

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3802

# **Automatsko raspoznavanje oblika lica**

Margareta Kušan

Zagreb, lipanj 2014.

Zagreb, 14. ožujka 2014.

## ZAVRŠNI ZADATAK br. 3802

Pristupnik: **Margareta Kušan (0036463538)**  
Studij: Računarstvo  
Modul: Računarska znanost

Zadatak: **Automatsko raspoznavanje oblika lica**

Opis zadatka:

Automatsko raspoznavanje oblika lica zasniva se na tehnikama računalnog vida a može poslužiti kao pomoć pri odabiru npr. naočala ili stila frizure u sustavima za automatsko preporučivanje. Na Zavodu za telekomunikacije je dostupan softver za detekciju u lica i značajki lica (npr. vrh nosa, kutevi usana, zjenica...) u digitaliziranoj slici lica, te se ovaj softver može koristiti kao osnova. Raspoznavanje oblika može se izvesti u dva koraka: segmentiranje područja lica od ostatka slike i samo raspoznavanje zasnovano na prevlačenju predložaka osnovnih oblika lica preko prethodno segmentirane slike.

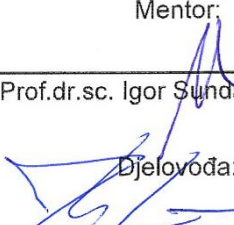
Vaša je zadaća proučiti sustav za detekciju značajki lica dostupan na Zavodu za telekomunikacije, te predložiti, implementirati i testirati metodu za automatsko raspoznavanje oblika lica.

Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Zadatak uručen pristupniku: 14. ožujka 2014.

Rok za predaju rada: 13. lipnja 2014.


Mentor:

  
\_\_\_\_\_  
Prof.dr.sc. Igor Sunday Pandžić

Djelovoda:

  
\_\_\_\_\_  
Doc.dr.sc. Tomislav Hrkać

Predsjednik odbora za  
završni rad modula:

  
\_\_\_\_\_  
Prof.dr.sc. Siniša Srblić



# SADRŽAJ

UVOD.....	1
1. Alati i tehnologije.....	2
1.1. Visage SDK.....	2
1.1.1. FaceDetector .....	2
1.2. OpenCV .....	3
2. Oblik lica .....	4
3. Segmentacija .....	6
4. Detekcija oblika lica .....	10
4.1. Predlošci .....	10
4.2. Funkcija preklapanja .....	12
5. Rezultati.....	13
5.1. Problemi i ograničenja.....	13
5.2. Baza.....	16
5.3. Matrica zabune.....	17
ZAKLJUČAK.....	20
LITERATURA.....	21
Naslov, sažetak i ključne riječi (HRV i ENG) .....	22

# UVOD

Ljudi oduvijek žele pronaći savršenstvo u okvirima ljudske ljepote. Jedan od tih okvira čini ljudsko lice. Postoje razne klasifikacije oblika lica koje pomažu stilistima, frizerima, kozmetičarima i optičarima da, uz korištenje naučenih algoritama, odaberu najbolje značajke za svako ljudsko lice.

Automatsko raspoznavanje oblika lica iz slike može se izvesti u 3 koraka. Prvi korak implementira detekciju značajki lica u slici (oči, nos itd.) koja se zasniva na tehnikama računalnog vida. Softver korišten za realizaciju detekcije značajki lica je Face Detect tehnologija tvrtke Visage Technologies. U drugom koraku, potrebno je segmentirati ljudsko lice od ostatka slike. Segmentacija je tehnika koja izdvaja tražene objekte od ostatka slike. Postoje razni načini segmentacije lica. U ovom radu odabrana je klasifikacija piksela na temelju svojstava boja RGB modela. Treći korak sadržava detekciju oblika lica na temelju zadanih predložaka. Broj predložaka i oblika lica varira ovisno o odabiru klasifikacije ljudskih oblika lica. Piksela po piksel analiza predložaka i segmentirane slike u sklopu funkcije preklapanja daje detektirani oblik lica.

Osim opisanog načina automatskog raspoznavanja oblika lica, postoji način koji implementira mjerenje udaljenosti i računanje kuteva između detektiranih značajki lica, međutim taj način nije razmatran u ovom radu.

# 1. Alati i tehnologije

Ovaj rad koristi Visage|SDK softver za detekciju karakterističnih točaka lica (*facial feature points*). Unutar Visage|SDK softvera te u ovom radu, koristi se OpenCV biblioteka. Kompletan rad se temelji na C++-u, uz iznimku obrade dobivenih datoteka ostvarenu u Pythonu. Za razvoj je korišten Microsoft Visual Studio 2010.

## 1.1. Visage|SDK

Visage|SDK je razvojno okruženje koje sadrži tehnologije iz područja računalnog vida i animacije virtualnih likova razvijan od strane tvrtke Visage Technologies<sup>1</sup>. Tehnologije u odvojenim paketima pružaju praćenje i detekciju ljudskog lica i njegovih značajki (usta, oči, obrve, nos itd.) u slici i videu. Neki od paketa: [1]

- Face Track - praćenje pokreta lica i glave u video zapisu
- Face Detect - otkrivanje izraza lica na slikama
- Head Track - praćenje pokreta glave u video zapisu
- Visual TTS - optički pretvarač teksta u govor
- Lip Sync - automatska sinkronizacija usana u stvarnom vremenu

### 1.1.1. FaceDetector

O ovom radu koristimo paket Face Detect koji detektira značajke jednog ili više ljudskih lica u slikama. Podržava većinu standardnih formata (JPEG, GIF itd.). Rezultat detekcije je set MPEG-4 značajki lica - karakteristične točke usana, nosa, očiju, obrva, brade, granice kose i lica te obraza. Minimalna veličina lica koje se detektira je 50 piksela u promjeru. Slike mogu biti u boji ili *grayscale*<sup>2</sup>. [5] U ovom se radu neće razmatrati *grayscale* slike zbog ograničenosti izvedbe segmentacije o čemu će biti govora u narednim poglavljima.

---

<sup>1</sup> [www.visagetechologies.com](http://www.visagetechologies.com)

<sup>2</sup> U sivim tonovima

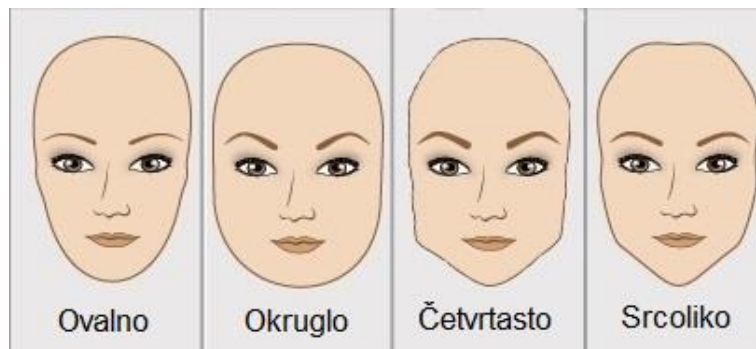
## 1.2. OpenCV

OpenCV (*Open Source Computer Vision*) je *open source* biblioteka za strojno učenje i računalni vid, dostupna za akademsku i komercijalnu upotrebu. OpenCV je dizajniran za računalnu učinkovitost (*computational efficiency*) s naglaskom na *real-time* aplikacije. Sadrži C++, C, Python, Java i MATLAB sučelja te podržava Windows, Linux, Android i Mac OS operacijske sustave. [3]

Ovaj rad koristi C++ sučelje OpenCV biblioteke, a implementacija je napravljena na Windows platformi.

## 2. Oblik lica

Oblik lica važan je faktor kod odabira kozmetike, frizure, obrva ili okvira naočala. Postoji više klasifikacija oblika lica. Osnovna podjela lica svodi se na 4 oblika: ovalno, četvrtasto, okruglo i srcoliko.



Slika 1. 4 osnovna oblika lica

Ovalni oblik lica karakterizira omjer duljine prema širini koji iznosi oko 1.5, te oblik jajeta. Ovo je najčešći oblik lica te se smatra najzahvalnijim. Kod odabira frizure i šminke, teži se vizualnom prividu ovalnog lica.

Kod četvrtastog oblika lica omjer duljine i širine je između 1 i 1.5 i to kod čela, jagodica i čeljusti. Karakteriziraju ga oštre, četvrtaste linije, uključujući oštru čeljust.

Okruglo lice ima omjer duljine i širine najbliži 1, te mu je širina čela, jagodica i čeljusti jednaka, ali je čeljust blago zaobljena, a lice oblije i mekše. Razlikuje se od četvrtastog u kutovima. Četvrtasto ima jake, uglate značajke, a okruglo je mekše.

Srcoliko lice karakterizira široko čelo, uska brada te izražene jagodice. Poznat je još kao obrnuti trokut.

Oblik lica se može mijenjati s godinama. Kod djece nema smisla mjeriti oblik lica zato što imaju više masnog tkiva i većinom se nalaze u okvirima okruglog oblika. Starenjem gubimo masno tkivo na licu te nam se koža počinje opuštati i tada se više krećemo prema četvrtastom obliku.

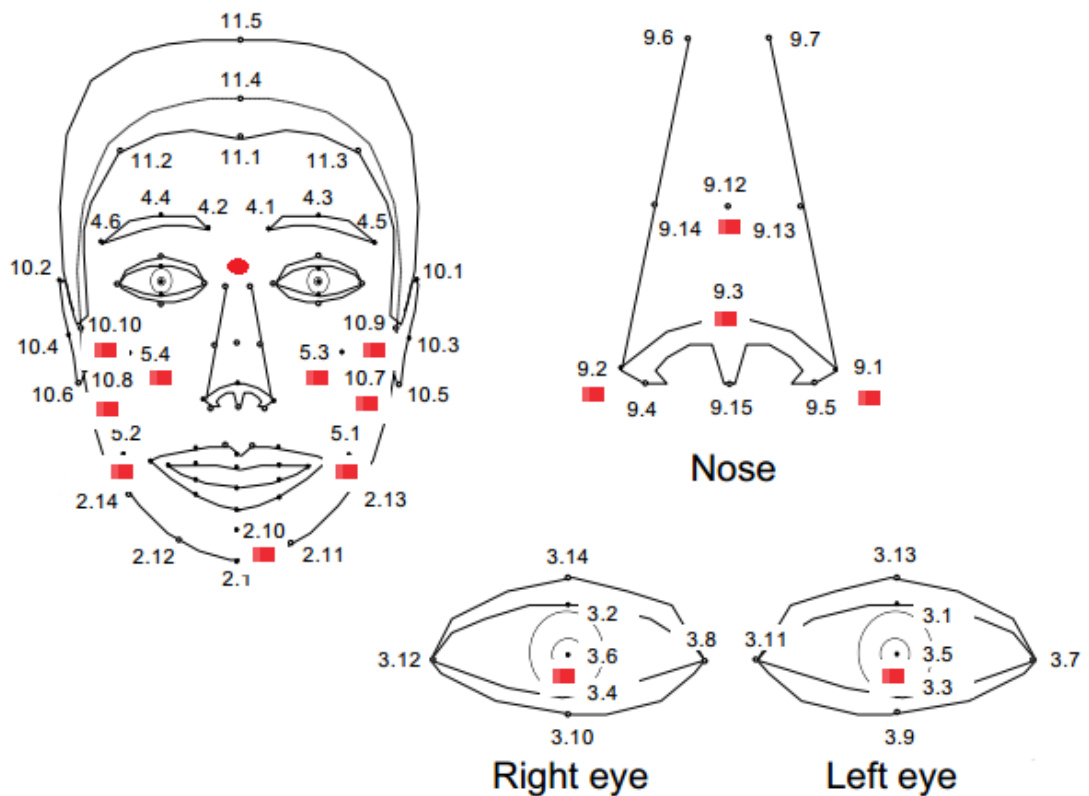


Iako je ova osnovna podjela najčešća, mnoga lica se ne mogu svrstati u ove okvire. Zato postoje podjele na više oblika. Najčešći od dodatnih oblika su duguljasto, dijamantno, pravokutno te trokutasto.

U ovom radu ćemo se koncentrirati na 4 osnovna oblika lica: ovalno, četvrtasto, okruglo i srcoliko.

### 3. Segmentacija

Podloga detekcije oblika lica je segmentacija lica. Odabrana je metoda poklapanja RGB<sup>3</sup> vrijednosti piksela s bojom kože. Boja kože na slici varira o dobu godine, rasi, osvjetljenju itd. Zbog toga se na svakoj slici računaju RGB vrijednosti karakterističnih točaka dobivenih pomoću FaceDetectora iz Visage|SDK.



Slika 2. Karakteristične točke (*Facial feature points*). [4]

Na gornjoj slici crveno označene točke su karakteristične točke korištene tijekom segmentacije. Karakteristične točke nam daju koordinate točke na slici koja je skalirana tako da je u donjem desnom kutu koordinata (1,1), a u gornjem lijevom (0,0). Kada  $x$  koordinatu pomnožimo sa širinom, a  $y$  koordinatu točke sa dužinom slike (formula 3.1.), dobijemo piksel u toj koordinati.

<sup>3</sup> Red, Green, Blue

$$\begin{aligned}x &= \text{image.width} * x\_coordinate \\y &= \text{image.height} * y\_coordinate\end{aligned}\quad (3.1.)$$

Nadalje radimo s pikselima. Točke koje su označene crvenom bojom na slici 2 su korištene u segmentaciji lica. Točke 10.10, 10.9, 10.8, 10.7, 9.12, 9.3, 9.2, 9.1, 5.4, 5.3, 5.2, 5.1 i 2.10 su direktno korištene za dobivanje RGB vrijednosti piksela. Točke 3.5 i 3.6 su korištene za dobivanje vrijednosti točke između očiju.

Klasična RGB klasifikacija boja<sup>4</sup> nije prikladna za određivanje intervala boje kože zato što osim boje prikazuje osvjetljenje koje bitno varira na površini ljudskog lica. Osvjetljenje možemo maknuti koristeći formulu 3.2. te na taj način dobiti boje u čistom obliku.<sup>5</sup>

Kad imamo piksel i njegove vrijednosti RGB, prema formuli 3.2. mičemo osvjetljenje iz boja te nam ostaju čiste boje.

$$\begin{aligned}R &= R / (R + G + B) \\G &= G / (R + G + B) \\B &= 1 - R - G\end{aligned}\quad (3.2.)$$

Sada nam se RGB vrijednosti kreću u intervalu [0,1]. Gledajući svaku od točaka, proširujemo interval za RGB vrijednosti unutar kojih se nalazi i boja kože.

Primjerice, za neku sliku  $I$ , R vrijednost mora biti u intervalu [0.38, 0.45], G u intervalu [0.15, 0.20] te B u intervalu [0.10, 0.15].

Znamo poziciju lica u slici što nam omogućuje detekcija karakterističnih točaka. Koristeći točku 3.5 (sredina lijeve zjenice) i 3.6 (sredina desne zjenice) možemo odrediti udaljenost između očiju.

$$\text{eyedist} = \text{leyex} - \text{reyex} \quad (3.3.)$$

Formula 3.3. prikazuje udaljenost između očiju  $\text{eyedist}$ , gdje je  $\text{leyex}$  x koordinata lijevog oka, a  $\text{reyex}$  x koordinata desnog oka. Koristeći tu udaljenost možemo stvoriti *bounding box*<sup>6</sup> oko lica za koji smo sigurni da sadrži lice. Prvo prema formuli 3.4. izračunamo koordinate točke između očiju.

<sup>4</sup> Klasična RGB klasifikacija boja sadrži brojeve u intervalu [0,255] za Red, Green i Blue komponentu.

<sup>5</sup> Tzv. *pure colors*

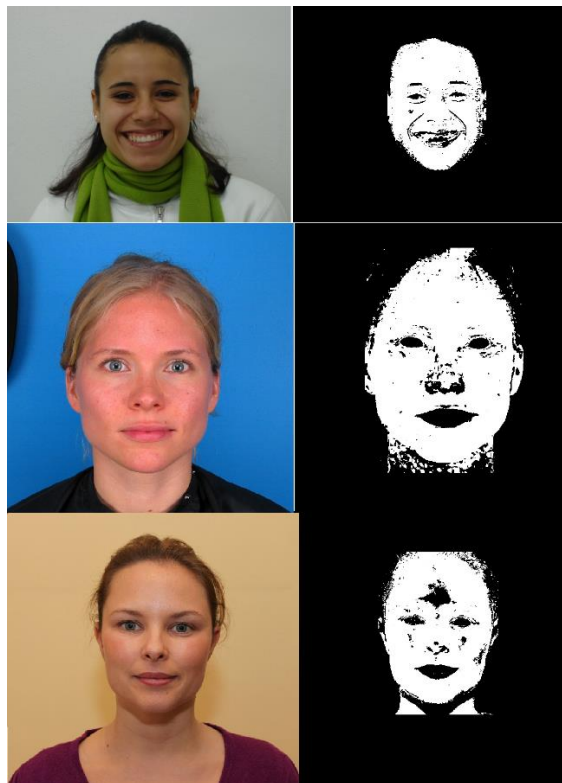
<sup>6</sup> Kvadrat na slici unutar kojeg se nalazi traženi objekt

$$\begin{aligned} \text{between\_eyes\_x} &= \frac{\text{leyex} + \text{reyex}}{2} \\ \text{between\_eyes\_y} &= \frac{\text{leyey} + \text{reyey}}{2} \end{aligned} \quad (3.4.)$$

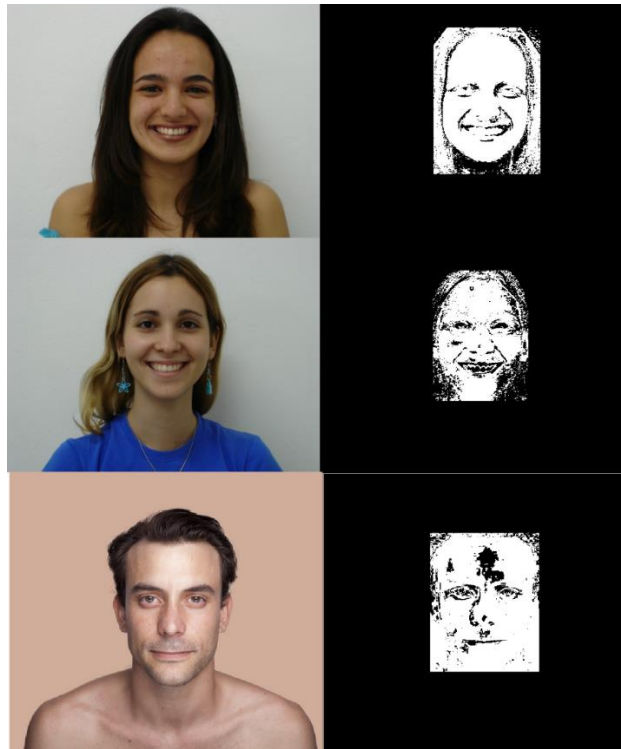
Gledajući velik skup slika, zaključeno je da se lice nalazi u intervalu  $1.2 * eyedist$  lijevo i desno od koordinate  $\text{between\_eyes\_x}$ , te  $1.4 * eyedist$  gore i  $1.90 * eyedist$  dolje od koordinate  $\text{between\_eyes\_y}$  (gledajući da pozitivna os koordinate y gleda prema dnu ekrana). Ove vrijednosti čine *bounding box* u kojem ćemo tražiti lice.

Kad smo izračunali sve vrijednosti, prolazimo kroz svaki piksel slike te pratimo sljedeći algoritam:

1. Ako se piksel ne nalazi unutar *bounding box*-a, postavi njegove RGB vrijednosti na [0,0,0] (crna boja).
2. Ako se piksel nalazi unutar *bounding box*-a:
  - a. Ako se RGB vrijednosti piksela nalaze unutar intervala boje kože, postavi njegove RGB vrijednosti na [255,255,255] (bijela boja).
  - b. Ako se RGB vrijednosti piksela ne nalaze unutar intervala boje kože, postavi njegove RGB vrijednosti na [0,0,0].



Slika 3. Dobri rezultati segmentacije



Slika 4. Loši rezultati segmentacije

Na slici 3. možemo vidjeti da, ako zanemarimo činjenicu da ne možemo segmentirati *grayscale* slike, ovaj pristup može biti podloga za detekciju oblika lica. Iz slike 4. vidimo da postoji problem ako je pozadina sličnih boja kao boja lica, te ponekad segmentacija zahvaća kosu i vrat, što nije poželjno.

## 4. Detekcija oblika lica

Ideja realizacije detekcije oblika lica se temelji na ideji preklapanja predložaka i segmentirane slike. Na taj način možemo dobiti maksimalan broj preklapanja u pikselima za svaki od 4 osnovna predloška.

### 4.1. Predlošci

Rezultati samog rada su jako ovisni o odabiru predloška. Sam predložak mora biti primjer lica koje se u potpunosti uklapa u određeni oblik lica.



Slika 5. Primjer predložaka

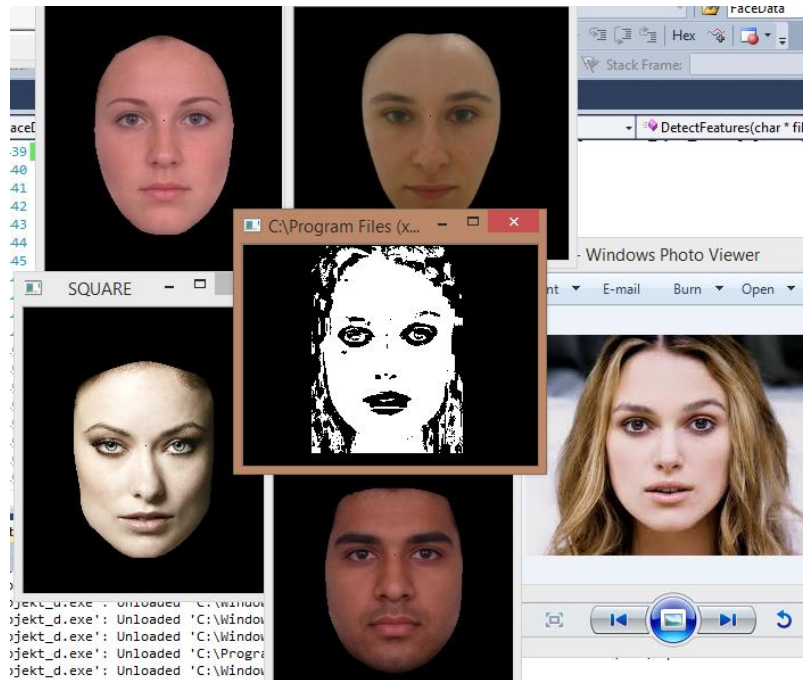
Predlošci ne trebaju biti iste veličine. Pamtimo udaljenost očiju *eyedist* (u pikselima) za svaki od 4 predloška. Prilikom obrade slike *I* računamo njezinu udaljenost između očiju *eyedist*. Veličina svakog predloška se prilagođava veličini slike na način da računamo omjer udaljenosti *eyedist* slike *I* i predloška *T*.

$$percent = \frac{eyedist\_image}{eyedist\_template} \quad (4.1.)$$

Formula 4.1. prikazuje računanje omjera veličine slike i predloška gdje je *percent* omjer, *eyedist\_image* udaljenost očiju slike *I* te *eyedist\_template* udaljenost očiju predloška *T*.

Zatim posebno pomnožimo širinu i dužinu slike predloška omjerom *percent* te dobijemo sliku predloška  $T$  u veličini koja odgovara veličini slike  $I$ .

Na taj način smo pripremili predložak za funkciju preklapanja.



Slika 6. Slika i predloži nakon prilagodbe veličine

## 4.2. Funkcija preklapanja

Cilj funkcije preklapanja<sup>7</sup> je dobiti broj piksela kojima je RGB vrijednost različita od [0,0,0] na slici i na predlošku. Jedan način dobivanja tog broja bio bi provlačenje predloška po cijeloj slici i računanje maksimalnog preklapanja. Taj algoritam ima veliku vremensku složenost te kao takav nije adekvatan za detekciju. Drugi način izravno preklapa predložak na sliku prema točki između očiju. Ovaj način se pokazao boljim te je funkcija preklapanja implementirana prema toj ideji.

Prilikom pripreme predloška izračunali smo koordinate točke između očiju. Širina piksela čije RGB vrijednosti tražimo određena je *bounding box*-om, koji smo objasnili u poglavlju 3. Ostale piksele ne treba gledati jer su segmentacijom postavljeni na [0,0,0]. Gledajući piksel po piksel istovremeno na slici i predlošku, zbrajamo piksele koji su različiti od [0,0,0], te zbrajamo ukupan broj piksela koji sadrže boju različitu od crne na svakom predlošku. Ukupan broj piksela predloška varira zbog promjene veličine predloška. Ponavljamo za svaki predložak.

Rezultat je broj podudaranja u pikselima za svaki predložak. Dijeleći broj preklapanja sa ukupnim brojem obojanih piksela na predlošku, dobijemo postotak preklapanja slike i predloška koji se sprema u datoteku.

---

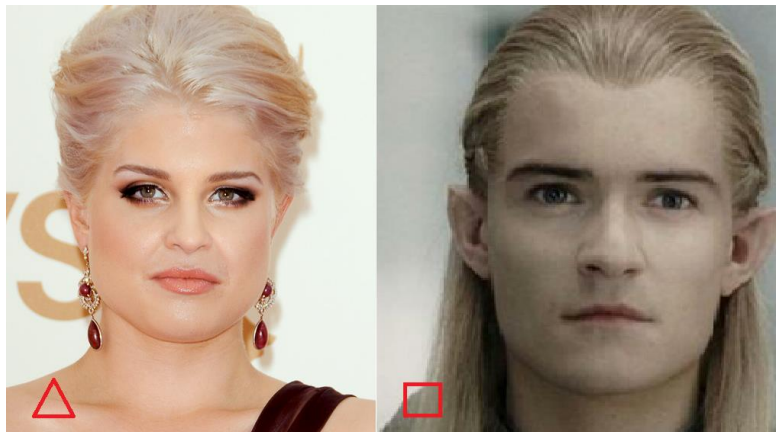
<sup>7</sup> Eng. *template matching*



## 5. Rezultati

### 5.1. Problemi i ograničenja

Problematika ovog zadatka leži u više činjenica. Prva od njih je činjenica da ne postoje široko prihvaćeni okviri klasifikacije oblika ljudskog lica. Postoje 4 osnovna oblika lica, koja se koriste u ovom radu, ali se dosta ljudskih lica ne može svrstati u te granice. Ljudska lica, iako sadrže jednake značajke, u smislu oblika mogu poprilično varirati. Zato postoje dodatne klasifikacije oblika lica. Ipak, najveći skup ljudskih lica je koncentriran u 4 osnovna oblika, što je ujedno razlog odabira te klasifikacije za ovaj rad.



Slika 7. Usporedba oblika lica: a) četvrtasto/okruglo, b) četvrtasto

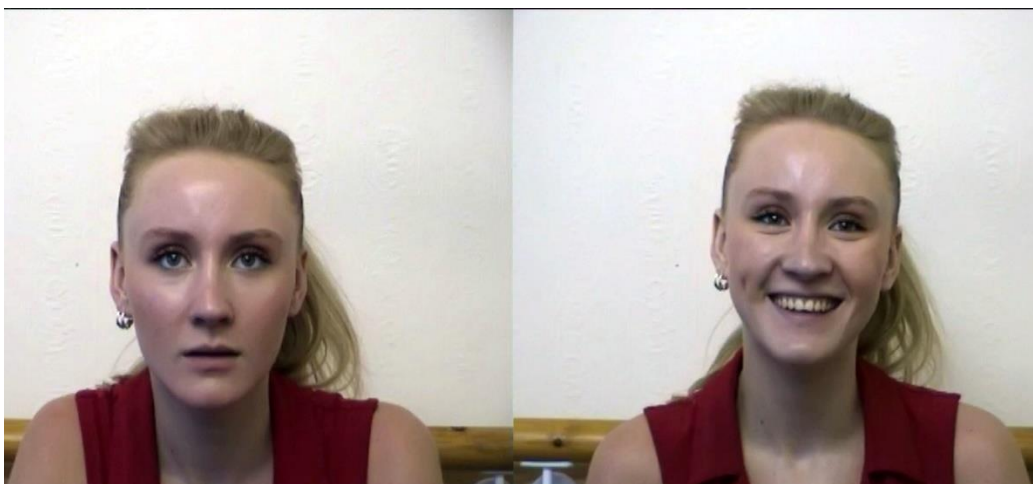
Na slici 7. vidimo upravo jedan takav primjer. Dok se lice na 7.b. savršeno očituje kao četvrtasti oblik lica, kod lica na 7.a. postoje karakteristike četvrtastog i okruglog. Problem je u tome što kod osobe na slici 7.a. čelo savršeno pristaje u okrugli oblik lica, a čeljust ima dosta izražene crte koje više odgovaraju četvrtastom obliku lica. Takvo lice se u dodatnim klasifikacijama naziva trokutasto.

Osim činjenice da se ne mogu sva lica svrstati u osnovnu klasifikaciju, postoje određena ograničenja same detekcije iz jedne statične slike. Osoba se treba slikati u savršeno frontalnoj pozi, bez odmaka u rotaciji glave u bilo kojoj osi.

Implementacija rotacije glave bi se mogla dodatno ostvariti, ali to nije tema ovog

rada. Ovaj problem se može riješiti na način da se zamoli korisnika da se slika na takav način.

Dodatni problem detekcije oblika lica su izrazi lica. Lice mora biti neutralnog izraza lica, zato što se konture oblika vidno mjenjaju kod bilo kakvog izraza, otvaranja ustiju, podizanja čela itd.



Slika 8. Izrazi lica: a) ovalno lice, b) izražajnije linije zbog promjene izraza lica

Na slici 8.a. možemo vidjeti ovalno lice u neutralnom položaju. Nema rotacije glave, linije lica su pravilne i uklapaju se u ovalno lice. Na slici 8.b. osoba je promijenila izraz lica te rotirala glavu u z i x osi. Ovakav izraz lica te pozicija glave mijenjaju dojam samog oblika lica. Čeljust poprima jače definirane linije te je promijenjen omjer duljine i širine lica. Ovakve promjene rezultiraju povećanjem poklapanja u broju piksela za četvrtasti oblik lica.

Uz opisana ograničenja, postoje ograničenja proizašla iz implementacije algoritma segmentacije. Prvo i spomenuto u poglavlju 3., slike ne smiju biti *grayscale*. Drugo, za najbolje rezultate, površina lica bi trebala biti što više otkrivena, što znači da se trebaju vidjeti vanjske crte lica. To se odnosi na naočale, kosu i bradu. Gledajući rezultate, uočeno je da brada i naočale imaju manji utjecaj na detekciju u usporedbi s kosom. Kao što je već prikazano na slici 4., kosa i pozadina mogu dosta utjecati na točnost segmentacije, a samim time i na konačne rezultate.

Kad se u funkciji preklapanja preklope slike prema točkama između očiju, postoji određena pretpostavljena širina lica određena predlošcima. Zato možemo do neke mjere ignorirati kosu koja se nalazi izvan okvira lica. Kosa koja se nalazi unutar okvira lica puno više utječe na točnost rezultata. Taj problem možemo riješiti i tako da zamolimo osobe da maknu kosu s lica te na taj način adekvatno otkriju sve rubove lica.

Treći problem segmentacije je činjenica da *bounding box* koji se dobiva pomoću parametra *eyedist* nije savršeno poklopljen s dužinom i širinom lica, što znači da kod nekih slika, osim lica, u *bounding box* ulazi i vrat.



Slika 10. Segmentacija koja uključuje vrat

Na slici 10. je primjer osobe kojoj je duljina glave ispod prosječne duljine glave, čime se u *bounding box*-u stvara višak prostora, a što rezultira ulaskom vrata u segmentaciju. S obzirom na činjenicu da je dužina glave ispod prosjeka, funkcija preklapanja gledajući predložak može lažno detektirati vrat kao bradu, točnije, 'produljiti' lice te krivo detektirati oblik lica.

Pretpostavljeni oblik lica koji zadajemo kao točan u dosta slučajeva ovisi o subjektivnom dojmu, što najviše od svih opisanih problema i ograničenja utječe na matricu zabune<sup>8</sup>.

---

<sup>8</sup> Eng. *matrix of confusion*

## 5.2. Baza

Baza testnih slika se sastoji od 418 slika ljudi različitog spola, dobi i rase. Većina slika u bazi nema neutralan izraz lica, te dosta slika ima blagu rotaciju glave. Neka lica se pojavljuju u više slika, uz različit izraz lica ili promijenjen položaj glave. Takve slike će poslužiti za detaljniju analizu osjetljivosti detekcije oblika lica.

Mapa iz koje FaceDetector radi *batch* detekciju<sup>9</sup> sadržava datoteku u kojoj se nalazi popis svih slika koje treba obraditi, slike i datoteku u kojoj se za svaku sliku nalazi pretpostavljeni oblik lica. U istoj mapi se nalazi i Python skripta 'test.py' koja za svaki od pretpostavljenih oblika vraća broj detektiranih oblika u svakom od 4 osnovna oblika lica. Pokretanjem i uspješnim izvršenjem aplikacije za svaku se sliku u istu mapu sprema datoteka u kojoj se nalazi maksimalan broj piksela za svaki od 4 oblika lica. Osim datoteke, u istu mapu se sprema segmentirana slika s prefiksom '\_D\_'.

Osim *batch* detekcije, moguće je provoditi detekciju oblika lica na svakoj slici posebno, u tom slučaju nije potrebno uzimati sliku iz pretpostavljane mape već iz bilo koje lokacije na računalu. Datoteka se automatski sprema na istu lokaciju na kojoj se nalazi slika, ali na ovaj način nema dodatnog spremanja segmentirane slike.

---

<sup>9</sup> Opcija FaceDetectora za obradu više slika iz jedne datoteke.

### 5.3. Matrica zabune

Matrica zabune prikazuje distribuciju pogrešaka. U vodoravnom prikazu (retci) se nalaze detektirani oblici lica. U okomitom prikazu (stupci) se nalaze oblici lica na ulazu. Vrijednosti na dijagonali označavaju točnost rezultata, odnosno postotak točno detektiranih lica za pojedini oblik.

Tablica 1.: Matrica zabune

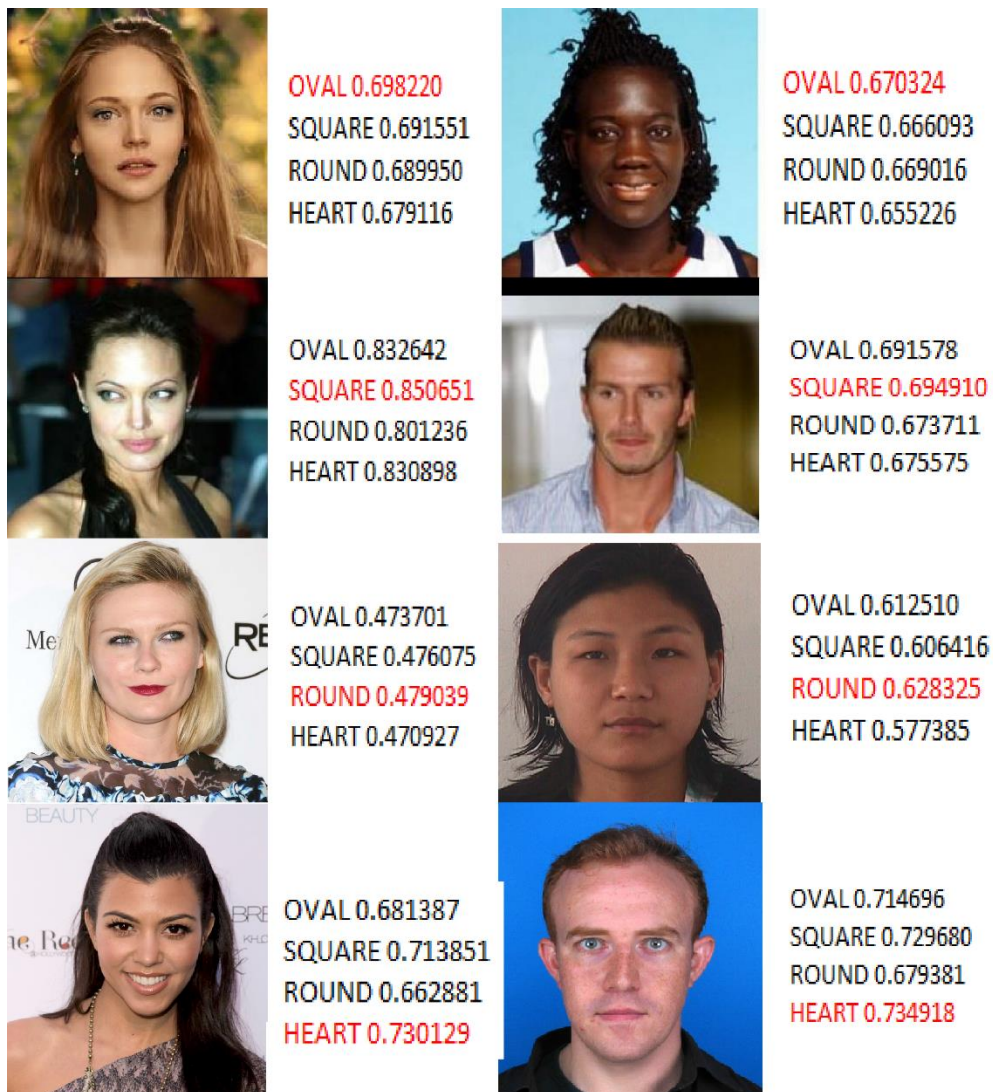
OBLIK LICA	OVALNO	ČETVRTASTO	OKRUGLO	SRCOLIKO
OVALNO	53.2 %	21.3 %	16.5 %	9 %
ČETVRTASTO	11.3 %	61.8 %	13.9 %	13 %
OKRUGLO	5.8 %	21.8 %	70.1 %	2.3 %
SRCOLIKO	17.9%	21.4 %	14.3 %	46.4 %

Ovalno i četvrtasto lice ne dijele mnogo zajedničkih karakteristika. Segmentacija je razlog velikog broja ovalnih lica krivo detektiranih kao četvrtasta. Segmentacija u velikom broju slučajeva obuhvaća vrat. Vrat može dati krivu percepciju definicije čeljusti. Ovakav primjer prikazan je na slici 10. na kojoj se nalazi osoba čiji oblik lica je detektiran kao četvrtast zbog udjela vrata u segmentiranoj slici.

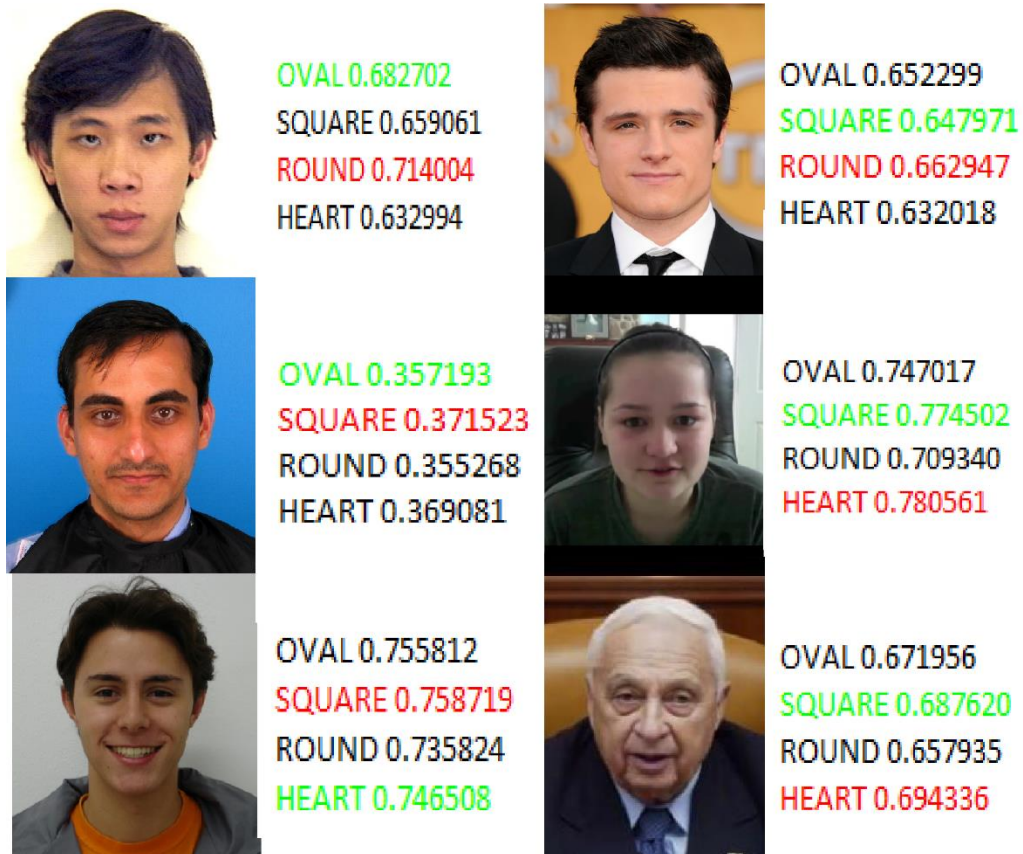
Razlika između okruglog i četvrtastog lica je u obliku čela koji kod četvrtastog lica poprima pravokutan izgled te u definiciji čeljusti koja kod četvrtastog lica poprima izraženije crte. Postoje mnoga lica koja čine kombinaciju ova dva oblika koja su u konačnici dosta slična. Kada kosa prekrije pravokutne linije u predjelu čela, dobije se privid mekših, okruglih linija, također kod izražajne ekspresije osmijeha linije čeljusti dobivaju izraženije crte. Stoga najveći dio krivo detektiranih četvrtastih oblika lica čini okrugli oblik te najveći dio krivo detektiranih okruglih lica čini četvrtasti oblik.

Razlika između četvrtastog i srcolikog lica je u izgledu čeljusti. Kod četvrtastog oblika lica brada je uvučenija u same linije čeljusti na način da se dobije privid ravne linije, a kod srcolikog oblika lica brada je izdužena i šiljasta. Manje razlike u samom obliku lica uz nepreciznu segmentaciju rezultiraju najvećim brojem srcolikih lica krivo detektiranih kao četvrtasta te većim brojem četvrtastih lica detektiranih kao srcolika.

Na kraju, ovalni i okrugli oblik lica imaju zajedničkih karakteristika, a to su mekše linije uzduž cijelog lica. Razlika je u omjeru duljine i širine. Stoga je broj ovalnih lica detektiranih kao okruglo lice dosta velik.



Slika 11. Primjeri dobre detekcije



Slika 12. Primjeri netočne detekcije: zelenom bojom je određen pravi oblik lica, a crvenom detektirani

# ZAKLJUČAK

Trenutna implementacija automatskog raspoznavanja oblika lica uz određena poboljšanja mogla bi rezultirati odličnim rezultatima.

U trenutno implementiranom načinu rada jedno od poboljšanja uključuje egzaktniju segmentaciju lica. To bi se moglo ostvariti eliminacijom suvišnih piksela koji određuju kosu, vrat, uši itd. Eliminacija se može bazirati na korištenju karakterističnih točaka lica.

Segmentacija lica se može izvesti drugačijim tehnikama koje se ne baziraju na klasifikaciji piksela temeljem svojstava boja RGB modela. Ovom modifikacijom u detekciju bismo uključili grayscale slike, te bi se smanjio utjecaj pozadinskih boja na slici.

Osim segmentacije, poboljšanje je moguće na području odabira same klasifikacije oblika lica koje se može bazirati na većem broju oblika lica. To bi povlačilo veći odabir predložaka te bi se na taj način veći broj ljudskih lica moglo svrstati u određeni oblik lica, primjerice lica koja u trenutnoj klasifikaciji predstavljaju kombinaciju nekih od 4 osnovna oblika lica.

Naknadno, kako bismo proširili skup slika, mogli bismo dodati eliminaciju rotacije glave pomoću matrica transformacije pogleda.

Na kraju, osim opisanog načina implementacije automatskog raspoznavanja oblika lica koji koristi funkciju preklapanja na razini piksela, mogli bismo koristiti način mjerenja udaljenosti i omjera dužina i širina između karakterističnih značajki lica.



# LITERATURA

- [1] Visage Technologies, Visage|SDK,  
<http://www.visagetechologies.com/products/visagesdk>
- [2] Visage Technologies, *VisageSDK 7.1 documentation*
- [3] OpenCV, <http://opencv.org>
- [4] Visage Technologies, *MPEG-4 Face and Body Animation (MPEG-4 FBA)*,  
<http://www.visagetechologies.com/uploads/2012/08/MPEG-4FBAOverview.pdf>
- [5] Visage Technologies, Visage|SDK,  
<http://www.visagetechologies.com/products/visagesdk/facedetect/>

## **Naslov, sažetak i ključne riječi (HRV i ENG)**

**NASLOV:** Automatsko raspoznavanje oblika lica

### **SAŽETAK:**

Automatsko raspoznavanje oblika lica iz slike u ovom radu temelji se na tri koraka: detekcija značajki lica, segmentacija lica iz slike i detekcija oblika lica pomoću predložaka i funkcije preklapanja. Za detekciju značajki lica korišten je FaceDetector iz Visage|SDK razvojnog okruženja. Segmentacija lica izvedena je klasifikacijom piksela na temelju svojstava boja RGB modela. Detekcija oblika lica izvedena je računajući preklapanja predloška i segmentirane slike u funkciji preklapanja. Za klasifikaciju oblika lica korištena su 4 osnovna oblika: ovalno, četvrtasto, okruglo i sroliko. Opisani su problemi i ograničenja ove implementacije i ovakvog pristupa, te su predložena poboljšanja detekcije oblika lica.

### **KLJUČNE RIJEČI:**

Visage Technologies, OpenCV, detekcija značajki lica, detekcija lica, FaceDetect, oblik lica, segmentacija kože, segmentacija lica, matrica zabune, Visage|SDK, funkcija preklapanja s predloškom

**TITLE:** Automatic face shape recognition

**SUMMARY:**

Implementation of automatic face recognition from still image described in this paper is based on three steps: facial features detection, face segmentation and face shape detection. Facial features are detected using Visage|SDK FaceDetect package developed by Visage Technologies. Face segmentation uses color pixel classification. Face shape recognition is implemented using template matching method based on 4 basic face shapes: oval, square, round and heart. Problems and limitations within used approach are analysed and described in this paper alongside with the results. Possible improvements are given and briefly described.

**KEY WORDS:**

Visage Technologies, OpenCV, facial features detection, face detection, FaceDetect, face shape, segmentation, face segmentation, matrix of confusion, Visage|SDK, template matching