati učenja i ispitivanja

Slično kao i u slučaju punjenja R-tablice, u ispitnim slikama nas zanimaju samo konture. Primjer jedne slike, njezine crno-bijele verzije i njezinih kontura je prikazan na Slici 5.



Slika 5. Ispitna slika u crno bijeloj i canny (konture) inačici

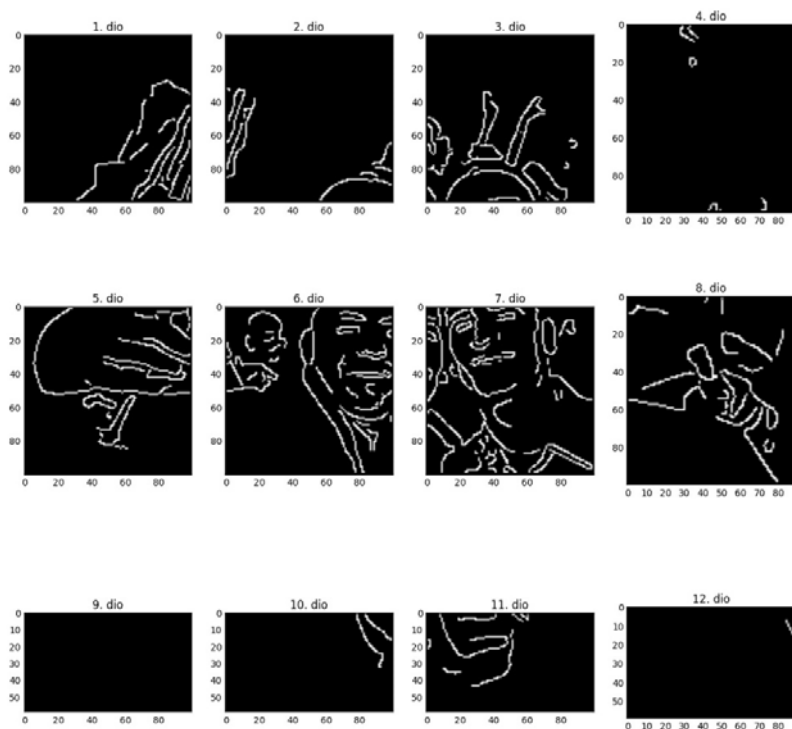
Umjesto primjene postupka GHT-a nad cijelom ispitnom slikom, slika se podijelila na više dijelova (Slika 6. i Slika 7.), iz svakog dijela su uzete konture (Slika 8.) i konačno nad svakom od njih je proveden postupak GHT-a.



Slika 6. Granice dijelova slike



Slika 7. Dijelovi slike



Slika 8. Konture dijelova slike



Rezultat za pojedini dio slike je jedna akumulacijska matrica istih dimenzija kao i dio slike. Njezine vrijednosti možemo promatrati kao svojevrzne vjerojatnosti da se u koordinatama tih vrijednosti nalazi pretpostavljena točka lica. Što je vjerojatnost veća, postupak je sigurniji da se upravo tamo nalazi tražena točka. Zato se odabiru točke akumulacijskih matrica koje imaju najveću vrijednost. Njihove koordinate i vrijednosti po svakom dijelu prikazane su na Slici 9.



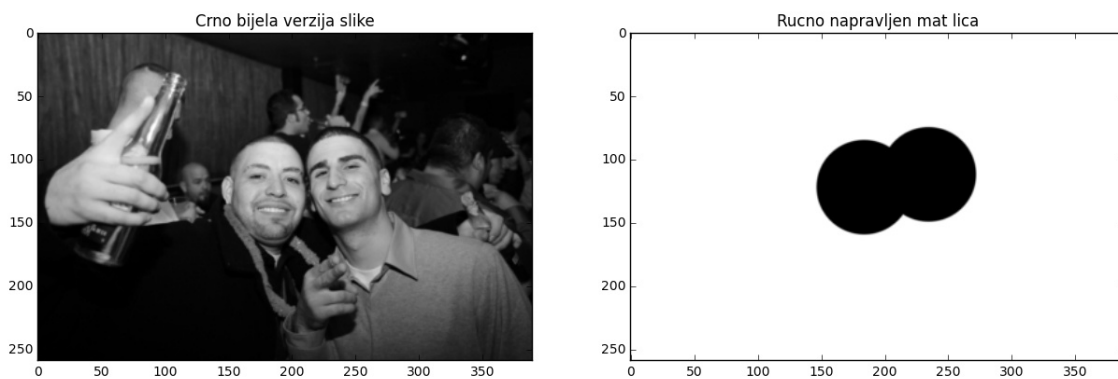
Slika 9. Pretpostavljene točke lica po svakom dijelu slike

Znajući način na koji se slika podijelila na više dijelova, najbolje točke pojedinih dijelova se mogu prikazati na originalnoj slici u kojoj se želi detektirati i lokalizirati lice (Slika 10.).



Slika 10. Pretpostavljene točke dijelova u ukupnoj slici

Umjesto ručnog pregleda koje točke su unutar lica, a koje izvan, napravljen je mat lica (Slika 11.). Pomoću njega moguće je automatski odrediti koje su točke izvan, a koje unutar lica, kao i njihov broj što se koristilo prilikom obrade više slika i evaluacije rezultata.



Slika 11. Originalna slika (lijevo) i mat lica (desno)

Pregledom vrijednosti točaka mogu se uočiti značajne razlike. Jasno je da će za neki mali prag  $T$  biti mnogo točaka koje se ne nalaze unutar lica ako se u svakom podijeljenom dijelu slike nije nalazilo lice. Podizanjem praga  $T$  preostat će samo točke u koje je postupak sigurniji. Potrebno je pronaći neki optimalni prag  $T$  koji bi u ovisnosti o potrebi uklonio pogrešno lokalizirane točke lica ili maksimizirao broj uspješno lokaliziranih lica. Također ne smije biti previsok kako se ne bi sve točke proglasile irelevantnima. Zato se postepeno povećavao prag  $T$  od nule do neke granice (sveukupna maksimalna vrijednost + neka konstanta) i dobio se pregled sljedećih vrijednosti:

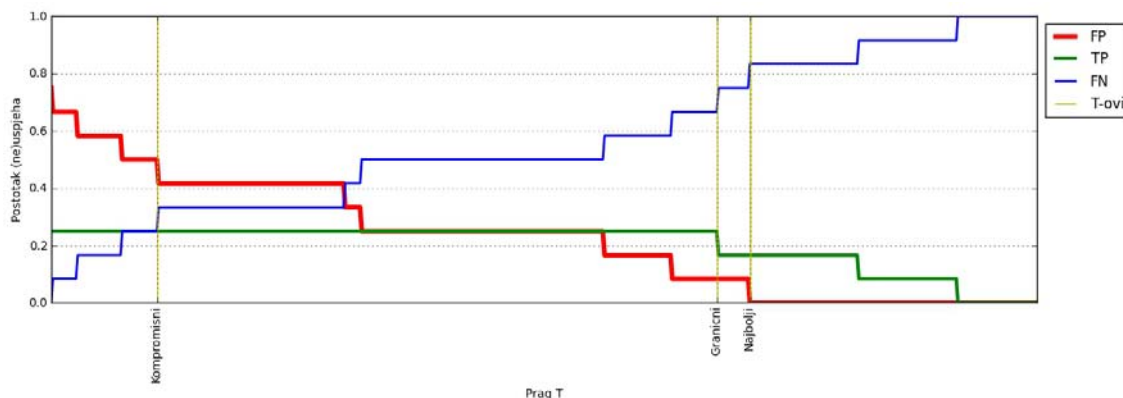
- Neuspješna lokalizacija (FP, *False Positive*):  
- nađene točke se ne nalaze unutar lica
- Uspješna lokalizacija (TP, *True Positive*):  
- nađene točke se nalaze unutar lica
- Neuspješna detekcija (FN, *False Negative*):  
- nađene točke imaju vrijednosti manje od nekog praga, smatra se da postupak nije detektirao lice na slici koja sadrži lice

Slika 12 prikazuje te vrijednosti ovisne o nekom pragu. Izračunate su i tri različita praga  $T$ :

- Kompromisni prag:  
- nakon ovog praga broj neuspješno detektiranih točaka je veći od uspješno lokaliziranih točaka
- Granični prag:  
- nakon ovog praga broj uspješno lokaliziranih točaka nije više maksimalan
- Najbolji prag:  
- nakon ovog praga broj pogrešno lokaliziranih točaka se svodi na minimum

Na Slici 12. uz vrijednosti uspješnosti prikazani su i izračunati pragovi za korištenu sliku.

Povećavanje pragova, odnosno korištenje tri navedena praga je prikazano na Slici 13. Jasno se može vidjeti koji su nedostaci i prednosti korištenja manjeg odnosno većeg praga  $T$ .



Slika 12.

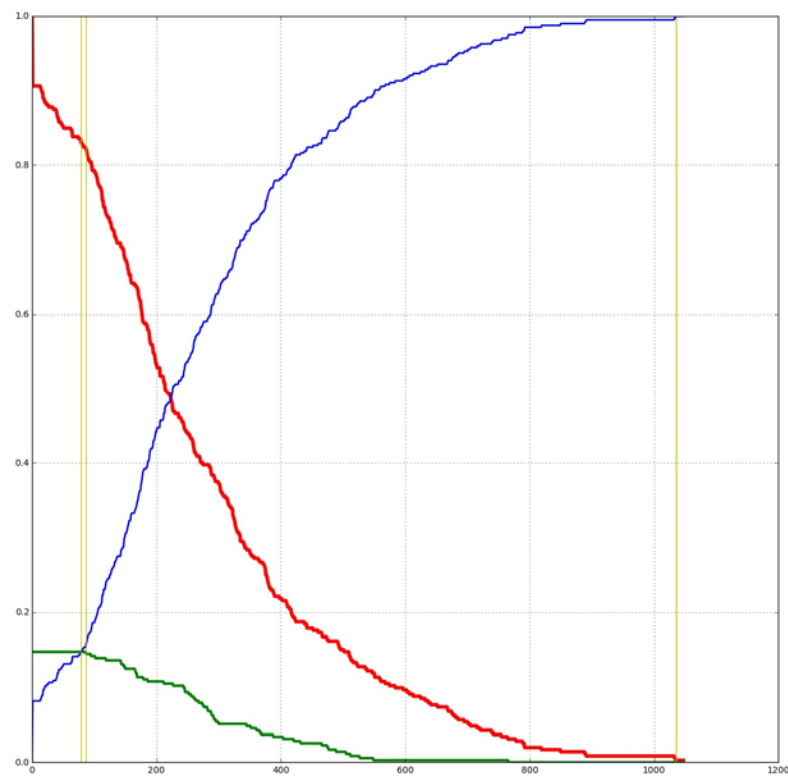


Slika 13.

### 3.3 Analiza rezultata

U svrhu evaluacije postupka korištena je ispitna baza AFW odnosno dio njezinih slika. Korišteno je ukupno 31 slika. Svaka slika je obrađena na prethodno opisani način. Za svaku sliku su izračunate najbolje točke dijelova slike i optimalni pragovi. Nakon obrade pojedine slike, takvi podaci su se spremali u obliku *pickle*<sup>4</sup> datoteke kako bi ih mogli koristiti prilikom evaluacije.

Nakon obrade svih ispitnih slika, napravljen je graf vrijednosti (ne)uspjeha ovisan o prag  $T$  prikazan na Slici 14. Na njoj su prikazani i izračunati optimalni pragovi.



Slika 14

Na temelju optimalnih pragova ponovno su promatrane točke pojedinih slika i odredio se broj neuspješno lokaliziranih lica (FP), broj uspješnog lokaliziranih lica (TP) i broj nedetektiranih lica (FN).

Tablica 1. prikazuje imena datoteka ispitnih slika i vrijednosti uspjeha ovisne o optimalnim pragovima.

<sup>4</sup> <https://docs.python.org/2/library/pickle.html>

	Kompromisni T = 79			Granični T = 87			Najbolji T = 1035		
Ime slike	FP	TP	FN	FP	TP	FN	FP	TP	FN
120563545.jpg	10	2	0	10	2	0	0	0	12
1648807314.jpg	11	1	0	11	1	0	0	0	12
174692320.jpg	8	3	5	8	3	5	0	0	16
2080551997.jpg	10	2	0	9	2	1	0	0	12
2122067003.jpg	12	0	0	12	0	0	0	0	12
2201628776.jpg	11	1	0	10	1	1	0	0	12
2288136183.jpg	12	0	0	12	0	0	0	0	12
2542662563.jpg	6	2	4	6	2	4	0	0	12
261068.jpg	11	1	0	11	1	0	0	0	12
2653111362.jpg	11	1	0	11	1	0	0	0	12
281972218.jpg	14	1	1	14	1	1	0	0	16
346731834.jpg	5	2	5	5	2	5	0	0	12
368987306.jpg	15	0	1	15	0	1	0	0	16
3929640120.jpg	7	2	3	7	2	3	0	0	12
399829006.jpg	11	1	0	10	1	1	0	0	12
40014967.jpg	10	2	4	10	1	5	0	0	16
4009440741.jpg	10	2	0	10	2	0	0	0	12
4237203680.jpg	5	3	4	5	3	4	0	0	12
4253907822.jpg	8	2	6	8	2	6	0	0	16
437595409.jpg	12	0	0	12	0	0	1	0	11
442651885.jpg	11	0	1	11	0	1	0	0	12
447910249.jpg	14	1	1	14	1	1	0	0	16
4739664721.jpg	9	3	0	9	3	0	0	0	12
642820626.jpg	9	1	2	9	1	2	0	0	12
70037463.jpg	10	2	0	10	2	0	0	0	12
878985234.jpg	8	1	3	8	1	3	0	0	12
9545523490.jpg	5	5	2	5	5	2	0	0	12
955659370.jpg	10	1	5	10	1	5	0	0	16
biber.jpg	4	5	3	4	5	3	0	0	12
face_1_test9.jpg	2	4	10	2	4	10	0	0	16
forrest_gump.jpg	11	1	0	11	1	0	0	0	12

Tablica 1. Izračunati su optimalni pragovi na temelju svih slika i zatim korišteni za pregled (ne)uspjeha

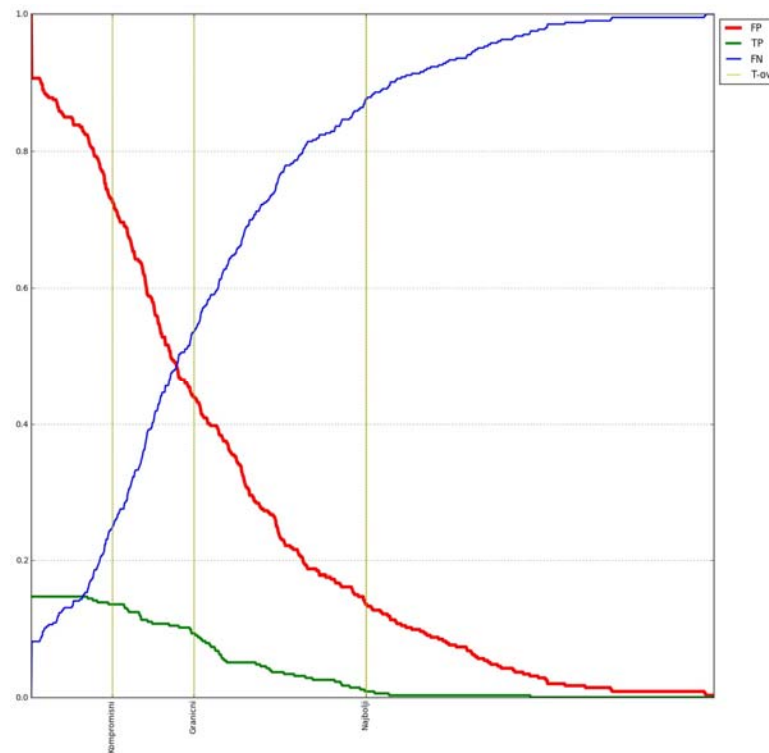
Pregledom prethodnog grafa i tablice zaključeno je da za korišteni set referentnih slika, ispitnih slika i korištenih parametara postupak generalizirane Houghove transformacije nije zadovoljavajuće ispunio zadatak detekcije i lokalizacije lica u ispitnim slikama.

Situacija se malo poboljšava ako se koriste optimalni pragovi dobiveni usrednjavanjem optimalnih pragova pojedinih slika. Tablica 2. prikazuje imena datoteka ispitnih slika i vrijednosti uspjeha ovisne o usrednjenim optimalnim pragovima.

	Kompromisni T = 124			Granični T = 249			Najbolji T = 514		
Ime slike	FP	TP	FN	FP	TP	FN	FP	TP	FN
120563545.jpg	8	2	2	3	2	7	0	0	12
1648807314.jpg	10	1	1	7	1	4	1	0	11
174692320.jpg	8	3	5	6	1	9	0	0	16
2080551997.jpg	9	2	1	2	1	9	0	0	12
2122067003.jpg	10	0	2	1	0	11	0	0	12
2201628776.jpg	10	1	1	7	1	4	1	0	11
2288136183.jpg	12	0	0	11	0	1	4	0	8
2542662563.jpg	5	2	5	1	2	9	0	0	12
261068.jpg	11	1	0	9	1	2	2	0	10
2653111362.jpg	11	1	0	11	1	0	9	1	2
281972218.jpg	11	1	4	6	1	9	1	0	15
346731834.jpg	5	2	5	4	1	7	0	0	12
368987306.jpg	12	0	4	4	0	12	0	0	16
3929640120.jpg	7	2	3	6	2	4	0	1	11
399829006.jpg	10	1	1	4	1	7	0	0	12
40014967.jpg	10	1	5	7	1	8	2	0	14
4009440741.jpg	9	2	1	6	2	4	1	0	11
4237203680.jpg	5	3	4	3	3	6	0	1	11
4253907822.jpg	8	2	6	4	0	12	0	0	16
437595409.jpg	12	0	0	12	0	0	11	0	1
442651885.jpg	10	0	2	6	0	6	2	0	10
447910249.jpg	11	1	4	9	1	6	3	0	13
4739664721.jpg	7	3	2	1	1	10	0	0	12
642820626.jpg	8	1	3	7	0	5	2	0	10
70037463.jpg	7	2	3	4	2	6	0	0	12
878985234.jpg	6	1	5	1	1	10	0	0	12
9545523490.jpg	4	3	5	1	2	9	0	0	12
955659370.jpg	7	1	8	0	1	15	0	0	16
biber.jpg	1	5	6	0	1	11	0	0	12
face_1_test9.jpg	2	3	11	1	2	13	0	0	16
forrest_gump.jpg	11	1	0	11	1	0	9	0	3

Tablica 2. Izračunati su prosjeci optimalnih pragova po pojedinim slikama i zatim korišteni za pregled (ne)uspjeha

Slika 15. prikazuje graf (ne)uspjeha s usrednjenim optimalnim pragovima.



Slika 15.

Mogući razlozi zašto je detekcija/lokalizacija lica u slici u prosjeku neuspješna:

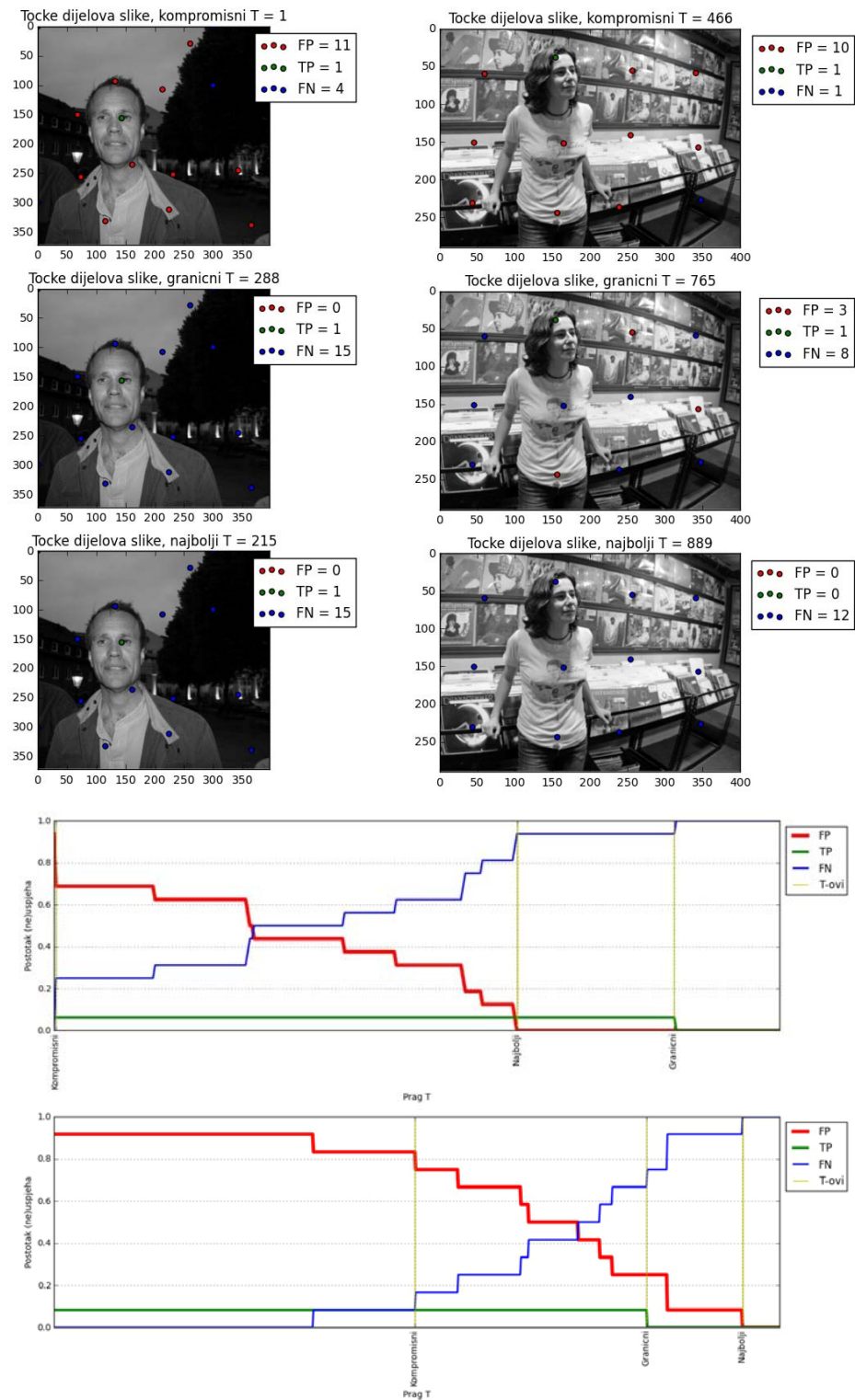
- ispitna slika je razdijeljena na dijelove premalih dimenzija što je uzrokovalo moguću podjelu jednog lica unutar slike na više dijelova čime se smanjila sigurnost postupka za lokalizaciju traženog oblika.
- postupak GHT-a nije prikladan za traženje takvih oblika i dijelovi slike koji sadrže različite oblike odnosno sadrže veći broj kontura akumuliraju točku veće vrijednosti zbog prisustva velikog broja gradijenata.
- korišteni parametri nisu pogodovali postupku.

Moguća rješenja za pretpostavljene razloge:

- moguće je da bi postupak GHT-a uspješnije lokalizirao lice kad bi se on odvijao nad cijelom slikom, a ne samo nad njezinim dijelovima. Ako se koristi način podjele slike potrebno je možda povećati dimenziju dijela slike čime se smanjuje šansa za podjelom lica na više dijelova. Mogući način je i korištenje putujućeg prozora po slici no zbog procesorske zahtijevnosti postupka on nije preporučljiv.
- primijećeno je da dijelovi slike koji sadrže više kontura redovito akumuliraju veću vrijednost točke. Moguće je možda pronaći parametre koji bi te konture sveli na minimum, ali i da se ne ugrozi detekcija kontura traženog lica.
- moguće je mijenjati mnoge parametre. Neki od najbitnijih su pragovi za određivanje kontura lica, kutovi i skaliranja, i sličnosti između vektora R-tablice. Promjenom tih parametara moguće je možda dobiti bolje rezultate.



Osim navedenog, moguće je i koristiti unaprjeđenja postupka GHT-a opisan u poglavlju 2.2.3 U nastavku u Slici 16. prikazani su primjeri uspješne i neuspješne lokalizacije ovisne o optimalnim pragovima.



Slika 16.

## 4. Opis programske implementacije rješenja

### 4.1 Opis programske implementacije

Postupak se implementirao pomoću programskog jezika Python<sup>5</sup> (verzija 2.7.10). Korišten je znanstveni skup biblioteka otvorenog koda SciPy<sup>6</sup>, odnosno njegova distribucija Anaconda<sup>7</sup>. Unutar tog skupa koristile su se metode sljedećih (pod)biblioteka:

- NumPy - za stvaranje i obradu matrica
- IPython - za izradu interaktivne Python bilježnice
- Matplotlib.pyplot - za prikaz i spremanje slika grafova
- SciPy.misc - za učitavanje i obradu slika
- skimage.feature - za dobivanje kontura slike

Potrebno je imati instalirane navedene biblioteke (osim biblioteke IPython u slučaju da se kod ne izvršava pomoću bilježnice već pomoću čistog Python koda) kako bi se program uspješno pokrenuo. Preporuka je instalirati distribuciju Anacondu za potpuni pregled i korištenje implementacije.

Osim navedenih eksternih biblioteka koristile su se sljedeće standardne biblioteke:

- os - za pristup datotekama direktorija
- math - za korištenje matematičkih funkcija
- threading - za višedretveni načina rada
- time - za mjerenje potrebno vremena nekih postupaka
- sys - za izračun zauzeća nekih objekata
- pickle - za spremanje/dohvaćanje objekata u obliku datoteka

Izrada programske implementacije se vršila pomoću interaktivne IPython<sup>8</sup> bilježnice (priložene uz dokumentaciju na DVD-u). Ona omogućuje podjelu programskog koda na više blokova (*cell*) koji se mogu nezavisno pokretati odnosno interpretacija programskog koda se ne izvršava slijedno po svim dijelovima nego je moguće odabrati koji će se točno dijelovi pokrenuti. Pokretanjem pojedinog bloka ispod njega se prikazuju rezultati za taj blok. Iako se

---

<sup>5</sup> <https://www.python.org/>

<sup>6</sup> <http://www.scipy.org/install.html>

<sup>7</sup> <https://www.continuum.io/why-anaconda>

<sup>8</sup> <http://ipython.org/notebook.html>

blokovi pokreću odvojeno, funkcije i Interpretirane varijable pojedinih blokova su dostupne svim naredno pokrenutim blokovima. Time se postiže visoki stupanj interaktivnosti jer je moguće primjerice napisati istu metodu u više verzija i staviti po jednu u pojedini blok. Također, dobiva se mogućnost brzog odabira drugačije verzije funkcije, varijable ili izračuna bez potrebe za brisanjem.

Primjerice, zbog potrebe mijenjanja nekih ključnih varijabli na početku programskog koda, zbog neke varijable na kraju koda nije potrebno pričekati da se kod između njih interpretira i izvrši, dovoljno je promijeniti samo u početnom bloku te početne varijable.

Zbog potrebe stalnog mijenjanja nekih konstanti i prilagodbi metoda takav način se intenzivno koristio u programskoj implementaciji.

IPython bilježnica se osim takve mogućnosti rada koristila i zato što omogućuje pisanje teksta između blokova programskog koda. Uz običan tekst moguće je pisati i HTML kod kao i LaTeX kod. Tekst se također može podijeliti u više blokova i pokretati zasebno. Pokretanjem bloka kodovi se automatski renderiraju i prikazuju. Time se omogućilo pisanje dokumentacije implementacije izravno uz programski kod. Osim toga takav način omogućuje jasnu i dinamičnu prezentaciju programskog koda.

Na priloženom DVD-u uz dokumentaciji nalazi se i HTML verzija IPython bilježnica. To je statički prikaz bilježnice koji nema interaktivnost kao IPython bilježnica, ali se može pregledati u web pregledniku bez potrebe za instaliranjem IPython sučelja.

Osim u obliku bilježnice priložen je i čisti Python kod konvertiran iz bilježnice.

Programski kod podijeljen je na jasne dijelove:

- Biblioteke - korištene biblioteke, uz svaku je napisan komentar za što služi
- Konstante - korištene konstante, uz svaku je napisan komentar za što služi
- Pomoćne dretve - dretve pomoću koje se ostvarila višedretvenost, svaka ima detaljno objašnjenje rada
- Pomoćne funkcije - funkcije s različitim (i opisanim) namjenama, podijeljene su na:
  - Pomoćne funkcije obrade - obrađuju podatke i vraćaju rezultate
  - Pomoćne funkcije prikaza - obrađuju podatke i prikazuju rezultate kao grafove
  - Glavna funkcija obrade - sveukupna obrada ispitne slike i mogući prikaz

Zatim slijede dijelovi postupka:

- Prikaz referentnih slika i njihovih rubova - prikazuju se referentne slike i njihove konture
- Stvaranje R-tablice - izgradnja R-tablice
- Primjer obrade jedne ispitne slike - prikaz i obrada jedne ispitne slike, sadrži sljedeće poddijelove:
  - Prikaz ispitne slike
  - Ispitna slika u crno bijeloj i *canny* (konture) inačici
  - Pregled načina podjele ispitne slike na dijelove
  - Pregled dijelova ispitne slike
  - Pregled kontura dijelova ispitne slike
  - Akumulacija nad svakim dijelom ili dohvaćanje već gotovih podataka ako je već prije izvršena i prikaz nađenih točaka i njihovih vrijednosti po dijelu ispitne slike
  - Prikaz nađenih točaka dijelova na ukupnoj ispitnoj slici

- Prikaz ručno napravljenog mata lica pomoću kojeg će se odrediti koje točke se nalaze unutar lica
- Prikaz grafa (ne)uspjeha ovisnih o korištenom pragu  $T$  i točaka koje su (ne)uspješno detektirale i lokalizirale lice ovisno o različitim izračunatim pragovima  $T$
- Obrada ispitnih slika - obrada ispitnih slika iz nekog skupa slika na prethodno opisan način (pomoću objedinjene funkcije)
- Evaluacija rezultata obrade ispitnih slika - za svaku ispitnu sliku se dobila i spremila informacija o najboljim točkama i njihovim vrijednostima. Korištenjem svih informacija ispitnih slika, napravljen je i prikazan ukupan graf (ne)uspjeha ovisnih o korištenom pragu  $T$ . Kao i za svaku pojedinu sliku izračunati su optimalni pragovi i zatim korišteni u svakoj ispitnoj slici. Time se dobila i prikazala tablica uspješnosti za svaku sliku.
- Konačna statistika - prikaz konačne statistike

## 4.2 Korištenje programske implementacije

Za korištenje programske implementacije potrebno je sljedeće:

- Instalirati programski jezik Python 2.x
- Instalirati skup biblioteka SciPy (preporuka: distribucija Anaconda)
- Namjestiti putanju direktorija referentnih i ispitnih slika pomoću konstanti na početku programskog koda:

```
PROJECT_DIR = 'C:/RU/Projekt/'
IMG_DIR_REF = 'RU_Projekt_IMG_REF/'
IMG_DIR      = PROJECT_DIR + 'RU_Projekt_IMG_IN/'
IMG_SAVE_DIR = PROJECT_DIR + 'RU_Projekt_IMG_OUT/'
```

napomena: ukoliko se želi staviti putanja s uobičajenim '/' umjesto '/' onda je potrebno staviti '\\' zbog tipa string tipa

Potrebno je također sadržavati sljedeće direktorije sa slikama (primjeri putanja su prema gornjem primjer konstanti):

- C:\RU\Projekt\ - glavni direktorij u kojem se nalaze ostali direktoriji
- C:\RU\Projekt\RU\_Projekt\_IMG\_REF - direktorij s referentnim slikama lica
- C:\RU\Projekt\RU\_Projekt\_IMG\_IN - direktorij s ispitnim slikama
- C:\RU\Projekt\RU\_Projekt\_IMG\_OUT - direktorij gdje se spremaju rezultati za pojedinu sliku

Nazivi direktorija mogu biti i drugačiji od ovih primjera, bitno je da se napomene pomoću konstanti gdje se koji direktorij nalazi i koje mu je ime. U direktoriju za spremanje slika se automatski nakon obrade slike stvara poddirektorij imena jednakog obrađenoj slici i zatim se u njega spremaju rezultati za tu ispitnu sliku.

## 5. Zaključak

Opisana je metoda detekcije oblika (lica) u slici pomoću generalizirane Houghove transformacije. Transformacija je mapiranje rubnog prostora u akumulacijski prostor tako da instance oblika tvore maksimume u akumulacijskom prostoru. Takvo mapiranje je opisano kao tablica rubnih točaka nazvana R-tablica.

Ova metoda ima sljedeća svojstva:

1. Bilo kakva promjena veličine, rotacije ili referentne točke može se kompenzirati promjenom R-tablice.
2. S obzirom na rubove lica, R-tablica može se jednostavno konstruirati i zahtijeva onoliko operacija koliko je točaka na rubu.
3. Oblici su spremljeni u kanonskom obliku, instance oblika detektirane su poznavajući transformaciju iz kanonskog oblika u instancu. Ako takva transformacija nije poznata, potrebno je isprobati sve moguće transformacije.
4. Ako je oblik zapravo kompozicija više oblika, tada se R-tablica može izgraditi jednostavnim kombiniranjem više R-tablica za svaki oblik.
5. Vrijednosti akumulacijske matrice mogu se interpretirati kao lokalno konzistentna informacija.
6. Generalizirana Houghova transformacija je paralelan algoritam.

Daljni rad je usmjeren prema karakterizaciji efikasnosti algoritma.

Moguća poboljšanja generalizirane Houghove transformacije:

- Kroz nesavršeno grupiranje
  - Pokazano je da je čak i s jako jednostavnim, nesavršenim grupiranjem moguće znatno poboljšati kako brzinu tako i preciznost prepoznavanja objekata.
  - Brzina algoritma povećana je sa  $O(mn^3)$  na  $O(mn^2)$
  - Upotrebom ovih tehnika znatno je smanjena vjerojatnost detektiranja lažno pozitivnih primjera
- Kroz mjeru dosljednosti oblika
  - Zbog nezavisnog odnosa točaka modela u GHT glasanju, međusobno isključive varijacije mogu podržati istu lokalizacijsku hipotezu, što može dovesti do povećanja lažno pozitivnih primjera.
  - Problemu se pristupa analizirajući uzorak glasanja te učenju strukturalnih razlika između uspješnih i neuspješnih lokalizacija.
  - Eksperimenti pokazuju 2% poboljšanje, stoga upućuju da se mogu iskoristiti za poboljšanje lokalizacije.
- Neovisnost o orijentaciji

## 6. Literatura

1. D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, Vol.13, No.2, 1981. , p.111-122  
<http://www.cse.iitd.ernet.in/~pkalra/csl783/GHT-paper.pdf>
2. S. Lazebnik, "Fitting: The Hough transform", University of North Carolina at Chapel Hill, predavanje 2009.  
[http://www.cs.unc.edu/~lazebnik/spring09/lec09\\_hough.pdf](http://www.cs.unc.edu/~lazebnik/spring09/lec09_hough.pdf)
3. S. Savarese, " Generalizing the Hough Transform " ,University of Michigan, predavanje 2013.  
[http://web.eecs.umich.edu/~silvio/teaching/EECS598/lectures/lecture10\\_3.pdf](http://web.eecs.umich.edu/~silvio/teaching/EECS598/lectures/lecture10_3.pdf)
4. Sachin Sudhakar Farfade, Mohammad Saberian, Li-Jia Li. Multi-view Face Detection Using Deep Convolutional Neural Networks.  
<http://arxiv.org/pdf/1502.02766v3.pdf>
5. Improving the generalized Hough transform through imperfect grouping.  
<http://faculty.washington.edu/cfolson/papers/pdf/ivc98.pdf>
6. P. Viola and M. J. Jones. Robust real-time face detection. International Journal of Computer Vision, 2004  
<http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf>
7. A Shape Consistency Measure for Improving the Generalized Hough Transform  
[https://www.researchgate.net/publication/274951666\\_A\\_Shape\\_Consistency\\_Measure\\_for\\_Improving\\_the\\_Generalized\\_Hough\\_Transform](https://www.researchgate.net/publication/274951666_A_Shape_Consistency_Measure_for_Improving_the_Generalized_Hough_Transform)
8. Orientation-invariant generalized Hough transform algorithm based on U-chord curvature  
<http://www.joca.cn/EN/abstract/abstract18539.shtml>