



Matematički fakultet, Univerzitet u Beogradu

ALGORITMI PROCESIRANJA SLIKA ZA DETEKCIJU LICA

Master rad

Marko Stanković

Beograd, 2014

Mentor: prof. dr Milan Tuba
Matematički fakultet, Univerzitet u Beogradu

Članovi komisije: prof. dr Miodrag Živković
Matematički fakultet, Univerzitet u Beogradu

prof. dr Vladimir Filipović
Matematički fakultet, Univerzitet u Beogradu

Datum odbrane: _____

Apstrakt

Detekcija lica se primenjuje u brojnim oblastima kao što su prepoznavanje lica, video nadzor, prepoznavanje izraza lica, interfejs između čoveka i računara, automatizacija formiranja baza podataka lica i mnoge druge, uz potencijal za proširenje omogućen dostupnošću jeftinijeg i jačeg hardvera. Ovaj rad predlaže algoritam koji za osnovu uzima prepoznavanje kože u YCbCr prostoru boja, i nadograđuje ga korišćenjem procene gustine ivica unutar detektovanih regiona i poređenjem njihovih spoljnih kontura sa predefinisanim šablonom pomoću Huovih momenata, ne bi li se uspešno izdvojila lica među potencijalnim kandidatima. Predloženi algoritam ne uzima u obzir poziciju karakterističnih crta lica, već samo njihovo prisustvo, čineći ga primenljivim bez obzira na pozu lica. Testiranje je pokazalo da je algoritam prilično pouzdan u samostalnoj primeni, kao i da je dobra polazna tačka za dalji razvoj korišćenjem naprednijih tehnika, i prilagođavanje za upotrebu u specifičnim oblastima gde je detekcija lica neophodna.

Ovom prilikom želim da se zahvalim svom mentoru, prof. dr Milanu Tubi, na savetima, dobronamernoj kritici i pomoći prilikom izbora literature, kao i članovima komisije na korisnim sugestijama. Takođe želim da se zahvalim porodici i prijateljima na moralnoj podršci koju su mi pružili tokom izrade rada.

Marko Stanković

Beograd, 2014

Sadržaj

1	Uvod	7
2	Digitalne slike	9
2.1	Formiranje digitalnih slika	9
2.2	Primena digitalnih slika.....	12
2.3	Tipovi digitalnih slika	13
3	Obrada digitalnih slika	14
4	Modeli i prostori boja	16
4.1	RGB	16
4.2	CMYK.....	17
4.3	CIELAB	18
4.4	HSV.....	19
5	Prepoznavanje objekata.....	21
6	Detekcija lica	25
6.1	Modeliranje boje kože	26
6.2	YCbCr prostor boja.....	27
6.3	Algoritmi za detekciju kože u YCbCr prostoru	28
7	Predloženi algoritam za detekciju lica	31
7.1	Detekcija kože.....	31
7.2	Morfološke operacije.....	32
7.3	Izdvajanje regiona kože	33
7.4	Kenijev detektor ivica	33
7.5	Određivanje gustine ivica	36
7.6	Huovi invarijantni momenti.....	39
7.7	Poređenje kontura.....	40
8	Programska implementacija	42

9	Testiranje i rezultati	45
10	Zaključak.....	52
11	Lista slika	53
12	Lista referenci.....	54

1 Uvod

Detekcija lica je prvi i nezobilazni korak u brojnim oblastima, kao što su prepoznavanje lica, video nadzor, prepoznavanje izraza lica, interfejs između čoveka i računara, automatizacija formiranja baza podataka lica, i mnogim drugim. Kao takva, detekcija lica je i tema brojnih naučnih radova.

Detekcija lica uglavnom podrazumeva lokalizaciju lica na slici i određivanje njegovih granica. Primene su brojne i to u najrazličitijim oblastima. Automatizacija detekcije lica na velikom broju slika se koristi za formiranje baza podataka lica, koje svoju primenu najčešće nalaze u brojnim istraživanjima. Detekcija lica je uglavnom prvi i nezaobilazni korak u procesu prepoznavanja lica, koje se često svodi na upoređivanje detektovanog lica sa postojećim licima u bazi, odnosno određivanje njihove međusobne sličnosti. Detekcija lica na video materijalu i njihovo čuvanje zajedno sa vremenom pojavljivanja se koristi u pojedinim programima za video nadzor zbog jednostavnije naknadne analize pojedinaca snimljenih sigurnosnim kamerama. Savremeni digitalni fotoaparati koriste detekciju lica za potrebe automatskog fokusiranja. Pojedine društvene mreže kao i programi za organizovanje kolekcija fotografija imaju mogućnost obeležavanja lica na fotografijama koje korisnik kasnije može da poveže sa pojedinim osobama. Lice koje je jednom označeno kao određena osoba se dalje može koristiti za prepoznavanje ostalih, neoznačenih lica. Detekcija lica se, u kombinaciji sa prepoznavanjem gestikulacije, koristi i u interfejsima između čoveka i računara.

Svrha ovog rada je predstavljanje jednog algoritma za detekciju lica koji se zasniva na modeliranju boje kože i detekciji kontura i ivica, uz kratak opis programske implementacije i analizu dobijenih rezultata, uključujući i poređenje sa rezultatima drugih srodnih algoritama.

Rad je podeljen na nekoliko delova. U poglavlju 2 biće reči o digitalnim slikama, načinima njihovog formiranja, tipovima slika i primeni. U poglavlju 3 su opisane neke metode obrade digitalnih slika. Poglavlje 4 je posvećeno modelima i prostorima boja, uz nekoliko konkretnih primera. Zatim se u poglavlju 5 govori o prepoznavanju objekata na slici, a u poglavlju 6 o detekciji lica pomoću različitih algoritama, uz nešto detaljniji osvrt na algoritme koji koriste boju kože. Poglavlje 7 je posvećeno predloženom algoritmu za detekciju lica, uz detaljan opis svih koraka. U poglavlju 8 je kratko opisana programska implementacija predloženog

algoritma, dok se u poglavlju 9 analiziraju rezultati algoritma kroz primere, i porede sa rezultatima srodnih algoritama. Na kraju, zaključak je izložen u poglavlju 10.

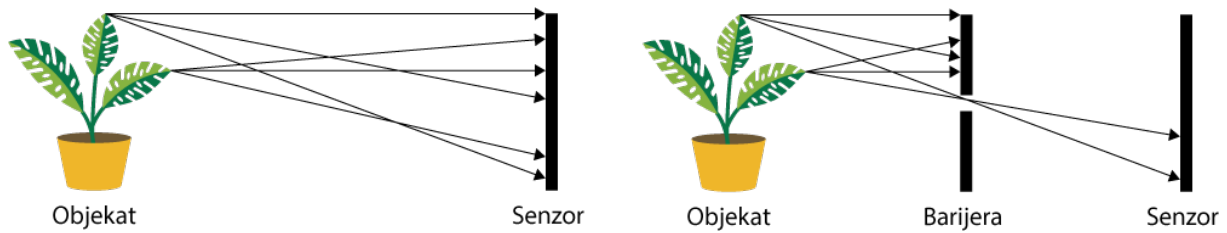
2 Digitalne slike

Digitalna slika je numerička (najčešće celobrojna) reprezentacija dvodimenzionalne slike. Vektorske slike, odnosno njihov sadržaj, opisan je prostim geometrijskim objektima, kao što su tačke, prave, krive i poligoni. Svaki od ovih objekata je predstavljen relativno malom količinom podataka, kao što su pozicija u koordinatnom sistemu, dimenzije, stil i boja linije i stil i boja ispunje objekta. Ovakav način predstavljanja slika ima brojne prednosti, kao što su mala veličina fajlova, mogućnost uvećanja slike bez gubitka kvaliteta i jednostavno menjanje slike koje podrazumeva menjanje i brisanje postojećih i dodavanje novih objekata. Ipak, primena vektorske grafike često nije pogodna, kao u slučaju prikazivanja fotografija, koje je teško aproksimirati geometrijskim objektima bez bojazni da će slika izgledati neprirodno. U ovakvim slučajevima je pogodnije koristiti rasterske slike, na koje se najčešće i misli kada se govori o digitalnim slikama.

Kod rasterskih slika podaci su predstavljeni u vidu konačne matrice, u kojoj se svaki element naziva piksel (engl. *pixel*, skraćeno od *picture element*, odnosno element slike). Svaki piksel, kao najmanja jedinica slike, sadrži podatke o boji i intenzitetu, dok je njegov položaj na slici određen položajem unutar matrice.

2.1 Formiranje digitalnih slika

Neophodni faktori formiranja digitalnih slika su energija koja dolazi od željenog objekta, sistem za fokusiranje te energije, i senzor za merenje količine energije. Potrebna energija obično dolazi od elektromagnetnih talasa, odnosno fotona, koji imaju svoju energiju, frekvenciju i talasnu dužinu. Često korišćeni su elektromagnetni talasi iz vidljivog dela spektra, odnosno vidljiva svetlost. Svetlost se usmerava ka željenom objektu. Odbijena svetlost se potom detektuje pomoću optičkog sistema. Fotosenzitivni materijal je moguće postaviti u blizini objekta, ali će u tom slučaju doći do mešanja odbijene svetlosti sa njegovih različitih delova, kao i svetlosti iz okruženja, što rezultuje praktično neupotrebljivom slikom. Rešenje je postavljanje barijere (slika 1) sa malim otvorom između objekta i materijala. Dobijena slika je izvrnuta, ali se obično vrši hardverska ili softverska korekcija kako posmatrač ovo ne bi primetio.



Slika 1: Osnovni koncept optičkog sistema bez barijere (levo) i sa barijerom (desno)

Mana korišćenja barijere je to što premalo svetlosti dolazi do senzora, pa se zato koriste optički sistemi sa sočivom. Sočivo se postavlja kako bi svetlost bila fokusirana na senzor. Kombinacija sočiva različitih zakrivljenja i promene udaljenosti između sočiva i senzora rezultuje mogućnošću posmatrača da posmatrani objekat uveličava i smanjuje, odnosno bolje fokusira svetlost i izoštrava sliku. Poslednji korak je merenje intenziteta svetlosti pomoću senzora, koji se sastoji od dvodimenzionalne mreže ćelija. Svaka ćelija meri količinu svetlosti koja padne na nju i pretvara je u električni napon, koji se potom pretvara u digitalni broj koji predstavlja vrednost jednog piksela slike.

Iako su digitalne slike formirane pomoću vidljive svetlosti možda i najčešće, u iste svrhe je moguće koristiti i talase iz ostatka elektromagnetnog spektra. Rendgenske slike (slika 2) se formiraju postavljanjem objekta između izvora rendgenskog zračenja i senzora koji meri intenzitet zračenja koji prolazi kroz određeni deo objekta. U slučaju ljudskog tela, kosti zaustavljaju rendgenske zrake, dok ih drugi unutrašnji organi u manjoj ili većoj meri propuštaju. Termalne slike se zasnivaju na detekciji infracrvenih zraka koji predstavljaju toplotu koju emituje neko telo. Rezultujuća slika se sastoji iz piksela od kojih svaki predstavlja temperaturu odgovarajućeg dela objekta. Slike zasnovane na gama zračenju se često koriste u medicini. Pacijentu se ubrizgava radioaktivni izotop koji emituje gama zrake tokom raspada, a odgovarajući senzor ih registruje. Gama zračenje se koristi i u astronomiji, s tim što se slike formiraju pomoću prirodne radijacije posmatranog nebeskog tela. Mikrotalasni deo spektra se koristi u radarskim sistemima. Radarska antena odašilje mikrotalase, a slika se formira na osnovu mikrotalasa odbijenih od tela. Radio talasi nalaze svoju primenu u medicini kod upotrebe magnetne rezonance, gde se radio talasi odašilju kroz ljudsko telo i uzrokuju odgovarajući povratni talas koji emituje tkivo. U astronomiji se koriste u takozvanim radio teleskopima, koji poseduju jednu ili više paraboličnih antena, obično udaljenih od većih naselja kako bi se izbeglo mešanje sa drugim izvorima radio talasa.



Slika 2: Rendgenski snimak ramena

Pored elektromagnetnih talasa, digitalne slike je moguće formirati koristeći i druge izvore. U geološkim istraživanjima često se koristi zvuk za pronalaženje nalazišta nafte. Velike čelične ploče koje vibriraju na određenoj frekvenciji se postavljaju na tlo, a na osnovu jačine i brzine povratnih zvučnih talasa formira se slika sastava zemljišta. U pomorskim istraživanjima obično se koriste vazdušni topovi koji se nalaze iza broda. Povratni zvuk detektuju specijalni senzori, a dobijeni podaci se koriste za formiranje trodimenzionalne mape. Primena ultrazvuka je česta u medicini (slika 3), naročito u ginekološkim ispitivanjima, ali i u drugim oblastima. Ultrazvuk se koristi i za otkrivanje jata riba ali i podmornica, korišćenjem sonara. U industriji primenu nalazi kroz akustičnu mikroskopiju, koja se koristi za detekciju naprslina nevidljivih ljudskom oku. Elektronska mikroskopija, za razliku od optičke, umesto svetlosti koristi usmereni snop elektrona, koji reaguju u sudaru sa posmatranim objektom. Reakcije se detektuju i pretvaraju u sliku, slično kao što bi se radilo sa odbijenom svetlošću kod konvencionalnih mikroskopa.



Slika 3: Slika dobijena pomoću ultrazvuka

Na kraju, digitalne slike ne moraju biti formirane na osnovu fizičkog objekta, već se mogu kompjuterski generisati. Jedan primer su fraktali, skupovi koji formiraju oblike koji obično imaju osobinu da su pojedini njihovi delovi približno slični celom objektu. Fraktali se definišu određenim matematičkim pravilima. Drugi primer su slike generisane pomoću trodimenzionalnih modela, koje se koriste u pojedinim simulatorima kao deo treninga, ali i u filmovima i video igrama.

2.2 Primena digitalnih slika

Jedna od prvih primena digitalnih slika je bila za potrebe novinskih kompanija. Bartlejn sistem za prenos slika su izumeli H. Bartolomju i M. MekFarlejn [1] 1920. godine, a prvi put je korišćen 1921. godine za slanje slika podmorskim kablom između Londona i Njujorka. Ovaj sistem je omogućio da se slika pošalje preko Atlantskog okeana za manje od tri časa, umesto nekoliko dana koliko je do tada bilo uobičajeno. Korišćena je specijalna štamparska oprema za kodiranje slike pre slanja, kao i rekonstrukciju signala i štampanje slike na odredištu (slika 4). Originalni Bartlejn sistem je kodirao slike u pet nivoa sive boje, a do 1929. godine je unapređen povećanjem nivoa sive na petnaest.



Slika 4: Slika preneti ranim Bartlejn sistemom

Iako su slike koje su prenošene na ovaj način bile digitalne, tek sa razvojem računara bila je moguća njihova obrada. Računarska obrada se obično vezuje za razvoj svemirskog programa početkom 1960-ih i dostupnost dovoljno jakih računara u tom periodu. Prve računarski obrađene slike bile su fotografije meseca koje je napravila sonda Rendžer 7, a na kojima je došlo do distorzije. Primenjene metode su dalje razvijane i korišćene prilikom narednih misija. Uporedo sa svemirskim programom, obrada digitalnih slika je počela da se primenjuje u medicini, astronomiji, industriji i brojnim drugim oblastima, a sa padom cena računara, umrežavanjem i razvojem interneta, područje primene je bilo sve šire.

2.3 Tipovi digitalnih slika

Kod binarnih slika se koristi jedan bit za predstavljanje jednog piksela. Kako bit ima svega dva stanja, 0 i 1, to znači da je svaki piksel u jednoj od dve predefinisane boje, a najčešće su to crna i bela. Nemogućnost prikazivanja međunijansi sive je veliko ograničenje za binarne slike u prikazivanju fotografija.

Slike u nijansama sive (eng. *grayscale*), koje se često nazivaju i crno-belim slikama, sačinjene su od piksela čije boje su različiti intenziteti sive, u rasponu od bele do crne boje. Najčešće se radi o 256 nivoa intenziteta sive, pa je za prikaz jednog piksela potreban jedan bajt. Imajući u vidu da je eksperimentima pokazano da ljudsko oko može da razlikuje najviše 200 nijansi sive, 256 nivoa je i više nego dovoljno da prikaže pun spektar između bele i crne bez uočljivih prelaza, pa je ovaj tip digitalnih slika pogodan za prikaz fotografija.

Slike u boji čine pikseli koji pored informacija o intenzitetu sadrže i informacije o boji. Za vizuelno prihvatljive rezultate, neophodne su tri vrednosti, odnosno kanala, koji opisuju boju svakog piksela u određenom prostoru boja. Najčešće korišćeni prostor boja je RGB, ali u upotrebi su i brojni drugi, kao što su CMYK, HSV, YCbCr itd. Kod RGB prostora se koristi 256 nivoa za svaku komponentu, tako da su potrebna tri bajta za prikaz jednog piksela. Ukupan broj boja koji se može prikazati u ovom prostoru je 16 777 216.

Indeksirane slike u boji imaju ograničenu paletu boja. Nazivaju se indeksirane zato što je boja piksela predstavljena indeksom, odnosno pozicijom te boje u predefinisanoj paleti. Kod prikazivanja fotografija pomoću indeksiranih slika dolazi do izvesnog gubitka informacija, ali to u mnogome zavisi od broja i izbora boja u paleti, kao i od opsega boja fotografije. Zbog ograničenja starijih monitora koji su mogli da prikažu samo 256 boja, bilo je pokušaja da se formira paleta boja koja bi bila najbolje prilagođena tadašnjem hardveru. Izabrano je 216 boja, koje se dobijaju kombinacijom po 6 nijansi crvene, zelene i plave. Ovo je paleta takozvanih *web-safe* boja, koja je korišćena kako bi slike bile prikazane isto na svim monitorima. Kako je danas većina monitora, pa čak i ekrana mobilnih uređaja 24-bitna, *web-safe* paleta se jako retko koristi.

3 Obrada digitalnih slika

Metode obrade slika se mogu podeliti u dve osnovne grupe, zavisno od vrste ulaznih podataka i rezultata obrade [2]. Prvu grupu čine metode čiji su ulazni i izlazni podaci slike, dok drugu grupu čine metode čiji ulazni podatak može biti slika, ali izlazni podaci su neki atributi izvedeni sa slike. U prvu grupu spadaju formiranje slika, poboljšanje, restauracija, obrada slika u boji, obrada pomoću talasića i kompresija. U drugu grupu spadaju segmentacija, reprezentacija i prepoznavanje objekata. Morfološka obrada za izlaz može imati sliku ali i neke attribute, pa ona tako pripada i prvoj i drugoj grupi.

Formiranje digitalnih slika je jedan od osnovnih procesa koji obično podrazumeva korišćenje hardvera za njihovo dobijanje na osnovu nekog fizičkog objekta. Formiranje obično uključuje i određenu vrstu preprocesiranja, kao što su skaliranje ili kompresija.

Poboljšanje slika je subjektivno, jer različiti posmatrači mogu različito oceniti kvalitet digitalne slike. Takođe, poboljšanje zavisi i od područja primene. Metode pogodne za poboljšanje fotografije možda neće biti pogodne za poboljšanje rendgenskog snimka i obratno. Algoritmi za poboljšanje slika se dele na one koji rade u spacijalnom, i one koji rade u frekventnom domenu.

Restauracija slika je za razliku od običnog poboljšanja zasnovana na matematičkim modelima degradacije slike, a ne na subjektivnoj proceni posmatrača. Tipičan primer je uklanjanje šuma.

Obrada slika u boji podrazumeva specifične algoritme koji u obzir uzimaju i informacije piksela o boji, a ne samo o intenzitetu kao u slučaju nijansi sive. Razumevanje modela i prostora boja je ključno za ove algoritme.

Talasići su mali talasi promenljive frekvencije i ograničenog trajanja. Obrada slika pomoću talasića otvorila je vrata piramidalnoj analizi i multirezolucionalnoj analizi.

Kompresija za cilj ima smanjenje prostora koji slika zauzima na disku, odnosno brži prenos preko mreže, uz što manji gubitak informacija koje slika sadrži. Tipičan primer je JPEG kompresija.

Morfološka obrada slika podrazumeva izdvajanje delova slike koji formiraju određeni oblik. Rezultat morfološke obrade može biti modifikovana slika, ali i određeni atributi nekog oblika.

Segmentacija predstavlja izdvajanje željenih objekata na slici. Ona se obično vrši na osnovu diskontinuiteta ili sličnosti intenziteta. Princip diskontinuiteta se zasniva na detekciji naglih promena u intenzitetu, kao što je slučaj kod traženja ivica. Sličnost intenziteta podrazumeva izdvajanje delova slike sličnog intenziteta, na primer kod vojne upotrebe termalnih kamera za detekciju viših temperatura koje mogu predstavljati ljudstvo ili opremu.

Reprezentacija uglavnom prati segmentaciju i dodatno opisuje izdvojene regione slike, bilo da je u pitanju samo granica ili cela površina regiona. Rezultat su podaci pogodni za matematičku analizu i računarsku obradu.

Prepoznavanje je uočavanje specifičnih objekata upoređivanjem atributa dela slike sa predefinisanim atributima koji najbolje opisuju jedan takav objekat. Neki od primera su uočavanje lica, pešaka ili vozila.

Druga potencijalna podela metoda za obradu slika uzima u razmatranje vrstu podataka koje koristi kao ulazne, njihovo poreklo, kao i složenost samog postupka. Tako se došlo do metoda niskog, srednjeg i visokog nivoa.

Metode obrade slika niskog nivoa koriste podatke direktno sa slike, odnosno informacije sačuvane na nivou piksela. Rezultati mogu biti izdvojene ivice, uglovi, zatvorene konture, ali ne i složenije informacije, kao što su vrste prikazanih objekata, njihov međusobni položaj ili položaj u odnosu na posmatrača.

Metode srednjeg nivoa koriste podatke dobijene primenom metoda niskog nivoa. Rezultati su složenije informacije o slici, kao što su prepoznavanje oblika, međusobni položaj objekata u prostoru i slično.

Metode visokog nivoa su ujedno i najsloženije. Njihov cilj je interpretacija podataka dobijenih metodama srednjeg nivoa. Rezultat su informacije koje posmatrač dobija proučavanjem slike, pa se ove metode mogu smatrati modeliranjem ljudske vizuelne percepcije. Kao takve, usko su povezane sa veštačkom inteligencijom.

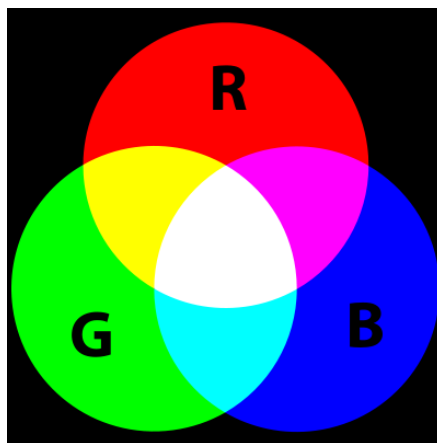
4 Modeli i prostori boja

Model boja je apstraktan matematički model koji se koristi za predstavljanje boja pomoću torki (tipično trojki) brojeva. Kombinovanje modela sa preciznim opisom interpretacije njegovih komponenti (uslovi gledanja i slično) formira prostor boja. Moguće je formirati prostor boja i bez modela. Takvi prostori su u suštini set boja kojima su dodeljeni nazivi. Tipičan primer je *Pantone Matching System* [3].

Jedan od prvih precizno definisanih prostora boja je CIE XYZ, kreiran od strane *International Commission of Illumination* 1931. godine [4]. Zasnovan je na CIE RGB modelu, za čije formiranje su iskorišćeni podaci dobijeni eksperimentima koje su krajem 20-tih godina dvadesetog veka izvodili V. D. Rajt [5] i Dž. Gild [6].

4.1 RGB

RGB model ima tri primarne komponente: crvenu, zelenu i plavu (eng. *red, green, blue*). Ovo je aditivni sistem, što znači da se sve boje dobijaju dodavanjem različitih intenziteta tri primarne komponente. Crna boja predstavlja odsustvo sve tri komponente, dok se bela dobija kombinovanjem sve tri komponente maksimalnih intenziteta (slika 5).



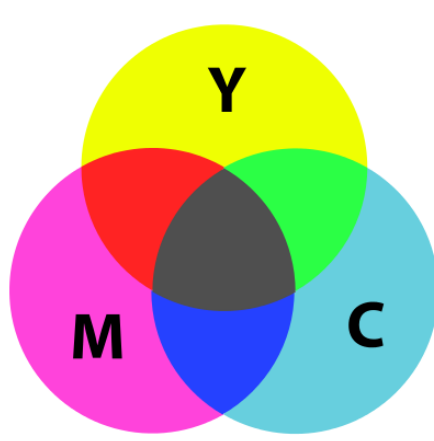
Slika 5: Mešanje R, G i B komponenti

Uređaji sa RGB ulazom su kamere, foto-aparati, skeneri i drugi, dok RGB izlaz imaju televizori, kompjuterski monitori, ekrani mobilnih uređaja, projektori i slični uređaji koji samostalno emituju svetlost. Vrednosti pojedinačnih komponenti se mogu predstaviti u intervalu od 0 do 1, koji se uglavnom koristi u teorijskoj analizi. Ređe se koristi reprezentacija u procentima. Najčešće korišćeni interval je od 0 do 255, odnosno korišćenje 8 bita za predstavljanje svake pojedinačne komponente, što rezultuje mogućnošću prikaza 16 777 216

različitih boja. Pojedini uređaji imaju mogućnost reprodukcije čak i većeg broja boja od onog ograničenog sa 24 bita, pa se za reprezentaciju boje koriste 30, 36 ili 48 bita.

4.2 CMYK

CMYK model je substraktivni sistem, koji se koristi kod uređaja koji ne emituju svetlost, već se zasnivaju na refleksiji, kao što su štampači. Osnovne komponente su cijan, magenta i žuta (eng. *cyan, magenta, yellow*) kojima se dodaje i crna zbog reprodukcije tamnijih nijansi. Cijan komponenta apsorbira crvenu svetlost, ali reflektuje zelenu i plavu. Magenta apsorbira zelenu svetlost i reflektuje crvenu i plavu, dok žuta apsorbira plavu, a reflektuje crvenu i zelenu svetlost (slika 6). Crna komponenta se koristi za predstavljanje odsustva reflektovane svetlosti, a kao posledica toga, savremeni štampači u boji imaju odvojen crni toner, odnosno kertridž, zbog nemogućnosti, ali i nepraktičnosti kvalitetnog prikazivanja crne boje kombinovanjem preostala tri pigmenta.



Slika 6: Mešanje C, M i Y komponenti

Ako su RGB vrednosti u intervalu od 0 do 255, jedna od mogućih formula za konverziju u CMYK je:

$$R' = \frac{R}{255}$$

$$G' = \frac{G}{255}$$

$$B' = \frac{B}{255}$$

$$K = 1 - \max(R', G', B')$$

$$C = \frac{(1 - R' - K)}{(1 - K)}$$

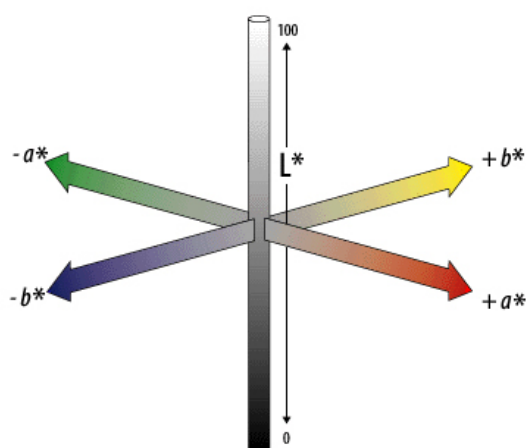
$$M = \frac{(1 - G' - K)}{(1 - K)}$$

$$Y = \frac{(1 - B' - K)}{(1 - K)}$$

Dobijene vrednosti su u intervalu od 0 do 1, ali uobičajeno je skaliranje do 100, odnosno određivanje pojedinačnih komponenti u procentima. Kako RGB i CMYK zavise od uređaja koji reprodukuje boje određenog modela, formula za konverziju nije jedinstvena, već se one definišu uz pomoć profila boja.

4.3 CIELAB

CIELAB je model čija je specifikacija formirana od strane međunarodne organizacije CIE (fr. *Commission internationale de l'éclairage*), koja je svojevrsni autoritet za boje, osvetljenje i modele i prostore boja. CIELAB je zamišljen kao model nezavistan od uređaja kojim će moći da se opišu sve boje vidljive ljudskom oku. Pored toga, ovaj model obuhvata i određene boje koje ljudsko oko ne vidi i nije ih moguće reprodukovati ni na jednom uređaju, pa je njihova upotreba uglavnom teoretska.



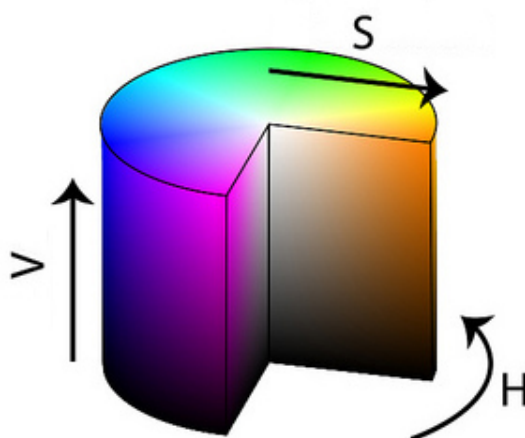
Slika 7: Grafički prikaz CIELAB modela boja

Zvanični naziv ovog standarda je 1976 CIE $L^*a^*b^*$, kako bi se nazivi komponenti razlikovali od onih iz 1948 L, a, b modela. L^* komponenta predstavlja osvetljenost i njene vrednosti se kreću od 0, odnosno crne, do 100, odnosno bele. Komponente a^* i b^* predstavljaju boju, i vrednosti mogu biti i pozitivne i negativne. Ove dve vrednosti se formiraju na osnovu

suprotnih boja, odnosno činjenice da jedna boja ne može biti istovremeno crvena i zelena, odnosno žuta i plava. Tako pozitivne vrednosti a^* komponente predstavljaju crvenu, a negativne zelenu boju. S druge strane, pozitivne vrednosti b^* komponente predstavljaju žutu, a negativne plavu boju (slika 7). Iako je ovo najprecizniji model boja, retko se koristi za neposredni prikaz, već je njegova primena češća u konverziji kao svojevrsni međumodel.

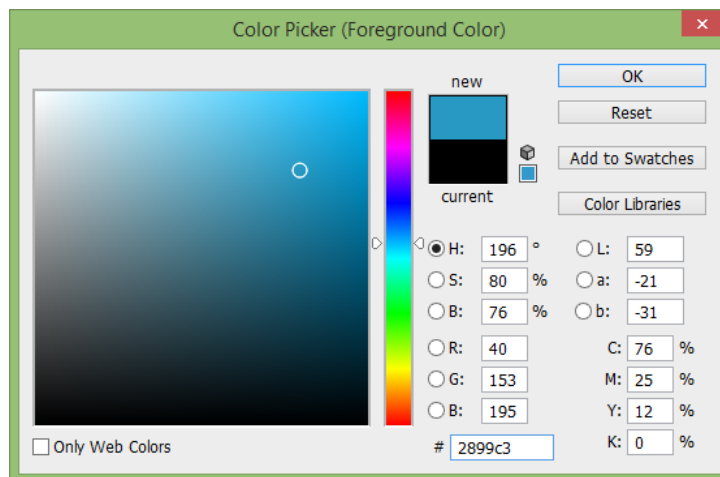
4.4 HSV

HSV je model koji je izveden iz RGB modela, kao pokušaj da se boje predstavljaju parametrima koji su bliži ljudskom poimanju boja, umesto neintuitivnog aditivnog modela. HSV komponente su nijansa, zasićenost i vrednost (eng. *hue, saturation, value*). Ovaj model se ponekad naziva i HSB (eng. *hue, saturation, brightness*). Može se predstaviti cilindrom (slika 8) kod koga je nijansa određena uglom oko vertikalne ose, zasićenost udaljenošću od vertikalne ose, a intenzitet pozicijom duž ose.



Slika 8: HSV cilindar

Formule za konvertovanje iz RGB modela u HSV i obratno su jednostavne pa je prebacivanje iz jednog u drugi model moguće u realnom vremenu. Zbog toga se HSV često koristi u programima za obradu digitalnih slika kod paleta za izbor boja (slika 9).



Slika 9: Paleta za izbor boja u programu Adobe Photoshop

Konverzija iz RGB modela u HSV, uz pretpostavku da su RGB vrednosti u intervalu od 0 do 255, se vrši prema sledećim formulama:

$$R' = \frac{R}{255}$$

$$G' = \frac{G}{255}$$

$$B' = \frac{B}{255}$$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

$$H = \begin{cases} 60^\circ \cdot \left(\frac{G' - B'}{\Delta} \text{ mod } 6 \right), & C_{max} = R' \\ 60^\circ \cdot \left(\frac{B' - R'}{\Delta} + 2 \right), & C_{max} = G' \\ 60^\circ \cdot \left(\frac{R' - G'}{\Delta} + 4 \right), & C_{max} = B' \end{cases}$$

$$S = \begin{cases} 0, & \Delta = 0 \\ \frac{\Delta}{C_{max}}, & \Delta \neq 0 \end{cases}$$

$$V = C_{max}$$

5 Prepoznavanje objekata

Objekat na slici može biti opisan specifičnim šablonom, odnosno obrascem. Šablon predstavlja skup atributa koji opisuju objekat. Klasa šablona je porodica šablona koji poseduju određene sličnosti. Prepoznavanje šablona, odnosno objekta, se svodi na pridruživanje nepoznatog šablona jednoj od definisanih klasa. Naravno, poželjno je da ovaj proces bude automatski ili uz minimalnu ljudsku intervenciju.

Postoje dva osnovna pristupa prepoznavanju objekata. Prvi se zasniva na teoriji odlučivanja i koristi šablone sa kvantitativnim atributima. Tipičan primer su vektori, predstavljeni matricom veličine $n \times 1$:

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

gde su elementi matrice a_i atributi koji opisuju šablon. Drugi pristup koristi strukturne metode i šablone sa kvalitativnim atributima, kao što su niske i drveta. Niske predstavljaju niz elemenata koji osim kvantitativnih atributa sadrže i podatke o međusobnim odnosima. Ako za primer uzmemo konturu stepenica, jasno je da se ona može opisati vektorom, ali je pogodnije opisati je sa svega dva elementa koji predstavljaju horizontalnu i vertikalnu ivicu i koji će ponavljanjem formirati nisku. Na ovaj način je moguće prepoznati konturu stepenica bez obzira na njihovu dužinu. Drveta su pogodna za opisivanje objekata kod kojih je moguće uočiti neku hijerarhijsku strukturu.

Suština pristupa koji koristi teoriju odlučivanja je dodeljivanje šablona x jednoj od klasa šablona $\omega_1, \omega_2, \dots, \omega_W$ pomoću funkcija $d_1(x), d_2(x), \dots, d_W(x)$ pridruženih svakoj od klasa. Potrebno je da funkcije imaju svojstvo da je, u slučaju da x pripada klasi ω_i :

$$d_i(x) > d_j(x), j = 1, 2, \dots, W, j \neq i$$

Jedan od načina za formiranje ovih funkcija je predstavljanje svake klase prototipom, odnosno jednim šablonom koji je najbolje opisuje. Funkcije se mogu definisati kao euklidska udaljenost vektora nepoznatog šablona i vektora prototipa. U slučaju da je poznat određen broj šablona koji pripadaju klasi, prototip je moguće formirati kao srednji vektor:

$$m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{ij}, i = 1, 2, \dots, W$$

gde je W broj klasa, a N_i broj vektora x_{ij} koji pripadaju klasi ω_i ($j = 1, 2, \dots, N_i$). Potom je potrebno izračunati euklidsku udaljenost:

$$D_i(x) = \|x - m_i\|$$

Kako je neophodno da funkcija $d_i(x)$ postigne maksimum za prototip najbližiji nepoznatom šablonu, moguće je definisati je na sledeći način:

$$d_i(x) = x^T m_i - \frac{1}{2} m_i^T m_i$$

Ova metoda daje dobre rezultate u slučaju da je udaljenost između prototipova velika u poređenju sa udaljenostima prototipova od šablona koji pripadaju njihovoj klasi.

Optimalni statistički klasifikatori koriste verovatnoću za određivanje klase šablona. Ako verovatnoću da šablon x pripada klasi ω_j označimo sa $p(\omega_j|x)$, a gubitak prilikom dodeljivanja šablona x klasi ω_i iako on pripada klasi ω_j označimo sa L_{ji} , možemo definisati prosečan gubitak prilikom dodele šablona x klasi ω_i :

$$r_i(x) = \sum_{k=1}^W L_{ki} p(\omega_k|x)$$

Primenom Bajesovog pravila:

$$p(A|B) = p(A) \frac{p(B|A)}{p(B)}$$

dobijamo formulu:

$$r_i(x) = \frac{1}{p(x)} \sum_{k=1}^W L_{ki} p(x|\omega_k) P(\omega_k)$$

gde je $P(\omega_k)$ je verovatnoća pojavljivanja klase ω_k . $\frac{1}{p(x)}$ je moguće eliminisati, pošto je zajedničko za svako i , pa tako dobijamo formulu:

$$r'_i(x) = \sum_{k=1}^W L_{ki} p(x|\omega_k) P(\omega_k)$$

Formulu je moguće dodatno pojednostaviti ako gubitak predstavimo kao:

$$L_{ji} = 1 - \delta_{ji}$$

gde je $\delta_{ji} = 1$ za $j = i$, odnosno $\delta_{ji} = 0$ za $j \neq i$, dobijamo:

$$r'_i(x) = \sum_{k=1}^W (1 - \delta_{ji}) p(x|\omega_k) P(\omega_k) = p(x) - p(x|\omega_i) P(\omega_i)$$

Slično kao ranije, $p(x)$ je moguće eliminisati, pošto je zajedničko za svako i . Tako se dolazi do funkcije

$$d_i(x) = p(x|\omega_i) P(\omega_i)$$

Vektor x se dodeljuje onoj klasi za koju funkcija ima najveću vrednost.

U pomenutim primerima za svaku klasu postoji veći broj šablona koji im pripadaju i na osnovu kojih se svojevrsnim treningom dolazi do klasifikatora. Klasifikator se više ne menja, i rezultati kvalifikacije zavise isključivo od podataka korišćenih za trening. Ukoliko ovi podaci nisu dostupni unapred, mogu se koristiti neuronske mreže, pomoću kojih se klasifikator trenira i po potrebi menja sa svakim obrađenim vektorom. Tipičan primer je klasifikacija pomoću perceptrona.

Sa druge strane, pristup klasifikaciji niski je nešto drugačiji. Procena sličnosti dve niske a i b može se izvršiti prostim prebrojavanjem elemenata dve niske koji imaju istu poziciju i koji su jednaki. Ako ovaj broj označimo sa α , možemo izračunati broj pozicija koje se ne poklapaju:

$$\beta = \max(|\alpha|, |\beta|) - \alpha$$

gde su $|\alpha|$ i $|\beta|$ dužine odgovarajućih niski, a funkcija maksimuma se koristi jer niske nisu nužno iste dužine. Procena sličnosti može se predstaviti formulom:

$$R = \frac{\alpha}{\beta} = \frac{\alpha}{\max(|\alpha|, |\beta|) - \alpha}$$

Rezultat je beskonačno u slučaju apsolutnog poklapanja, a nula u slučaju da ne postoji par elemenata niske koji su jednaki. Problem koji se javlja kod zatvorenih kontura je određivanje početka niske. Metodom grube sile je moguće pomerati početak jedne od niski i potom računati sličnost onoliko puta koliko ona ima elemenata, ali je zbog optimizacije poželjno pronaći poziciju koja je makar blizu početka.

Što se tiče prepoznavanja oblika pomoću drveta, moguć je sintaksički pristup. Naime, stablo se opisuje gramatikom, primenom čijih pravila se može formirati odgovarajući oblik. Provera da li se nepoznato stablo može opisati tom gramatikom se vrši pomoću konačnog automata, strukture koja između ostalog sadrži konačan broj stanja, alfabet gramatike i pravila mapiranja elemenata alfabeta u stanja. Osnovno pravilo je da je odnos gramatike i automata $1 - 1$, odnosno jezik (u ovom slučaju stablo) će biti prepoznat ako je moguće generisati ga pomoću te gramatike.

6 Detekcija lica

Postoji veliki broj algoritama za detekciju lica, ali se svi mogu podeliti u četiri kategorije, predložene od strane Janga, Krigmana i Ahudže [7]: algoritmi zasnovani na znanju, algoritmi zasnovani na nepromenljivim karakteristikama lica, algoritmi koji koriste poklapanje šablona, i algoritmi zasnovani na izgledu.

Algoritmi zasnovani na znanju koriste postojeća znanja o karakteristikama ljudskog lica i pretvaraju ih u skup pravila. Neka od tih pravila su veoma jednostavna, kao na primer simetrija i prisustvo očiju, nosa i usta na regionu lica. Druga se dobijaju eksperimentalno, kao što su relativna rastojanja i pozicije pojedinih crta lica, koristeći veliki broj fotografija lica za određivanje prosečnih vrednosti. Problem kod ovakvih algoritama se javlja kod pretvaranja ljudskog znanja u skup pravila. Ako se koristi mali broj pravila, rezultati izvršavanja algoritma mogu sadržati veliki broj lažno pozitivnih rezultata, dok se kod korišćenja velikog broja pravila često javlja veliki broj lažno negativnih rezultata, odnosno lica koja neće biti detektovana. Jedno od mogućih unapređenja je upotreba hijerarhijskih pravila koje su opisali Jang i Huang [8]. Njihov algoritam sadrži tri nivoa pravila, gde najviši nivo opisuje lice na najopštiji način, dok se najniži nivo bavi detaljnim crta lica. Za potrebe ovog algoritma se pored originalne slike formiraju slike u nižoj rezoluciji. Što je nivo viši, rezolucija je niža zbog manjka potrebe za analizom finih detalja. Na prvom nivou se formira region, takozvani prozor, koji se pomera po slici i pritom analizira piksele koji ulaze u njegove okvire. Pravila koja se ovde primenjuju su relativno jednostavna i odnose se, između ostalog, na uniformnost intenziteta pojedinih regiona unutar prozora, kao i odnos intenziteta između različitih regiona. Potencijalni kandidati se obrađuju u višoj rezoluciji koristeći pravila drugog nivoa. Ovde se vrši lokalno ujednačavanje histograma [2] i detekcija ivica. Pravila trećeg nivoa se primenjuju na slici najviše rezolucije i koriste se za detekciju pojedinih crta lica kao što su oči i usta.

Algoritmi koji se zasnivaju na nepromenljivim karakteristikama lica se zasnivaju na činjenici da ljudi lako detektuju lica, bez obzira na pozu i osvetljenje. Osnovna ideja je dakle izdvajanje onih karakteristika lica koja ne zavise od poze, osvetljenja, niti bilo kojih drugih promenljivih uslova. Neki od ovih algoritama izdvajaju crte lica kao što su oči, obrve, nos i usta i koriste ih za određivanje celokupnog regiona lica. Drugi se, pak, zasnivaju na prepoznavanju tekstura karakterističnih za ljudsko lice, kao što su teksture kože i kose. Prepoznavanje kože na

osnovu boje takođe se primenjuje u algoritmima za detekciju lica. Brojni algoritmi koriste kombinacije prethodno pomenutih karakteristika lica, koristeći na primer boju kože i oblik lica za pronalaženje potencijalnih kandidata, a finije crte lica za dodatnu proveru.

Algoritmi zasnovani na poklapanju šablona koriste standardni šablon koji se formira tako da dobro opisuje lice, odnosno njegove crte. Lice se detektuje na osnovu korelacije između fotografije, odnosno jednog njenog dela, i pripremljenog šablona. Šabloni mogu biti predefinisani, i njihovo formiranje i korišćenje je relativno jednostavno, ali problem se javlja kada postoje varijacije u veličini, pozici i obliku. Ako je šablon formiran na osnovu *en face* slika lica, on neće biti pogodan za detekciju profila lica. Jedno od mogućih rešenja je formiranje većeg broja šablona za različite poze. Složeniji pristup je korišćenje deformabilnih šablona, koji se prilagođavaju mogućim varijacijama. Osim toga, moguće je korišćenje i podšablona, odnosno šablona za prepoznavanje pojedinačnih crta lica nakon detekcije lica u svrhe dodatne provere.

Algoritmi zasnovani na izgledu takođe koriste šablone, ali je njihovo formiranje zasnovano na statističkoj analizi i mašinskom učenju. Šablon se kreira na osnovu velikog broja slika na kojima se nalazi lice, ali i onih na kojima nema lica. Delovi slike se potom upoređuju sa šablonom, i oni koji sa određenom verovatnoćom zadovoljavaju uslove se obeležavaju kao detektovano lice.

Pomenute kategorije algoritama nisu strogo razdvojene, tako da postoje brojni algoritmi koji se mogu svrstati u dve, pa i više kategorija.

6.1 Modeliranje boje kože

Modeliranje boje kože i njena detekcija je tehnika koja se koristi ne samo za detekciju lica, već i u druge svrhe, kao što su detekcija ruke i šake za analizu i prepoznavanje gestova, filtriranje slika neprimerenih određenim uzrastima, video kompresiju zasnovanu na detekciji sadržaja, ujednačavanje boja, i druge. Postoje brojni pristupi pomenutom problemu, od kojih su neki opisani u [9] i [10].

Algoritam za modeliranje boje kože u HSI prostoru boja je opisan od strane Kjeldsena i Kendera [11]. Njihov metod ne zavisi od intenziteta, i svaki pojedinačni piksel se klasifikuje kao koža samo ako su nijansa i zasićenost unutar određenih granica.

Eksperimentalno se može doći do odgovarajućih granica i u drugim prostorima boja, kao što su to opisali Kukharev i Novosielski u slučaju YCbCr prostora [12]. U ovom algoritmu su, za razliku od prethodno pomenutog, određene granice za sve tri komponente.

Krouli i Koutaz [13] su osmislili jednostavan način za detekciju lica koji podrazumeva korišćenje normalizovanog histograma. Najveći problem ovog algoritma se javlja kada se na slici nalaze i drugi regioni kože, kao na primer ruka. Stoga je potrebna dodatna verifikacija takvih regiona, kao što je detekcija treptanja u slučaju video zapisa.

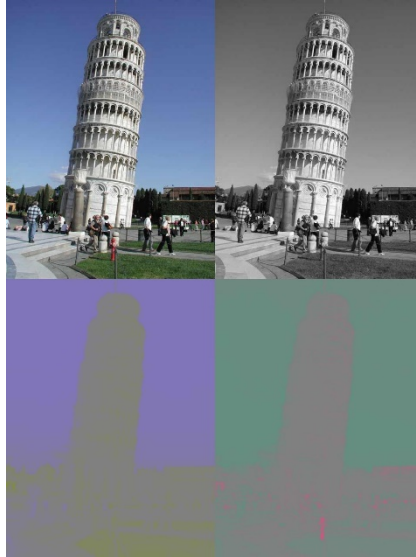
Pojedini algoritmi koriste više prostora boja, kao onaj predložen u [14], koji kombinuje RGB, HSI i YCbCr. Rezultati tri algoritma, od kojih se svaki zasniva na samo jednom prostoru boja, formiraju uniju, na taj način smanjujući broj lažno negativnih rezultata.

Veliki broj pomenutih algoritama za detekciju kože radi na nivou piksela, pa je jasno kako njihovo izvršavanje može da potraje jako dugo kod velikih slika. Mahmud [15] predlaže tehniku za brže izvršavanje, koja podrazumeva smanjivanje dimenzije slika na predefinisane vrednosti, kao i preskakanje predefinisano broja piksela kod testiranja na boju kože, imajući u vidu veliku verovatnoću da su susedi piksela koji je detektovan kao koža i sami pikseli kože. Ova tehnika je jako korisna kod prilagođavanja već poznatih algoritama za rad u sistemima u realnom vremenu, kao i za primene gde region kože nije potrebno odrediti potpuno precizno.

Nakon detekcije regiona kože, neophodna je dodatna validacija kako bi se potvrdilo da je u pitanju lice. Metoda opisana u [16] koristi detekciju crta lica u YCbCr prostoru boja.

6.2 YCbCr prostor boja

Algoritmi koji koriste YCbCr prostor boja za detekciju kože su se pokazali kao efikasni [2]. Iako je pokazano da TSL prostor boja daje bolje rezultate [17], YCbCr se češće koristi zbog svoje česte primene u standardima za kompresiju fotografija i video materijala, kao što su JPEG i MPEG, ali i zbog sličnosti sa TSL prostorom u smislu razdvajanja luminanse i hrominanse. Zajedno sa YUV i YIQ prostorima pripada porodici prostora koji se koriste za televizijske prenose. Y komponenta predstavlja luminansu, odnosno osvetljenost, dok Cb i Cr predstavljaju razliku plave, odnosno crvene (slika 10).



Slika 10: Slika u boji i njene Y, Cb i Cr komponente

Razdvajanje luminanse od hrominanse omogućuje smanjenje propusnog opsega, kompresiju i druge transformacije koje za posledicu imaju poboljšanje efikasnosti sistema. Jedna od najčešćih primena je smanjivanje propusnog opsega i svođenje slike u boji na nijanse sive, imajući u vidu da je ljudsko oko mnogo osetljivije na intenzitet nego na informacije o boji.

Konverzija iz često korišćenog RGB prostora boja u YCbCr je prilično jednostavna. Luminansa se računa kao težinska suma RGB komponenti, dok se komponente hrominanse računaju oduzimanjem prethodno dobijene luminanse od plave, odnosno crvene komponente.

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$$

$$Cb = B - Y$$

$$Cr = R - Y$$

Na sličan način je moguće izračunati i Cg komponentu kao razliku zelene boje, ali s obzirom da je $Cb + Cr + Cg$ konstantna vrednost, dovoljno je čuvati podatke o svega dve komponente koje opisuju boju.

6.3 Algoritmi za detekciju kože u YCbCr prostoru

Postoji veliki broj algoritama za detekciju kože koji koriste YCbCr prostor boja. Kukharev i Novosielski [12] su formirali model boje kože zasnovan na uzorku od 25 lica različitih rasa, isključujući osobe izrazito tamne boje kože. Uz normalizaciju Y, Cb i Cr komponenti na raspon

od 0 do 255, došlo se do sledećih vrednosti koje su dale najbolje rezultate na izabranim primerima:

$$Y > 80$$

$$85 < Cb < 135$$

$$135 < Cr < 180$$

Detekcija kože je samo prvi korak u njihovom algoritmu za detekciju i prepoznavanje lica, koji se sastoji i od analize crta lica, reduktora i poklapanja šablona. Nakon detekcije, sledi algoritam za prepoznavanje, koji se sastoji iz niza manje ili više modifikovanih metoda poznatih od ranije.

Čai i Ngan [18] su razvili algoritam koji koristi specijalne karakteristike boje ljudske kože. Model boje kože se formira na osnovu hrominanse piksela, nakon čega se primenjuju dodatne procedure koje izdvajaju regione sa većom verovatnoćom da budu lice. Eksperimentalnim putem se došlo od sledećih granica za Cb i Cr komponente:

$$77 \leq Cb \leq 127$$

$$133 \leq Cr \leq 173$$

Nepostojanje utvrđenih granica za Y komponentu za rezultat ima mogućnost detekcije kože ljudi svih rasa, ali uz nešto veći broj lažno pozitivnih rezultata nego što bi to bio slučaj da su određene različite granice za svaku rasu posebno.

Detekcija kože je često bitan korak u algoritmima za detekciju eksplicitnih slika. Prethodno pomenute granice nisu bile pogodne za rešavanje problema opisanog u [19], bilo zbog nemogućnosti detekcije različitih boja kože, bilo zbog velikog broja lažno pozitivnih rezultata. Nakon opsežne analize histograma na velikom uzorku, autori su došli do optimalnih granica za detekciju kože:

$$80 \leq Cb \leq 120$$

$$133 \leq Cr \leq 173$$

Na osnovu broja piksela proglašanih za kožu, određuje se njihov procenat na slici koja se ispituje. Ukoliko taj procenat pređe određenu granicu, slika se označava kao eksplicitna.

Imajući u vidu njegovu izuzetnu jednostavnost, ovaj algoritam daje prilično dobre rezultate, pošto eksplicitne slike uglavnom imaju veliki procenat površina koje predstavljaju kožu.

7 Predloženi algoritam za detekciju lica

Algoritam o kome će biti reči u ovom poglavlju je opisan od strane autora ovog rada i odobren za objavljivanje u [20].

7.1 Detekcija kože

Prvi korak algoritma za detekciju lica je primena algoritma za detekciju kože. Rezultat je binarna slika, na kojoj je za svaki piksel određeno da li je koža ili nije (slika 11). Na slici će se javiti i šum, sačinjen od piksela iz obe grupe.



Slika 11: Originalna slika (levo) i regioni detektovani kao koža (desno)

Na slici je jasno uočljiv region koji predstavlja lice, unutar koga su karakteristične crte lica predstavljene crnim pikselima, što znači da se ne uklapaju u izabrani model boje kože. Pored lica, vidljiv je i region vrata, ali i velika količina šuma, naročito na ivicama regiona kose, koje su zbog prelamanja svetlosti poprimile boju koja odgovara modelu, iako to na prvi pogled nije očigledno zbog toga što je veliki deo kose izrazito crn i u kontrastu sa bojom lica.

Uklanjanje šuma se obavlja operacijama otvaranja i zatvaranja, o kojima će biti više reči u poglavlju 7.2. Otvaranje se sastoji iz morfoloških operacija [21] erozije i potom dilatacije, dok se zatvaranje sastoji iz dilatacije, a zatim i erozije.

7.2 Morfološke operacije

Osnovne morfološke operacije se izvršavaju nad pojedinačnim pikselima slike, i kao rezultat vraćaju sliku istih dimenzija kao i original. Vrednost rezultujućeg piksela se izračunava na osnovu vrednosti susednih piksela originalne slike. Susedstvo piksela određuje takozvani strukturni element, koji je binarna matrica. Samo pikseli koji u matrici imaju vrednost 1 pripadaju susedstvu. Centralni piksel matrice predstavlja piksel čija se nova vrednost izračunava. Tipično je strukturni element mnogo manjih dimenzija nego originalna slika.

Kod erozije je vrednost rezultujućeg piksela minimalna vrednost susednih piksela. U slučaju binarne slike, dovoljno je da jedan od piksela ima vrednost 0, kako bi i rezultujući piksel dobio tu vrednost. Za potrebe izračunavanja vrednosti ivičnih piksela, podrazumeva se da pikseli van granica slike imaju maksimalnu vrednost, kako ne bi uticali na rezultat. Operacija erozije na binarnoj slici za rezultat ima oduzimanje piksela od objekta predstavljenog jedinicama.

Druga osnovna morfološka operacija, dilatacija, izračunava vrednost piksela kao maksimum susednih piksela, pa je kod binarne slike dovoljan jedan susedni piksel sa vrednošću 1, da i rezultujući piksel dobije istu vrednost. Pikseli van granica slike dobijaju minimalnu vrednost kako ne bi uticali na dilataciju ivičnih piksela. Rezultat dilatacije je dodavanje piksela na objekat koji čine pikseli sa vrednošću 1.

Otvaranje slike A strukturnim elementom B definiše se kao:

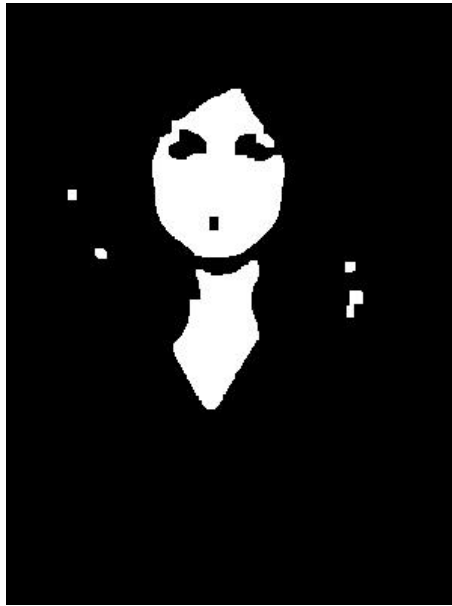
$$A \circ B = (A \ominus B) \oplus B$$

gde je \ominus operacija erozije, a \oplus operacija dilatacije. Posledica otvaranja na binarnoj slici je uklanjanje šuma koji čine pikseli sa vrednošću 1, ali i zaobljavanje uglova.

Zatvaranje slike A strukturnim elementom B definiše se kao:

$$A \bullet B = (A \oplus B) \ominus B$$

Rezultat na binarnoj slici je uklanjanje šuma koji čine pikseli sa vrednošću 0, odnosno zatvaranje manjih rupa u objektima koje čine pikseli sa vrednošću 1.



Slika 12: Regioni kože nakon primene morfoloških operacija

Na slici 12 je vidljivo da je velika količina šuma uklonjena, iako je preostalo nekoliko grupica belih piksela koji očigledno nisu koža. Ranije jasne i relativno detaljne, crte lica sada su transformisane u crne „mrlje”. Takođe, region vrata je sada još jasnije razdvojen od lica.

7.3 Izdvajanje regiona kože

Sledeći korak je detekcija regiona kože na binarnoj slici. Region kože sastoji se iz piksela kože takvih da između svaka dva piksela u regionu postoji bar jedna putanja koja vodi isključivo preko drugih piksela kože. Svaki od detektovanih regiona je potencijalno lice, ali regioni koji nemaju „rupe”, odnosno skupine piksela koji nisu detektovani kao koža unutar svojih granica, mogu sa velikom sigurnošću biti eliminisani, imajući u vidu da se boja pojedinih delova lica kao što su oči, usta i nozdrve dovoljno razlikuju od boje kože definisane ranije opisanim modelom.

Nije teško uočiti da svako lice ima veliki broj ivica koje formiraju pojedini njegovi delovi kao što su oči, nos i usta, dok je na drugim delovima tela taj broj srazmerno manji. Stoga je sledeći korak eliminacija preostalih regiona kože na osnovu gustine ivica (eng. *edge density*).

7.4 Kenijev detektor ivica

Postoji veliki broj algoritama za detekciju ivica na slici. Neki od najpoznatijih su opisani i upoređeni u [22]. Jedan od najčešće korišćenih algoritama osmislio je Dž. Keni [23] i poznat je pod nazivom Kenijev detektor ivica. Po rečima autora, optimalni algoritam za detekciju

ivica mora da postigne tri cilja. Prvi je mali stepen grešaka, što znači da algoritam mora da prepozna što veći broj ivica na slici, uz što manji broj lažno pozitivnih rezultata. Drugi cilj je dobra lokalizacija, odnosno minimizacija rastojanja između piksela prepoznatog kao ivica i stvarne ivice na slici. Treći cilj je minimalni odziv, što znači da ivica treba biti označena samo jednom, kao i da prisustvo šuma na slici ne sme uticati na pojavljivanje lažno pozitivnih rezultata.

Prvi korak je uklanjanje šuma primenom Gausovskog filtera [2], što rezultuje eliminacijom lažno pozitivnih ivica u vidu šuma:

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \cdot A$$

Rezultat je zamućena slika sa uklonjenim šumom.

Drugi korak je nalaženje gradijenta intenziteta slike. Primenjuju se četiri filtera za detekciju horizontalnih, vertikalnih i dijagonalnih ivica. Sobelov operator za detekciju ivica vraća prvi izvod u horizontalnom pravcu (G_x) i vertikalnom pravcu (G_y). Gradijent ivice i pravac se računaju na sledeći način:

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Pravac ivice se zaokružuje na jedan od četiri ugla koji predstavljaju vertikalu, horizontalu i dve dijagonale.

Treći korak je provera da li je intenzitet gradijenta lokalni maksimum u pravcu gradijenta. Na primer, ukoliko je ugao koji definiše pravac gradijenta jednak nuli, odnosno pravac je sever – jug, piksel će biti smatran delom ivice ukoliko je njegov intenzitet gradijenta veći od intenziteta gradijenata piksela istočno i zapadno od njega. Smer ne igra nikakvu ulogu, pa se tako ne pravi razlika između ugla od 0 stepeni i ugla od 180 stepeni, odnosno pravca sever – jug i jug – sever.

Poslednji korak je odbacivanje piksela koji ne pripadaju ivicama zavisno od njihovog intenziteta gradijenta. Za piksele sa visokim intenzitetom gradijenta se sa velikom sigurnošću može reći da zaista pripadaju ivicama, dok se za one sa niskim intenzitetom može reći da im sigurno ne pripadaju. Stoga se najpre formiraju dve granice intenziteta, gornja i donja. Pikseli sa intenzitetom gradijenta većim od gornje granice se proglašavaju za jake ivice koje su automatski uključene u rezultujuće ivice, dok se pikseli ispod donje granice odbacuju. Pikseli čiji je intenzitet između dve granice se proglašavaju za slabe ivice. Slabe ivice mogu biti pridodate rezultujućim ivicama, samo ako su povezane sa nekom jakim ivicom. Slabe ivice koje nisu povezane sa jakim ivicama se odbacuju kao šum. Rezultat algoritma je binarna slika na kojoj su pikseli ili proglašeni za ivicu ili nisu. Postoje brojne modifikacije ovog algoritma, kao što su one opisane u [24] i [25].

Kako Kenijev algoritam kao rezultat ima ivice debljine jednog piksela, poželjno je kandidate za lica skalirati na uniformnu veličinu, kako gustina ivica ne bi varirala u zavisnosti od veličine lica na slici. Nakon skaliranja kandidata na maksimalnu širinu i visinu od 100 piksela, uz zadržavanje odnosa stranica, vrši se operacija zatvaranja na binarnoj slici sa detektovanim pikselima kože, i to u više iteracija. Razlog je zatvaranje rupa koje predstavljaju oči i usta, oko kojih se inače nalazi najveći broj ivica na licu. Rezultat je površina koja bi, u slučaju da je kandidat zaista lice, pored piksela kože uključivala i ostale elemente lica koji inicijalno nisu bili detektovani. Za značajne ivice proglašavaju se samo one koje su detektovane unutar ove površine (slika 13), eliminišući okolne piksele koji bi potencijalno imali nesrazmerno veliki broj ivica (na primer kosa), a uključujući relevantne delove lica koji formiraju ivice (oči, nos, usta).



Slika 13: Region lica nakon izvršene operacije zatvaranja (levo) i detektovane ivice unutar tog regiona (desno)

7.5 Određivanje gustine ivica

Pojam gustine ivica je opisan u [26]. Naime, za sliku $i(x, y)$ intenzitet ivica označen je sa $e(x, y)$. Za region slike r sa gornjim levim uglom na poziciji (x_1, y_1) i donjim desnim uglom na poziciji (x_2, y_2) , gustina ivica se definiše kao:

$$f = \frac{1}{a_r} \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} e(x, y)$$

gde je a_r površina regiona, $a_r = (x_2 - x_1 + 1)(y_2 - y_1 + 1)$.

Zbog prirode algoritma prikazanog u ovom radu, pre svega činjenice da primena Kenijevog detektora ivica rezultuje binarnom slikom, ali i potrebe da se posmatraju samo ivice unutar potencijalnog kandidata, a ne i one koje formiraju okolni pikseli koji pripadaju pravougaonom regionu, bilo je potrebno prilagoditi formulu za izračunavanje gustine ivica:

$$g = \frac{1}{A} \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} e(x, y)c(x, y)$$

gde $e(x, y)$ uzima vrednost 1 za piksele koje je Kenijev algoritam proglasio za ivice, a 0 za ostale, dok $c(x, y)$ uzima vrednost 1 za piksele koji su unutar potencijalnog kandidata, a 0 za ostale. S tim u vidu se definiše i A :

$$A = \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} c(x, y)$$

Dobijena gustina ivica g je u intervalu $[0,1]$. Ovu vrednost je moguće normalizovati množenjem sa 100, čime se gustina efektivno svodi na procenete.

Bilo je potrebno odrediti pogodne granice gustine ivica u određenom regionu koje dobro opisuju lice i na osnovu kojih se pojedini kandidati mogu eliminisati. Jasno je da pogrešan odabir ovih granica može da ima negativne posledice na rezultate algoritma. Na primer, ako je donja granica isuviše niska, rezultati će sadržati veliki broj lažno pozitivnih rezultata, odnosno površina kože koje nisu lica. Broj lažno pozitivnih rezultata će biti veći na slikama na kojima se pojavljuje više različitih delova tela, kao što su fotografije celog tela. Nasuprot tome, suviše visoka donja granica će, pored eliminisanja lažno pozitivnih imati i nepovoljnu

posledicu velikog broj lažno negativnih, odnosno lica koja su greškom odbačena. Gornja granica, osim u slučaju da je jako niska, odnosno bliska donjoj granici i da na taj način značajno ograničava raspon koji služi za opisivanje lica, mnogo manje utiče na uspešnost algoritma. Suviše visoka gornja granica u teoriji može imati uticaja time što će povećati broj lažno pozitivnih rezultata, ali u praksi je najčešće upravo lice deo vidljivog dela kože koji sadrži najviše ivica. Zbog ovog minimalnog benefita, ali i potencijalne opasnosti da se pogrešnim određivanjem gornje granice umanji kvalitet rezultata, algoritam je pojednostavljen time što je zadržana samo donja granica.

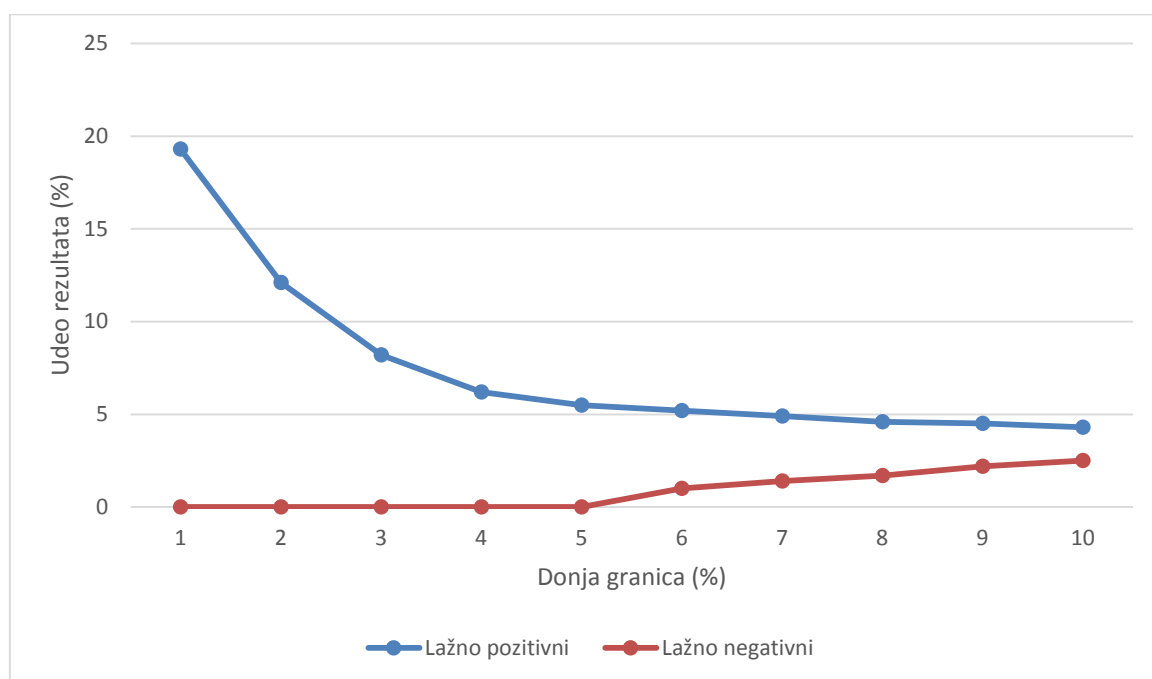
Određivanje pogodne granice je vršeno analizom regiona sa većeg broja slika koji su, u skladu sa prethodnim koracima algoritma, proglašeni za potencijalne kandidate. Kvalitet rezultata sledećeg koraka je određen za vrednosti granice između jednog i deset procenata, u koracima od po jednog procenta, prostim prebrojavanjem lažno pozitivnih i lažno negativnih rezultata, i izračunavanjem njihovog učešća u ukupnom broju rezultata. Kategorizacija potencijalnih kandidata ovde ne zavisi previše od subjektivnog utiska posmatrača, jer ovaj korak ne menja oblik ni poziciju regiona. Ovo znači da ako region nije idealno određen u prethodnim koracima (na primer ne obuhvata celo lice već samo jedan njegov deo), posmatrač će ga smatrati za pogodak (ili promašaj) nezavisno od promene donje granice, sve što je potrebno odrediti je da li se rezultat algoritma poklapa sa procenom posmatrača. Rezultati prebrojavanja lažno pozitivnih predstavljeni su procentualno u sledećoj tabeli:

Donja granica (%)	1	2	3	4	5	6	7	8	9	10
Lažno pozitivni (%)	19,3	12,1	8,2	6,2	5,5	5,2	4,9	4,6	4,5	4,3

Može se primetiti da je procenat lažno pozitivnih rezultata pri granici od jednog procenta relativno nizak, imajući u vidu i jako nisku graničnu vrednost. Razlog je u tome što su prethodni koraci algoritma eliminisali većinu regiona koji nisu lica, pa u preostalim regionima najveći procenat predstavljaju upravo regionima koji jesu lica. Rezultati analize lažno negativnih rezultata predstavljeni su u sledećoj tabeli:

Donja granica (%)	1	2	3	4	5	6	7	8	9	10
Lažno negativni (%)	0	0	0	0	0	1,0	1,4	1,7	2,2	2,5

Primetno je da je procenat lažno negativnih rezultata konstantan za vrednosti donje granice između jedan i pet odsto i da je jednak nuli, uz podsećanje da u procenat lažno negativnih rezultata ne ulaze regioni koji su u prethodnim koracima pogrešno odbačeni, pa tako pomenuti procenti nisu veći od nule. Imajući u vidu sve dobijene podatke, primetno je da procenat lažno pozitivnih opada sa povećanjem granice. Kada se ovo zapažanje iskombinuje sa konstantnim nizom lažno negativnih, jasno je da donju granicu u najmanju ruku ne treba pomerati ispod pet procenata, jer bi u tom slučaju samo bio narušen kvalitet rezultata. Grafički prikaz dobijenih rezultata vidi se na slici 14.



Slika 14: Procenat lažno pozitivnih i lažno negativnih rezultata zavisno od donje granice

Na grafikonu je jasnije da procenat lažno pozitivnih opada sporije za vrednosti granice preko pet procenata, nego što procenat lažno negativnih raste. To znači da bi se povećanjem donje granice broj lažno negativnih rezultata povećao više nego što bi broj lažno negativnih opao. Osim toga, zbog prirode algoritma bi lažno negativni bili trajno odbačeni. Nasuprot tome, zadržavanje nešto većeg broja lažno pozitivnih rezultata moguće je procesom eliminacije izbaciti u sledećem koraku. Stoga je za potrebe algoritma izabrana donja granica gustine ivica od pet procenata.

Poslednji korak algoritma je eliminacija onih kandidata čiji oblik nije sličan obliku lica. Na ovaj način će iz rezultata biti uklonjeni oni delovi tela koji su iz nekog razloga prošli prethodne

provere (na primer ruka sa tetovažama ili ožiljcima). U ove svrhe koriste se Huovi invarijantni momenti.

7.6 Huovi invarijantni momenti

Moment slike predstavlja težinski prosek intenziteta piksela slike, odnosno neku njihovu funkciju. Prostorni momenat reda (p, q) definiše se na sledeći način:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

Ovaj momenat nije invarijantan u odnosu na translaciju, rotaciju i skaliranje. Stoga se uvodi centralni momenat:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy$$

gde je

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$$

Centralni momenti nisu invarijantni u odnosu na skaliranje, pa se oni normalizuju:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$

gde je $\gamma = \frac{p+q+2}{2}$.

Na osnovu normalizovanih centralnih momenata, Hu [27] je izveo sedam invarijantnih momenata:

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$

$$+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})]$$

$$\begin{aligned}\phi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned}$$

Huovi momenti su konačnog reda. Oni su invarijantni u odnosu na translaciju, skaliranje i rotaciju. Sedmi Huov momenat se koristi za razlikovanje slika dobijenih refleksijom (slika u ogledalu).

7.7 Poređenje kontura

Za eliminaciju kandidata na osnovu oblika konture potrebno je uvesti binarnu sliku sa predefinisanim konturom koja približno odgovara prosečnom licu. Sličnost između potencijalnog lica i ovakve konture se izračunava jednom od sledećih formula:

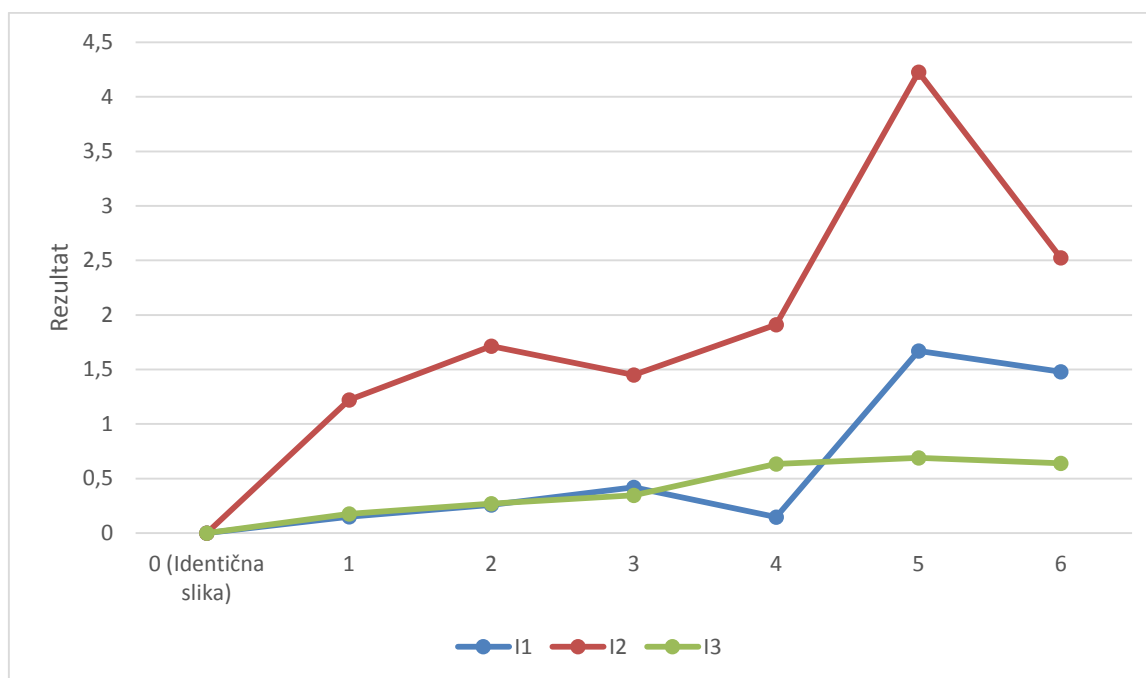
$$\begin{aligned}I_1(A, B) &= \sum_{i=1}^7 \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right| \\ I_2(A, B) &= \sum_{i=1}^7 |m_i^A - m_i^B| \\ I_3(A, B) &= \sum_{i=1}^7 \frac{|m_i^A - m_i^B|}{|m_i^A|}\end{aligned}$$

gde je

$$m_i^A = \text{sgn}(h_i^A) \log(h_i^A), m_i^B = \text{sgn}(h_i^B) \log(h_i^B)$$

za i -te Huove momente slika A i B – h_i^A i h_i^B .

Pomenute formule su preuzete iz definicije metode cvMatchShapes biblioteke Emgu CV [28]. Za potrebe upoređivanja rezultata ove tri funkcije, generisano je sedam jednostavnih slika crne pozadine sa po jednom konturom bele ispune. Huovi momenti jedne od njih upoređeni su sa momentima kontura ostalih slika (uključujući i sebe samu) pomoću sve tri funkcije. Rezultati su predstavljeni na slici 15.



Slika 15: Rezultati poređenja testnih kontura pomoću tri različite funkcije

Primetno je da za poređenje momenta konture sa momentom njene identične kopije sve tri funkcije daju vrednost nula, dok se kod poređenja različitih kontura rezultati razlikuju, ponekad i drastično. Primetno je da je na testiranim slikama druga funkcija davala veće vrednosti rezultata, ali i to da funkcije različito procenjuju sličnosti inicijalne slike sa druge dve slike. Na primer, vidljivo je da druga funkcija procenjuje da je treća slika sličnija inicijalnoj nego druga, dok se na osnovu rezultata prve i treće funkcije dolazi do upravo suprotnog zaključka. Analizom javno dostupnih programskih kodova, dolazi se do zaključka da se nijedna od tri funkcije ne favorizuje, iz čega proizilazi da svaka od njih ima svoje prednosti i mane.

Za potrebe opisanog algoritma korišćena je treća formula. Slično kao kod izbora granice za gustinu ivica, ciljevi su bili eliminacija lažno pozitivnih uz prioritetni cilj da se ne poveća broj lažno negativnih. Kompromisna granica je postavljena na 0,5 pomoću testnih slika. Kandidati koji zadovolje ovaj uslov se obeležavaju kao lica i predstavljaju rezultat celokupnog algoritma.

8 Programska implementacija

Opisani algoritam za detekciju lica je implementiran i programski za potrebe testiranja i demonstracije. Aplikacija je pisana koristeći razvojni alat Microsoft Visual Studio Ultimate, verzije 2010 i 2013. Korišćeni programski jezik je C#, uz upotrebu .NET platforme, verzije 4.0. U pitanju je desktop aplikacija koja koristi Windows forme. Korišćena je i eksterna biblioteka Emgu CV verzije 2.4.0.1717, koja je .NET *wrapper* za popularnu biblioteku za obradu digitalnih slika OpenCV, što znači da omogućava pozivanje OpenCV funkcija prilagođenih jeziku C++ iz .NET kompatibilnih jezika. Aplikacija omogućava izbor slike sa diska na kojoj će biti vršena detekcija lica kao i iscrtavanje detektovanih regiona i njihov prikaz direktno u prozoru, uz istovremeno generisanje slike za potrebe poređenja rezultata sa slikama na kojima su lica manuelno obeležena.

Emgu CV je izabran zbog specifičnih klasa i metoda čijim korišćenjem se znatno skraćuje vreme razvoja u poređenju sa vremenom potrebnim za ručnu implementaciju konverzije između različitih prostora boja, morfoloških operacija i brojnih drugih ključnih segmenata aplikacije koristeći samo izvornu biblioteku .NET klasa.

Za potrebe manipulacije ulaznim slikama, kao i međurezultatima, korišćena je klasa *Image<TColor, TDepth>*, gde je *Tcolor* prostor boja u kome se čuva slika (moguće vrednosti su *Gray, Bgr, Bgra, Hsv, Hls, Lab, Luv, Xyz i Ycc*), dok *Tdepth* predstavlja dubinu, odnosno veličinu pojedinačne komponente izabranog prostora boja (*Byte, SByte, Single, double, UInt16, Int16 or Int32*). Ulazna slika je predstavljena sa *Image<Ycc, Byte>*, imajući u vidu da je YCbCr prostor boja korišćen za detekciju kože koja je i prvi korak algoritma. Za međurezultate je korišćen *Image<Gray, Byte>*, znajući da se nakon detekcije kože koristi binarna slika na kojoj se vrše morfološke operacije a potom i detekcija kontura, pa su nijanse sive i više nego dovoljne. Za detekciju ivica je takođe potrebna slika u nijansama sive. Na kraju, za čuvanje slike za kasnije poređenje sa ručno obeleženim licima se koristi *Image<Bgr, Byte>*.

Za izvršavanje morfoloških operacija korišćene su dve metode klase *Image<TColor, TDepth>*. Za eroziju je korišćena *Image<TColor, TDepth> Erode(int iterations)*, a za dilataciju *Image<TColor, TDepth> Dilate(int iterations)*. U oba slučaja koristi se matrica veličine 3×3 kao strukturni element, gde svi pikseli osim centralnog predstavljaju njegovo susedstvo.

Parametar *int iterations* predstavlja broj iteracija izvršavanja erozije, odnosno dilatacije. U implementaciji su korišćene po dve iteracije pre detekcije kontura, odnosno po deset iteracija za određivanje regiona u kome će se vršiti procena gustine ivica. Veći broj iteracija je korišćen zbog nemogućnosti izmene strukturnog elementa, pa je na ovaj način postignut rezultat sličan onom koji bi bio dobijen da je korišćena veća matrica. Rezultat primene ove dve metode je takođe klase *Image<TColor, TDepth>*.

Nalaženje kontura je pojednostavljeno primenom metode *Contour<Point> FindContours()*. Ova metoda se primenjuje nad binarnom slikom bez parametara i rezultat je objekat klase *Contour<Point>* kojim je predstavljen niz kontura detektovanih na slici. Inicijalno, ovaj rezultat opisuje prvu pronađenu konturu, dok se ostalim konturama pristupa preko atributa *HNext* i *HPrev*, koji predstavljaju sledeću, odnosno prethodnu konturu van trenutne konture, kao i preko atributa *VNext* i *VPrev*, koji predstavljaju sledeću, odnosno prethodnu konturu unutar trenutne konture. Korišćenjem pomenutih atributa moguće je obići sve detektovane konture. Ukoliko atribut ima *null* vrednost, onda ne postoji kontura koja zadovoljava odgovarajući uslov.

Provera broja kontura unutar neke druge konture je vršena tako što je najpre formiran objekat klase *List<KeyValuePair<Contour<Point>, int>>*, koji predstavlja listu objekata koji sadrže ključ koji je kontura, i celobrojnu vrednost koja označava da li je kontura koža (vrednost 1) ili rupa (vrednost 0). Sve konture su prisutne u listi. Potom se za svaki objekat u listi čiji je ključ kontura kože proverava da li postoji objekat čiji je ključ kontura rupe koja je u potpunosti unutar te konture kože. Ova provera je vršena korišćenjem metode *double InContour(PointF point)*, koja za predatu tačku vraća negativnu vrednost ako je tačka van konture, nulu ako je na konturi, i pozitivnu vrednost ako je tačka unutar konture. U implementaciji je za sve tačke konture, kojima se pristupa preko indeksa, proveravano da li se nalaze unutar druge konture. Ako se sve tačke nalaze unutar konture, broj rupa se povećava za jedan.

Detekcija ivica Kenijevim algoritmom je takođe implementirana u klasi *Image<TColor, TDepth>*, i to u metodi *Image<TColor, TDepth> Canny(TColor thresh, TColor threshLinking)*. Parametar *thresh* predstavlja gornju granicu, a parametar *threshLinking* predstavlja donju, odnosno granicu slabih ivica koje potencijalno treba povezati sa jakim ivicama, kao što je

prethodno objašnjeno u opisu Kenijevog algoritma. U implementaciji su granice identične i postavljene na $Gray(255)$, odnosno maksimalni intenzitet, što znači da će najpre biti izabrane samo najjače ivice, dok slabih ivica neće ni biti zbog jednakosti dva parametra.

Poređenje dve konture znatno je olakšano implementacijom metode `double cvMatchShapes(IntPtr object1, IntPtr object2, CONTOURS_MATCH_TYPE method, double parameter)` u klasi `CvInvoke`. Parametri `object1` i `object2` mogu biti konture ili slike u nijansama sive. Parametar `method` predstavlja način poređenja, a moguće vrednosti su `CV_CONTOUR_MATCH_I1`, `CV_CONTOUR_MATCH_I2` i `CV_CONTOUR_MATCH_I3`. Ove tri formule za poređenje su prethodno opisane. Poslednji parameter se, prema dostupnoj dokumentaciji, trenutno ne koristi.

Aplikacija za poređenje rezultata je takođe napisana u programskom jeziku C# na .NET platformi verzije 4.0. Ulazne slike su pripremljene tako da su regioni lica pravougaonici crvene boje za rezultat algoritma, odnosno plave boje za manuelno označena lica, dok su svi ostali pikseli crne boje. Rezultati se dobijaju prebrojavanjem parova piksela sa istim koordinatama na dve slike koji su deo regiona lica na obe slike, samo na prvoj, ili samo na drugoj. Ovi podaci se prikazuju u samoj aplikaciji.

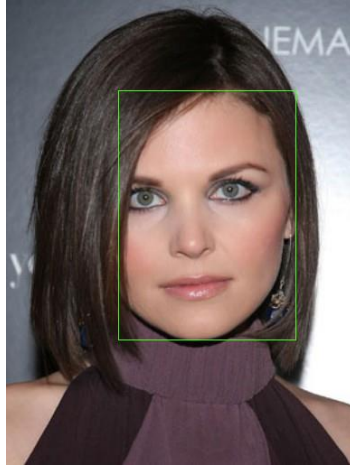
9 Testiranje i rezultati

Testiranje algoritma je vršeno na skupu slika sa jednim ili više lica prikupljenih sa veba. Zbog prirode algoritma za detekciju kože, korišćene su isključivo slike u boji, imajući u vidu da slike u nijansama sive ne sadrže potrebne informacije za inicijalno klasifikovanje piksela. Pored toga, vođeno je računa i da generalni kvalitet slika bude dobar, što podrazumeva da slike nisu bile u premalim rezolucijama, kao i da osvetljenje nije bilo takvo da značajno utiče na prirodnu boju kože.



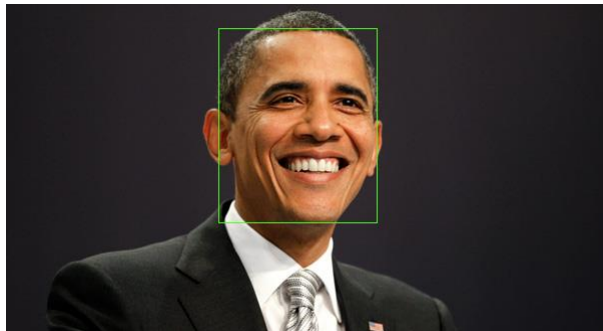
Slika 16: Konačni rezultat algoritma

Na slici 16 se vidi konačni rezultat algoritma na primeru korišćenom za opisivanje svih međurezultata. Kontura koja je proglašena za lice obeležena je na originalnoj slici opisanim pravougaonikom čije su ivice paralelne sa ivicama slike. Algoritam je dao jako dobar rezultat u ovom slučaju, jer je lice jasno uokvireno i većina posmatrača bi i sama ograničila lice na ovoj slici pravougaonikom gotovo identične pozicije i dimenzija. Region vrata nije detektovan ni kao poseban ni kao deo prikazanog rezultata. Razlog je najpre senka na liniji brade koja je bila dovoljno tamna da se ne uklopi u izabrani model kože i na taj način razdvoji dva regiona. Pored toga, naredni koraci algoritma su obezbedili da ovaj region bude odbačen. Osim njega, odbačeni su i manji region koji nisu predstavljali kožu već preostali šum, uglavnom na ivicama kose.



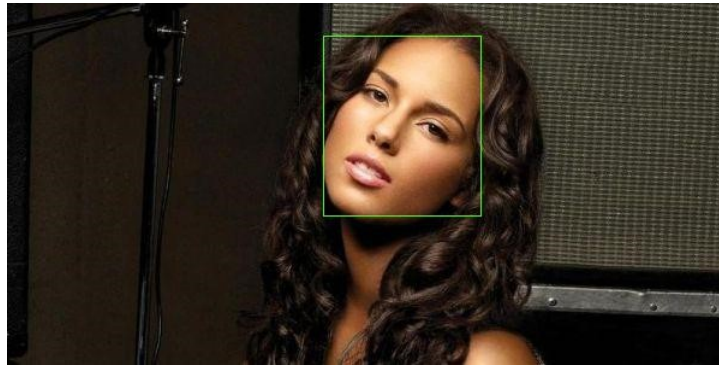
Slika 17: Rezultat algoritma na delimično zaklonjenom licu

Na slici 17 vidi se još jedan primer rezultata algoritma, ovoga puta na licu koje je jasno razdvojeno od ostalih regiona kože. Iako je vidljivi deo kože lica jasno uokviren, posmatraču je jasno da jedan deo lica nije vidljiv i da je prekriven kosom, konkretno desna strana lica. Ovaj deo lica može se smatrati lažno negativnim rezultatom, ali je u ovom slučaju detektovan dovoljno veliki deo lica da se rezultat može smatrati vizuelno dosta dobrim. Takođe, primetne su i druge površine kože u području ramena koje su uspešno odbačene.



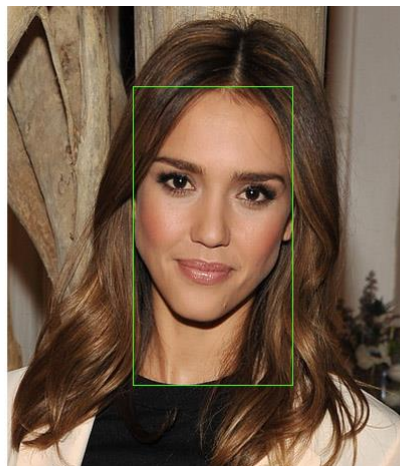
Slika 18: Rezultat detekcije lica tamnije kože

Na slici 18 je rezultat algoritma na primeru lica tamnije kože. Pored toga lice nije ni potpuno *en face*. Rezultat je dosta dobar i ignoriše deo vrata i ušiju. Deo desnog uveta je takođe unutar finalnog pravougaonika, pa se to može smatrati lažno pozitivnim rezultatom. Sve u svemu, rezultat je definitivno vizuelno jako dobar.



Slika 19: Detekcija tamnije kože na blago zarotiranom licu

Slika 19 predstavlja rezultat na primeru lica takođe tamnije kože ali blago zarotiranog. Još jednom je lice pravilno uokvireno, odbacujući druga manja područja kože. Rezultat je dobar jer je lice uokvireno tamnom kosom i tamnom senkom brade koje ga razdvaja od ostatka kože.



Slika 20: Rezultujući region koji uključuje lažno pozitivne piksele

Primer slike 20 pokazuje rezultujući region koji sadrži i očigledno pogrešno izabrane piksele vidljivog dela vrata. Analizom međurezultata algoritma, utvrđeno je da senka brade nije bila dovoljna da razdvoji ova dva regiona čak ni nakon primene morfoloških operacija otvaranja i zatvaranja, a dopunsko zatvaranje većeg intenziteta je obezbedilo da kompletan region ne bude odbačen u kasnijim koracima. I pored ovih lažno pozitivnih piksela, jasno je lažno negativnih piksela gotovo i da nema, pa su dobre strane algoritma nadmašile one negativne, pa je konačni rezultat zadovoljavajuć, iako ne idealan.



Slika 21: Detekcija na slici sa više lica

Na slici 21 vidimo dva dobro osvetljena lica u prvom planu, kao i tri lica u pozadini koja su znatno slabije osvetljena i od kojih je samo jedno *en face*. Nijedno od ova tri lica nije detektovano, a analizom međurezultata utvrđeno je da se zbog loše osvetljenosti njihova koža nije uklopila u model boje kože, pa su tako eliminisana već u prvom koraku. Čak i da su ova lica bila bolje osvetljena, bilo bi teško očekivati da sva budu detektovana, imajući u vidu da je jedno od njih prikazano iz profila, što znači da je prisustvo karakterističnih crta lica umanjeno, a da je jedno okrenuto još dalje od posmatrača tako da se na njemu uopšte ne vide oči a ni ostale crte koje utiču na prisustvo ivica i koje su ključne za rad algoritma. Što se tiče dva lica u prvom planu, vidljivo je da su oba detektovana i obeležena, ali uz prisustvo određenog broja piksela koji iako predstavljaju kožu, nisu deo lica, a razlog je isti kao u primeru sa slike 20. Ipak, ostali odvojeni regioni kože koji su u prvom planu su ispravno odbačeni. U ovom slučaju je teško oceniti kvalitet rezultata. Iako su oba lica u prvom planu uokvirena, doduše nešto većim pravougaonicima od idealnih, lica u pozadini su predmet za diskusiju. Iako će svaki posmatrač lako uočiti tri osobe u pozadini, jedno od njih je gotovo potpuno sakriveno, tako da će ocena rezultata zavisi od subjektivnog utiska posmatrača i njegove procene očekivanog rezultata.



Slika 22: Detekcija na slici sa velikim brojem lica

Na slici 22 prikazan je rezultat detekcije velikog broja lica. Iako su sva prisutna lica detektovana, kvalitet varira. Neka od lica su u prvom koraku algoritma spojena sa drugim regionima prekrivenim kožom, uglavnom rukama, što rezultuje većim površinama rezultujućih pravougaonika nego u idealnom slučaju. Ipak, većina lica su celom svojom površinom unutar uokvirenih rezultata, s izuzetkom lica drugog s desna u donjem redu. Analizom međurezultata je utvrđeno da je zbog senke i brade na licu došlo do razdvajanja na dva regiona, odnosno pojave lažno negativnih piksela. Lažno pozitivni pikseli su prisutni jer nisu narušili uslove za uspešan prolazak kroz sledeće korake algoritma. Primetno je da nijedan od jasno ograničenih regiona kože koji nisu lica nije uokviren, što znači da su uspešno eliminisani u kasnijim koracima. Može se smatrati da su rezultati zadovoljavajući, iako rezultujući pravougaonici nisu optimalni. Posmatraču će svakako biti jasno da se u pozadini takođe nalazi veliki broj lica, ali su ona pored lošije osvetljenosti i van fokusa. Pored toga, nijedan posmatrač ne bi mogao da sa sigurnošću označi takva lica, a možda ne bi ni mogao ni da uoči da je uopšte reč o licima da nije ostalih elemenata slike na osnovu kojih se dolazi do zaključka da je u pitanju publika na fudbalskoj utakmici. S obzirom da bi ovo bio težak zadatak i za čoveka, teško je očekivati da računarski algoritam da bolje rezultate, pa se tako ovo sa razlogom ne može smatrati nedostatkom.

Umesto prostog prebrojavanja detektovanih lica i upoređivanja tog broja sa stvarnim brojem lica na slici, do procene efikasnosti algoritma došlo se na drugačiji način. Naime, za svaku

sliku korišćenu u testiranju, napravljena je odgovarajuća slika na kojoj je region koji predstavlja lice obeležen ručno. Radi jednostavnijeg postupka testiranja i ubrzanja procedure, region lica je označen pravougaonikom čije su ivice paralelne sa ivicama slike. Potom je ova slika upoređena sa slikom koja je rezultat algoritma, a na kojoj su lica takođe obeležena pravougaonicima. Rezultat poređenja su četiri vrednosti. Prva je broj pozitivnih rezultata, odnosno piksela koji su označeni kao deo regiona lica na obe slike. Broj lažno negativnih rezultata je broj piksela koji su na pripremljenoj slici označeni kao lice, ali ne i na rezultatu algoritma. Nasuprot tome, broj piksela koji na pripremljenoj slici nisu označeni kao lice ali jesu na rezultatu algoritma, predstavljaju lažno pozitivne rezultate. Na kraju se računa ukupan broj piksela koji su označeni kao lice bar na jednoj od dve slike. Step en uspešnosti algoritma računa se na sledeći način:

$$\text{Pozitivni} = \frac{\text{Broj pozitivnih}}{\text{Ukupan broj}} \cdot 100$$

$$\text{Lažno negativni} = \frac{\text{Broj lažno negativnih}}{\text{Ukupan broj}} \cdot 100$$

$$\text{Lažno pozitivni} = \frac{\text{Broj lažno pozitivnih}}{\text{Ukupan broj}} \cdot 100$$

Dobijeni brojevi su procenti odgovarajuće klasifikovanih piksela. Testiranje je dalo sledeće rezultate:

Pozitivni	Lažno negativni	Lažno pozitivni
85,14%	3,53%	11,33%

Procenat lažno negativnih piksela je relativno mali, ali je broj lažno pozitivnih nešto veći. Nakon pregleda rezultata, utvrđeno je da većina lažno pozitivnih piksela potiče sa slika na kojima je neposredno pored lica bio neki drugi region kože, na primer ruka. Do ove nepreciznosti dolazi jer algoritam ne detektuje individualne crte lica, na osnovu kojih bi se moglo proceniti koji deo prepoznatog regiona zapravo predstavlja lice. Takođe je bitno napomenuti i da je manuelno označavanje lica za potrebe ovog testiranja potpuno subjektivno, tako da bi i rezultati mogli da variraju zavisno od osobe koja je za to zadužena.

Dobijeni rezultati su upoređeni sa rezultatima algoritama opisanih u [14] i [16]. Pomenuti algoritmi su izabrani jer dele neke ključne osobine sa algoritmom predstavljenim u ovom radu, o čemu će više reči biti u nastavku teksta.

Algoritam opisan u [14] takođe se zasniva na modeliranju boje kože, s tim što se region kože formira kao unija regiona detektovanih u YCbCr prostoru, ali i u druga dva prostora koje su autori ocenili kao pogodne. Konačni rezultati su pokazali da je korišćenjem više prostora boja postignuto da je udeo lažno pozitivnih i lažno negativnih znatno manji nego kada je samostalno korišćen bilo koji od tri izabrana prostora. Naime, udeo lažno pozitivnih rezultata je 3,09%, dok je udeo lažno negativnih svega 1,73%. Na prvi pogled deluje da su dobijeni rezultati bolji od onih prikazanih u ovom radu. Ipak, ako znamo da se za lažno negativne rezultate proglašavaju čitava lica koja nisu detektovana, a za lažno pozitivne celi detektovani regioni koji nisu lica, jasno je da se tako procenjena uspešnost algoritma ne može direktno porediti sa ovde prikazanim rezultatima testiranja koji su na nivou pojedinačnog piksela. Naravno, uticaj na rezultate može imati i izbor slika na kojima je algoritam testiran.

Još jedan algoritam koji koristi YCbCr prostor boja opisan je u [16]. Osim za detekciju kože, autori su ovaj prostor iskoristili i za detekciju očiju i usta. Testiranje je vršeno na više grupa slika, od kojih su neke sadržale lica u specifičnim pozama. Procenat detektovanih lica se kreće od 18,18% za lica iz profila, do 91,63% za *en face* fotografije niže rezolucije. Slično kao u prethodno pomenutom algoritmu, definicija ispravno detektovanih lica se značajno razlikuje od one koju ovaj rad predlaže, pa je svako direktno poređenje nezahvalno.

Za poređenje rezultata dvaju ili više algoritama za detekciju lica, idealno je utvrditi jedinstvenu meru kvaliteta i vršiti testiranje na istom uzorku. Naravno, uzorak je potrebno formirati tako da ne daje prednost nijednom algoritmu. Ipak, ukoliko su slike na kojima je vršeno testiranje valjano izabrane, dobijeni rezultati mogu biti okvirna mera kvaliteta, a eventualno poslužiti i za grubo poređenje.

10 Zaključak

Osnovna pretpostavka je bila da detekcija kože na slici može biti ključni deo algoritma za detekciju lica. Ova ideja je unapređena detekcijom ivica kao vrstom dodatne validacije u pojedinim regionima, imajući u vidu da u većini slučajeva region lica ima daleko više ivica nego bilo koji drugi region kože, i tu se Kenijev detektor ivica pokazao kao dobar metod. Na kraju, Huovi momenti su poslužili u finalnoj validaciji regiona na osnovu njegovog oblika.

Očigledna mana predstavljenog algoritma je nemogućnost rada sa slikama u nijansama sive ili slikama na kojima su informacije o boji značajno izmenjene, imajući u vidu prirodu izabranog algoritma za detekciju kože. Ovo znači da slike moraju u startu biti dobrog kvaliteta, jer slike napravljene uređajima sa senzorom slabijeg kvaliteta (kao što su stariji mobilni telefoni sa kamerom) često imaju dosta šuma i ne mogu da verno reprodukuju boje. Osvetljenost takođe mora biti dobra, inače postoji opasnost da regioni kože ne budu prepoznati korišćenjem postojećeg modela. Ne treba ni trošiti reči na to da će primena pojedinih filtera nad slikom rezultovati drastično lošijim rezultatima. Druga značajna mana je nemogućnost razdvajanja spojenih regiona kože, što rezultuje većim brojem lažno pozitivnih rezultata. Ovaj nedostatak se može otkloniti detekcijom crta lica, kao što su oči, nos i usta, i formiranjem regiona lica na osnovu njihovih relativnih položaja. Granica određena kod poređenja kontura pomoću Huovih momenata je možda previše konzervativna. Jedno od mogućih rešenja je korišćenje većeg broja predefinisanih kontura lica, imajući u vidu da različite frizure, položaji lica i brojni drugi faktori u značajnoj meri utiču na oblik konture vidljivog dela kože lica. Na taj način granica bi mogla biti i manja ukoliko se za poređenje sa njom koristi minimalni rezultat poređenja konture lica sa svakom od predefinisanih kontura.

Iako predstavljeni algoritam poseduje izvesna ograničenja, pre svega u vidu potrebe za kvalitetnim slikama u boji, testiranja su pokazala da može biti dobra polazna tačka za nadogradnju korišćenjem naprednijih tehnika koje bi proizvele bolje rezultate i proširile područje primene.

11 Lista slika

Slika 1: Osnovni koncept optičkog sistema bez barijere (levo) i sa barijerom (desno)	10
Slika 2: Rendgenski snimak ramena	11
Slika 3: Slika dobijena pomoću ultrazvuka	11
Slika 4: Slika preneta ranim Bartlejn sistemom.....	12
Slika 5: Mešanje R, G i B komponenti.....	16
Slika 6: Mešanje C, M i Y komponenti	17
Slika 7: Grafički prikaz CIELAB modela boja	18
Slika 8: HSV cilindar	19
Slika 9: Paleta za izbor boja u programu Adobe Photoshop	20
Slika 10: Slika u boji i njene Y, Cb i Cr komponente	28
Slika 11: Originalna slika (levo) i regioni detektovani kao koža (desno)	31
Slika 12: Regioni kože nakon primene morfoloških operacija	33
Slika 13: Region lica nakon izvršene operacije zatvaranja (levo) i detektovane ivice unutar tog regiona (desno)	35
Slika 14: Procenat lažno pozitivnih i lažno negativnih rezultata zavisno od donje granice	38
Slika 15: Rezultati poređenja testnih kontura pomoću tri različite funkcije.....	41
Slika 16: Konačni rezultat algoritma.....	45
Slika 17: Rezultat algoritma na delimično zaklonjenom licu.....	46
Slika 18: Rezultat detekcije lica tamnije kože	46
Slika 19: Detekcija tamnije kože na blago zarotiranom licu.....	47
Slika 20: Rezultujući region koji uključuje lažno pozitivne piksele.....	47
Slika 21: Detekcija na slici sa više lica.....	48
Slika 22: Detekcija na slici sa velikim brojem lica	49

12 Lista referenci

- [1] M. D. McFarlane, "Digital pictures fifty years ago," *Proceedings of the IEEE*, vol. 60, no. 7, pp. 768-770, 1972.
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*.: Prentice Hall, 2008.
- [3] PANTONE MATCHING SYSTEM[®]. [Online].
<http://www.pantone.com/pages/pantone/pantone.aspx?pg=19970>
- [4] "CIE. (1932)," in *Commission Internationale de l'Éclairage Proceedings*. Cambridge: Cambridge University Press, 1931.
- [5] W. D. Wright, "A re-determination of the trichromatic coefficients of the spectral colours," in *Transactions of the Optical Society*, 1928, pp. 141-164.
- [6] J. Guild, "The colorimetric properties of the spectrum," in *Philosophical Transactions of the Royal Society of London*, A230, 1931, pp. 149-187.
- [7] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58, 2002.
- [8] G. Yang and T. S. Huang, "Human Face Detection in Complex Background," *Pattern Recognition*, vol. 27, no. 1, pp. 53-63, January 1994.
- [9] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A Survey of Skin-Color Modeling and Detection Methods," *Pattern Recognition*, vol. 40, no. 3, pp. 1106-1122, March 2007.
- [10] V. Vezhnvets, V. Sazonov, and A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques," in *Proceedings of GraphiCon*, Moscow, Russia, September 2003, pp. 85-92.
- [11] R. Kjellden and J. Kender, "Finding Skin in Color Images," in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 312-317.
- [12] G. Kukharev and A. Novosielski, "Visitor identification - elaborating real time face recognition system," in *Proceedings of 12th Winter School on Computer Graphics (WSCG)*, Plzen, Czech Republic, February 2004, pp. 157-164.
- [13] J. L. Crowley and J. Coutaz, "Vision for Man Machine Interaction," *Robotics and Autonomous Systems*, vol. 19, no. 3-4, pp. 347-358, March 1997.
- [14] S. K. Singh, D. S. Chauhan, M. Vatsa, and R. Singh, "A Robust Skin Color Based Face Detection Algorithm," *Tamkang Journal of Science and Engineering*, vol. 6, no. 4, pp. 227-234, December 2003.
- [15] T. M. Mahmoud, "A New Fast Skin Color Detection Technique," *World Academy of Science, Engineering and Technology*, vol. 19, pp. 501-505, July 2008.
- [16] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face Detection in Color Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696-706, May 2002.
- [17] J. C. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamtsu, "Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic Detection of Human Faces in Color Images," in *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 54-61.

- [18] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551-564, June 1999.
- [19] J. A. M. Basilio, G. A. Torres, G. S. Pérez, L. K. T. Medina, and H. M. P. Meana, "Explicit image detection using YCbCr space color model as skin detection," in *Proceedings of the 2011 American conference on applied mathematics*, 2011, pp. 123-128.
- [20] M. B. Stankovich and M. Tuba, "Using Edge Density and Contour Matching in Skin Color Based Face Detection," *International Journal of Mathematical Models and Methods in Applied Sciences*, Accepted for publication.
- [21] J. Serra, *Image Analysis and Mathematical Morphology*.: Academic Press, Inc., 1983.
- [22] E. Nadernejad, S. Sharifzadeh, and H. Hassanpour, "Edge Detection Techniques: Evaluations and Comparisons," *Applied Mathematical Sciences*, vol. 2, no. 32, pp. 1507-1520, 2008.
- [23] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, November 1986.
- [24] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *The International Journal of Computer Vision*, vol. 1, no. 2, pp. 167-187, May 1987.
- [25] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 1485-1490, September 2005.
- [26] S. L. Phung and A. Bouzerdoum, "Detecting People in Images: An Edge Density Approach," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Honolulu, 2007, pp. I-1229-I-1232.
- [27] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179-187, February 1962.
- [28] cvMatchShapes Method (object1, object2, method, parameter). [Online]. <http://www.emgu.com/wiki/files/1.4.0.0/html/5a26fa7f-3939-9dbf-1611-039c2f90ba2b.htm>