

Projekt iz predmeta Raspoznavanje uzoraka

Ak. god. 2015/16

**Detekcija lica u video zapisima
uporabom fuzije postupaka Viola-Jones
i informacije o dubini scene uporabom
stereo vida**

Autori:

Bojan Lovrović

Tomislav Huljina

Antun Maldini

Bruno Pregun

Mihael Presečan

Jure Ratković

Siječanj, 2016.

Sadržaj:

1. Projektni zadatak.....	1
1.1 OPIS PROJEKTOG ZADATAKA.....	1
1.2 PREGLED I OPIS SRODNIH RJEŠENJA.....	1
1.3 KONCEPTUALNO RJEŠENJE ZADATAKA.....	2
2. Postupak rješavanja zadatka.....	2
2.1 SNIMANJE I PRIPREMA RGB-D VIDEOZAPISA.....	2
2.2 DETEKCIJA LICA VIOLA-JONES POSTUPKOM.....	3
2.2.1 Viola-Jones postupak.....	3
2.3 PROVJERA DETEKTIRANIH LICA SVM KLASIFIKATOROM.....	6
2.3.1 Stroj potpornih vektora.....	6
2.4 OZNAČAVANJE LICA I KREIRANJE IZLAZNOG VIDEOZAPISA.....	8
3. Ispitivanje rješenja.....	8
3.1 ISPITNA BAZA.....	8
3.2 REZULTATI UČENJA I ISPITIVANJA, ANALIZA REZULTATA.....	10
4. Opis programske implementacije rješenja.....	10
5. Zaključak.....	11
6. Literatura.....	12

1. Projektni zadatak

1.1 Opis projektnog zadatka

Zadatak je detekcija lica u videozapisu. Ulazi u programsko rješenje su jedan RGB videozapis (snimka scene u vidljivom spektru) i jedan grayscale videozapis (dubinska snimka scene), koji pokrivaju identično vidno polje, bez paralakse. Informacija o dubini scene se, unatoč naslovu, ne dobiva stereo vidom – jer bio to bio zadatak već sam po sebi – već direktno iz odgovarajuće kamere. Za snimanje videozapisâ se koristi Kinect uređaj, koji sadrži i standardnu RGB kameru i depth senzor; druga RGB-D kamera bi isto poslužila. Izlaz je jedan RGB videozapis sa označenim detektiranim licima.

Na RGB slici se vrši detekcija lica Viola-Jones postupkom, te se pozicije na kojima su detektirana lica dodatno provjeravaju na dubinskoj slici SVM klasifikatorom.

1.2 Pregled i opis srodnih rješenja

Za treniranje i ispitivanje je korištena mala, vlastita baza od desetak kratkih RGB-D videozapisa. Do standardnih baza RGB-D slika/videozapisâ na koje smo nailazili nismo mogli doći, jer nisu dostupne studentima.

Neka od srodnih rješenja:

Robust real-time face detection, 2004. (autori Viola, Jones)

Osnovni Viola-Jones postupak, sa nekoliko bitnih novosti (za svoje vrijeme): postupak ne radi direktno na slici, već na njenoj reprezentaciji (integralna slika) koja omogućuje iznimno brzo računanje značajki; koriste se jednostavni klasifikatori trenirani AdaBoost algoritmom koji izdvaja mali broj bitnih značajki iz velikog skupa istih; umjesto monolitnog klasifikatora, koristi se kaskada progresivno složenijih klasifikatora. Rezultat je ubrzanje od reda veličine u odnosu na najbolje dotadašnje postupke, i prvi sustav za detekciju lica u stvarnom vremenu.

Improving face detection with depth, 2016. (autori Meyer, Alfano, Do)

Proširenje Viola-Jones postupka sa predobradom na temelju dubinske informacije. Kao i u našem rješenju, koriste se RGB-D slike dobivene pomoću Kinect uređaja; na ulazu i izlazu sustava je slika, ne videozapis. Postupak je sljedeći: za svaki piksel, na temelju dubinske slike računa se očekivana veličina lica na toj udaljenosti; na temelju tih dimenzija, određuje se susjedstvo piksela koje će se ispitati; susjedstvo se na dubinskoj slici ispituje i na temelju njega se određuje može li piksel biti dio lica. Rezultat prvog dijela su grupe piksela za koje je određeno da mogu biti dijelovi lica, a sačinjavaju manji dio slike. Na temelju tih grupa se određuju mogući prozori u kojima mogu biti lica, i nad njima se pokreće Viola-Jones postupak. Rezultat je znatno povećana preciznost i oko 3 puta veća brzina u odnosu na osnovni Viola-Jones postupak.

1.3 Konceptualno rješenje zadatka

Ulaz, kao što je ranije spomenuto, se sastoji od RGB i *depth* ulaza.

Svaki od tih videozapisa se prvo rastavi u individualne *frameove*; kako oba moraju biti istog trajanja i istog *frameratea*, rezultat će biti N RGB frameova i N grayscale frameova, odnosno N parova.

Za svaki od tih N parova, provedemo postupak koji za izlaz ima jedan, obrađeni RGB *frame*. Novonastalih N obrađenih RGB *frameova* zalijepimo nazad u videozapis identičnog trajanja i *frameratea* kao i originalni videozapisi.

Postupak za svaki par je sljedeći: na RGB *frameu* se provede detekcija lica Viola-Jones postupkom; rezultat detekcije je pravokutnik koji na slici omeđuje detektirano lice (*bounding box*). Taj dio uparene dubinske slike se daje kao ulaz SVM klasifikatoru, koji odlučuje je li zaista riječ o licu ili nije. Na taj način se eliminiraju lažno pozitivno klasificirani primjeri (*false positives*). SVM klasifikator je prethodno natreniran na reprezentativnom skupu lica i ne-lica. Na kraju, na RGB *frameu* se označi pravokutnik koji je rezultat detekcije Viola-Jones postupkom, čija boja ovisi o tome je li i SVM klasifikator također označio taj segment kao lice. Naravno, u slučaju više lica, imat ćemo i više pravokutnika.

2. Postupak rješavanja zadatka

1. Snimanje RGB-D videozapisa i treniranje SVM klasifikatora
2. Za svaki RGB frame, provesti Viola-Jones postupak za detekciju lica
3. Za svako detektirano lice, SVM klasifikatorom provjeriti odgovarajući dio dubinske slike
4. Označiti detektirana lica i kreirati izlazni videozapis

2.1 Snimanje i priprema RGB-D videozapisa

Prvi korak je nabava i predobrada materijala koji služi kao ulaz u sustav. Sustav na ulazu očekuje dva videozapisa jednake rezolucije, duljine, i *frameratea*. Prvi je običan RGB videozapis vidljivog spektra. Drugi je videozapis dubine scene – idealno *grayscale*, ali ovdje kodiran u R i G kanale, s manje značajnim bitovima u G kanalu. Samo snimanje videozapisa radimo pomoću Kinect kamere koja za izlaz daje RGB i dubinske slike u VGA rezoluciji, koje zatim lijepimo u videozapis. U općem slučaju, bilo bi potrebno napraviti registraciju (skaliranje i preklapanje) videozapisa prije postavljanja istih za ulaz sustava.

U našem rješenju, zahtjeve rezolucije, duljine i *frameratea* rješavamo prilikom samog snimanja Kinectom (snimanjem jednakog broja slika, jednake rezolucije, i lijepljenjem s jednakim *framerateom*), dok paralaksu eliminiramo *hardcoded* ispravkom koordinata omeđujućeg pravokutnika.

Također, potrebno je istrenirati SVM klasifikator; za to nam je potreban *dataset* sa i pozitivnim primjerima (dubinskim mapama lica) i negativnim primjerima (dubinskim mapama ne-lica). Te primjere također generiramo iz videozapisa, pomoću djelomično funkcionalnog sustava, na sljedeći način: na frameovima RGB videozapisa se obavlja detekcija lica Viola-Jones postupkom, sa omeđujućim pravokutnicima (*bounding boxes*) kao rezultatom. Dio pripadne dubinske slike omeđen tim pravokutnicima se izrezuje i zapisuje kao

zasebna slika, rezultirajući dubinskim mapama svih objekata koje Viola-Jones postupak na temelju RGB informacije smatra licima, iz svih frameova videozapisa. Za svaku takvu dubinsku mapu "lica" se, ovisno o željenom omjeru pozitivnih i negativnih primjera, izrezuje i odgovarajući broj dijelova dubinske slike koji se ne preklapaju sa dijelovima na kojima je detektirano lice. Tako dobiveni isječci dubinskih slika se ručno prekontroliraju kako bi se uklonili lažno pozitivni i lažno negativni primjeri Viola-Jones algoritma, odnosno da bi SVM klasifikator trenirali na dobrom skupu za učenje. Ovaj zadnji dio je ključan preduvjet da bi kompletan sustav u konačnici radio bolje nego detekcija samo Viola-Jones postupkom.

Viola-Jones postupak radi na statičnoj 2D slici, i ne može iskoristiti dodatnu informaciju koju videozapis nudi. Kako naše rješenje inkorporira Viola-Jones postupak, tako videozapis promatra isključivo kao niz statičnih slika, od kojih se svaka obrađuje zasebno. Ulazni videozapis stoga treba prije same detekcije rastaviti na pojedinačne RGB-D slike (odnosno, parove RGB i dubinske slike). Kako u konkretnoj implementaciji snimku iz Kinecta dobivamo kao niz slika koje kasnije lijepimo u videozapis, koristimo direktno te slike umjesto da rastavljamo videozapise nazad na slike.

2.2 Detekcija lica Viola-Jones postupkom

Na svakom RGB frameu provodimo detekciju lica Viola-Jones postupkom, pomoću gotove funkcije ugrađene u Matlab *CV toolbox*. Rezultat je niz omeđujućih pravokutnika, koji omeđuju segmente slike u kojima je detektirano lice.

2.2.1 Viola-Jones postupak

Viola-Jones postupak detekcije lica je vrlo popularan i brz postupak. Radi na običnim 2D grayscale slikama, odnosno ne koristi informaciju o boji, već radi samo sa intenzitetom. Postupak je u svojoj osnovi sljedeći:

1. Veličina prozora se postavi na početnu veličinu (veličina najmanjeg lica koje se još uvijek može detektirati; osnovna rezolucija detektora)
2. Prozor se pomiče preko cijele slike; u svakom koraku se sadržaj prozora klasificira kao lice ili ne-lice
3. Prozor se skalira za konstantni faktor (obično 1.25), i postupak se ponavlja
4. Kad veličina prozora više ne može rasti (odnosno kad premaši veličinu slike), detekcija je gotova
5. Segmente koji su klasificirani kao lica, a preklapaju se, se računa kao jedno lice; omeđujući pravokutnik se računa kao prosjek omeđujućih pravokutnika tih segmenata

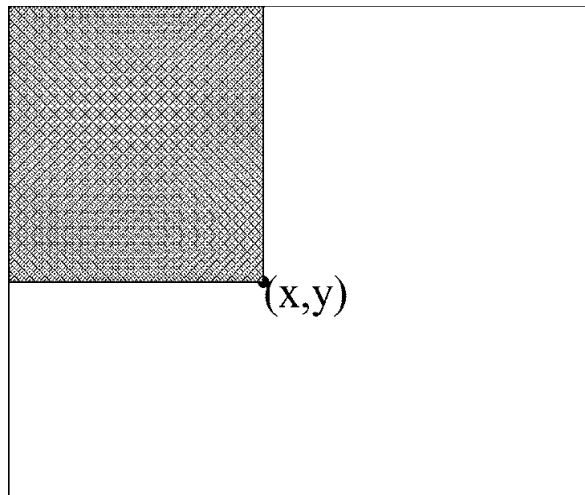
Veliku brzinu postiže pomoću nekoliko stvari:

1. integralne slike i *Haar-like* značajke
2. korištenje AdaBoost algoritma za treniranje pojedinih klasifikatora
3. kaskada progresivno složenijih klasifikatora

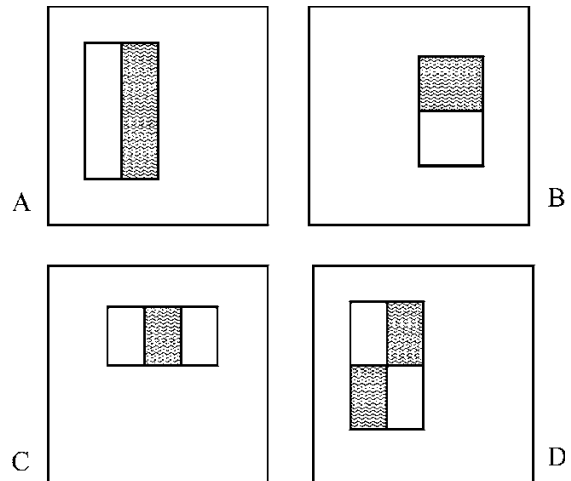
Integralna slika je način reprezentacije originalne slike koji omogućava brzo računanje *Haar-like* značajki. Za sliku dimenzija $X \times Y$, element slike (x,y) će biti jednak sumi svih piksela

(uključivo) iznad i lijevo od njega, odnosno $ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$ gdje je $ii(x, y)$ element integralne slike, a $i(x, y)$ element originalne slike.

Haar-like značajke su tako nazvane jer podsjećaju na Haarove bazne funkcije. Takva značajka se sastoji od određenog broja pravokutnika, podijeljenih u dvije grupe. Vrijednost značajke se računa kao razlika sume piksela unutar prve grupe pravokutnika i sume piksela unutar druge grupe. Ako uzmemo prvu grupu kao bijelu, a drugu kao crnu (vidi sliku), vrijednost značajke je razlika integrala bijelog i crnog područja.

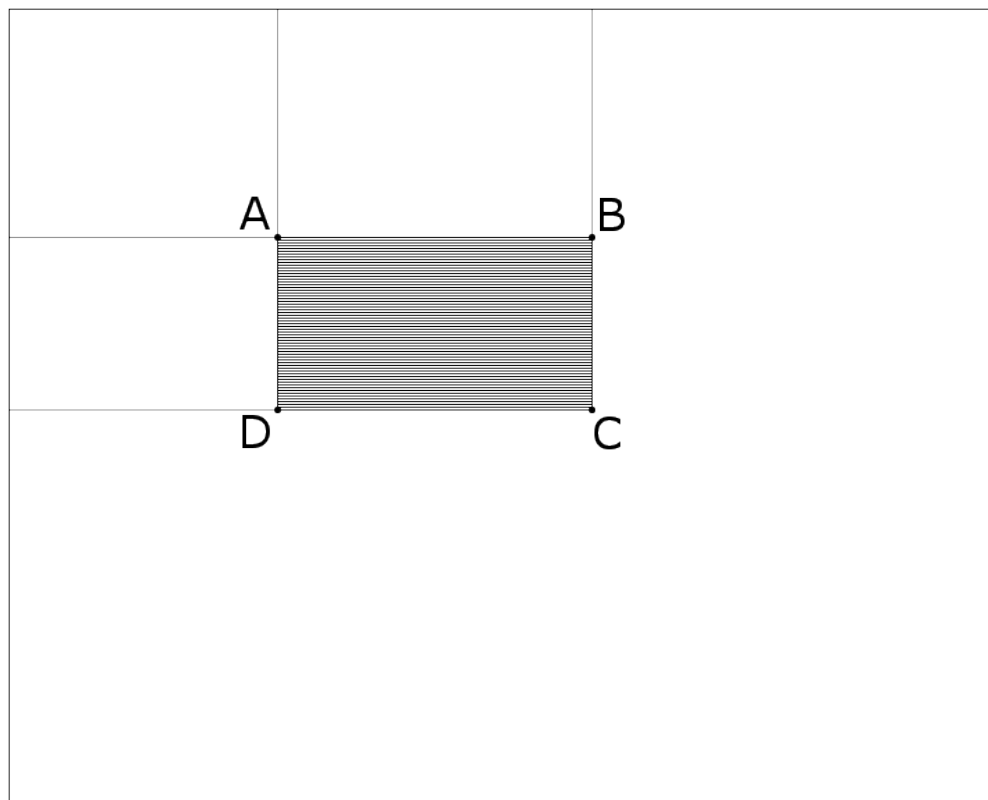


Slika 1: Integralna slika



Slika 2: Haar-like značajke

Integralna slika omogućuje vrlo brzo računanje tih integrala (odnosno, s obzirom da je riječ o diskretnom prostoru, sumâ). Za pravokutnik ABCD kao na slici, vrijedit će $P = I(C) + I(A) - I(B) - I(D)$ gdje je P integral pravokutnika, a $I(X)$ vrijednost integralne slike u točki X ; integral pravokutnika se može izračunati samo iz vrijednosti integralne slike u njegovim vrhovima. Kako se integralna slika može izračunati u jednom prolazu kroz sliku ($O(X \cdot Y)$), računanje značajki je iznimno brzo, i ne ovisi o skaliranju značajke.



Slike 3: Računanje integrala

AdaBoost algoritam (skraćeno od *Adaptive Boosting*) je algoritam za treniranje boosted klasifikatora. *Boosted* klasifikator je linearna kombinacija izlaza K slabih klasifikatora; svaki od njih sam za sebe ne mora biti dobar – dovoljno je da klasificira primjere bolje nego nasumično pogađanje, da bi *boosted* klasifikator mogao dobro klasificirati primjere. U ovom postupku, slabi klasifikatori su ograničeni na računanje vrijednosti jedne značajke i usporedbu s nekim pragom. To znači da efektivno AdaBoost algoritam odabire odgovarajuće značajke, i to relativno mali broj značajki iz ogromnog skupa potencijalnih značajki.

S ovakvim slabim klasifikatorom, *boosted* klasifikator je ustvari dvoslojni perceptron.

Kaskadiranje klasifikatora nije potpuno nova ideja. Motivacija za ovo je činjenica da su skoro svi segmenti slike ne-lica. Klasifikator, da bi ocijenio radi li se o licu ili ne, mora biti poprilično složen za postizanje dobrih rezultata. Kako svaka slika zahtijeva klasifikaciju velikog broja segmenata, to rezultira dugim trajanjem detekcija. Rješenje za taj problem je kaskadiranje većeg broja klasifikatora, koji djeluju kao filtri.

Prva razina kaskade se tako sastoji od vrlo jednostavnog klasifikatora od dvije značajke, koji ima vrlo velik odziv/osjetljivost (*recall*) i relativno velik *false positive rate*; filtrira većinu ne-lica, a propušta dalje skoro sva lica. Svaka sljedeća razina kaskade je klasifikator složeniji od prethodnog, koji zahtijeva više vremena za izračun. Ako segment uspije proći kroz sve razine kaskade klasificiran kao pozitivan, proglašava ga se licem. Ako neka od razina kaskade odbaci primjer kao negativan, segment se odbacuje i ne prosljeđuje se sljedećoj razini kaskade.

Kako većina segmenata slike ne sadrži lice, i veliku većinu njih se može odbaciti već u prvim par razina, rezultat je drastično ubrzanje postupka, jer samo mali dio segmenata zasluži detaljnije promatranje.

Za željenu osjetljivost (*detection rate*) D i *false positive rate* F ukupne kaskade, vrijedi $D = \prod d_i$ gdje je d_i osjetljivost i -tog klasifikatora a K broj klasifikatora (razina kaskade), te $F = \prod f_i$ gdje je f_i *false positive rate* (FPR) i -tog klasifikatora. Ako želimo, naprimjer, osjetljivost od $D=0.9$ i FPR od $F=10^{-6}$, to možemo postići kaskadom od 10 klasifikatora sa osjetljivošću 0.99 i FPR-om 0.25.

2.3 Provjera detektiranih lica SVM klasifikatorom

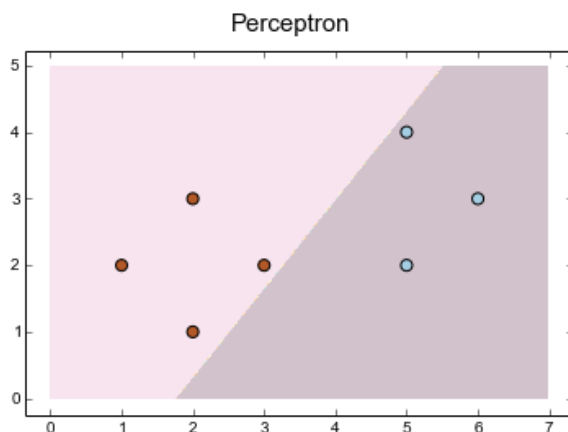
Nakon što je na RGB *frameu* provedena detekcija lica Viola-Jones postupkom, detektirana lica se dodatno provjeravaju. Ovo je korak koji čini razliku u odnosu na standardni Viola-Jones postupak.

Viola-Jones postupak kao rezultat daje omeđujuće pravokutnike. Svaki od njih omeđuje segment RGB slike koji bi trebao predstavljati lice. Odrede se segmenti dubinske slike koji prikazuju iste te objekte koji su na RGB slici detektirani kao lica. U slučaju da su RGB i dubinska slika iste rezolucije, bez paralakse, i pokrivaju isto vidno polje, bit će riječ o istim omeđujućim pravokutnicima; inače je potrebno raditi transformaciju. U našem slučaju, potrebno je ispraviti samo paralaksu/vidno polje. Ti segmenti dubinske slike se zatim predaju SVM klasifikatoru, koji ih klasificira kao lica ili ne-lice.

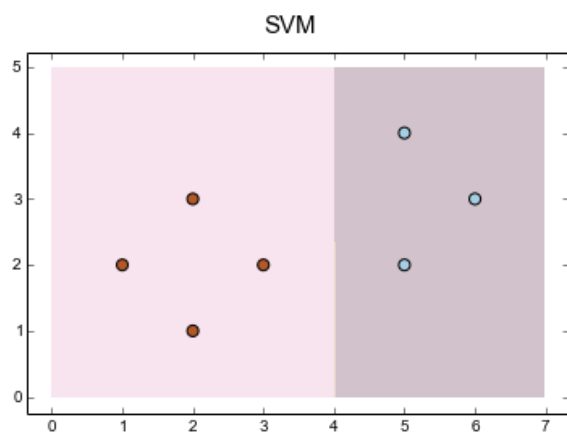
2.3.1 Stroj potpornih vektora

Stroj potpornih vektora (*support vector machine*) je popularan diskriminativni model za nadzirano učenje, koji može aproksimirati bilo koju funkciju.

Osnovni model je poopćeni linearni model bez aktivacijske funkcije: $h(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x})$ gdje je $h(\mathbf{x})=0$ decizijska hiperravnina, \mathbf{w} vektor težina, \mathbf{x} primjer, a $\Phi(\mathbf{x})$ taj primjer preslikan u *feature space*. Ako je $h(\mathbf{x}) \geq 0$, primjer se klasificira kao pozitivan; za $h(\mathbf{x}) < 0$ klasificira se kao negativan. Slično kao i kod drugih modela poput perceptrona ili logističke regresije, proces učenja se svodi na pronalazak vektora \mathbf{w} koji minimizira pogrešku na skupu za učenje, odnosno hiperravninu koja najbolje odvaja pozitivne i negativne primjere s obzirom na funkciju pogreške. Motivacija za SVM je poboljšanje perceptrona: perceptron pronalazi bilo koju (i često ne najbolju) granicu koja odvaja primjere, dok bi idealno granica bila "negdje u sredini". SVM rješava taj problem, tako da pronalazi granicu sa maksimalnom marginom. Margina je udaljenost od hiperravnine do najbližeg primjera; granica maksimalne margine je pozicionirana na pola puta između pozitivnih i negativnih primjera.



Slika 4: Decizijska granica perceptrona



Slika 5: Decizijska granica SVM-a

Udaljenost primjera \mathbf{x} od hiperravnine je $d = \frac{h(\mathbf{x})}{\|\mathbf{w}\|}$. Za hiperravninu koja ispravno klasificira primjere (a takvu i tražimo), vrijedit će $|d| = \frac{y \cdot h(\mathbf{x})}{\|\mathbf{w}\|}$, gdje je $y^{(i)} = \text{sgn}(h(\mathbf{x}))$ klasifikacija i -tog primjera (+1 za pozitivne, -1 za negativne). Margina, koju treba maksimizirati, je tada $\frac{1}{\|\mathbf{w}\|} \min_i \{y^{(i)} \cdot h(\mathbf{x})\}$. Taj izraz nije naročito pogodan za optimizaciju, pa se stvar da pojednostavniti postavljanjem apsolutne udaljenosti najbližeg primjera na $y^{(i)} \cdot h(\mathbf{x}) = 1$. Problem se tada svodi na maksimizaciju izraza $\frac{1}{\|\mathbf{w}\|}$ uz uvjete $y^{(i)} \cdot h(\mathbf{x}) \geq 1$ za $i \in [1, N]$ gdje je N broj primjera, jer je udaljenost svakog primjera veća ili jednaka udaljenosti najbližeg primjera. Radi jednostavnosti, maksimizaciju izraza $\frac{1}{\|\mathbf{w}\|}$ svodimo na minimizaciju izraza $\frac{1}{2} \|\mathbf{w}\|^2$. Problem maksimalne margine je tako sveden na problem konveksne optimizacije uz ograničenja, što se rješava pomoću metode Lagrangeovih multiplikatora.

To znači da ćemo imati minimizaciju pripadne Lagrangeove funkcije bez ograničenja:

$$L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i \{y^{(i)} (\mathbf{w}^T \cdot \mathbf{x}^{(i)} + w_0) - 1\}$$

gdje je $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N)$ vektor Lagrangeovih multiplikatora.

Ovaj problem se može rješavati direktno, ali možemo rješavati i njegov dualni problem – nalaženje maksimuma donje granice primarnog problema. Uz određene uvjete (jaku dualnost), rješenje oba problema će biti isto. Do dualne formulacije se dolazi deriviranjem po \mathbf{w} i w_0 i uvrštavanjem; dobivamo problem optimizacije dualne Lagrangeove funkcije, ovisne o dualnim varijablama α_i :

$$\tilde{L}(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^T \mathbf{x}^{(j)} \text{ uz uvjete } \alpha_i \geq 0 \text{ i } \sum \alpha_i y^{(i)} = 0.$$

Za razliku od primarnog problema, koji je imao $n+1$ varijabli (gdje je n broj značajki primjera), dualni problem ima N varijabli. To čini prijelaz u dualni problem računski isplativim kad je $N \ll n$.

Za poopćeni linearni model, u jednakostima izvoda ćemo umjesto \mathbf{x} imati $\Phi(\mathbf{x})$. Tako će, nakon pronalaska optimalnih parametara, model biti $h(\mathbf{x}) = \sum_{i=1}^N \alpha_i y^{(i)} \Phi(\mathbf{x})^T \Phi(\mathbf{x}^{(i)}) + w_0$. Iz Karush-Kuhn-Tuckerovih (KKT) uvjeta slijedi da $\alpha_i = 0$ kad i -ti primjer nije na margini (odnosno kad ne vrijedi $y^{(i)} h(\mathbf{x}^{(i)}) = 1$). U gornjoj sumi tako preostaju samo primjeri koji leže točno na margini; kako su primjeri ustvari vektori značajki, i određuju položaj margine ("podupiru" marginu), nazivamo ih potpornim vektorima, a iz toga se i cijeli model zove stroj potpornih vektora.

Ključan je način pojavljivanja primjera u dualnom izrazu: pojavljuju se isključivo u kontekstu skalarnog produkta dva primjera preslikana u *feature space* $\Phi(\mathbf{x})^T \Phi(\mathbf{x})$. Intuitivna interpretacija dualnog izraza je usporedba sličnosti danog primjera \mathbf{x} sa svim potpornim vektorima. Taj skalarni produkt možemo zamijeniti nekom funkcijom $K(\mathbf{x}, \mathbf{x}')$ koja računa skalarni produkt primjera \mathbf{x} i \mathbf{x}' preslikanih u *feature space*. Takvu funkciju zovemo jezgrenom (kernel) funkcijom. Pritom je dovoljno da K zadovoljava određene uvjete, odnosno da se radi o Mercerovoj jezgri, da bi $K(\mathbf{x}, \mathbf{x}')$ stvarno odgovaralo skalarnom

produktu primjera preslikanih u *neki* prostor. U našem rješenju se koristi stroj potpornih vektora sa radijalnim baznim funkcijama (RBF) kao jezgrama; prostor u koji se primjeri (implicitno) preslikavaju nije konačnog broja dimenzija.

Za linearno neodvojive slučajeve, koristi se varijanta modela sa tzv. mekom marginom (*soft margin*, naspram ranijeg modela sa čvrstom marginom – *hard margin*). Tu se dozvoljava da primjeri ulaze unutar margine, ali se takvi ulasci kažnjavaju, a kazna nastoji minimizirati.

2.4 Označavanje lica i kreiranje izlaznog videozapisa

Na svakom RGB frameu se iscertavaju omeđujući pravokutnici detektiranih lica koja su prošla provjeru SVM klasifikatora. Ovisno o izboru korisnika, drugom bojom se iscertavaju i omeđujući pravokutnici koji su detektirani Viola-Jones postupkom, ali nisu zadovoljili SVM klasifikator. Na taj način se ilustrira efekt SVM klasifikatora.

Tako obrađeni RGB frameovi se lijepe u izlazni videozapis.

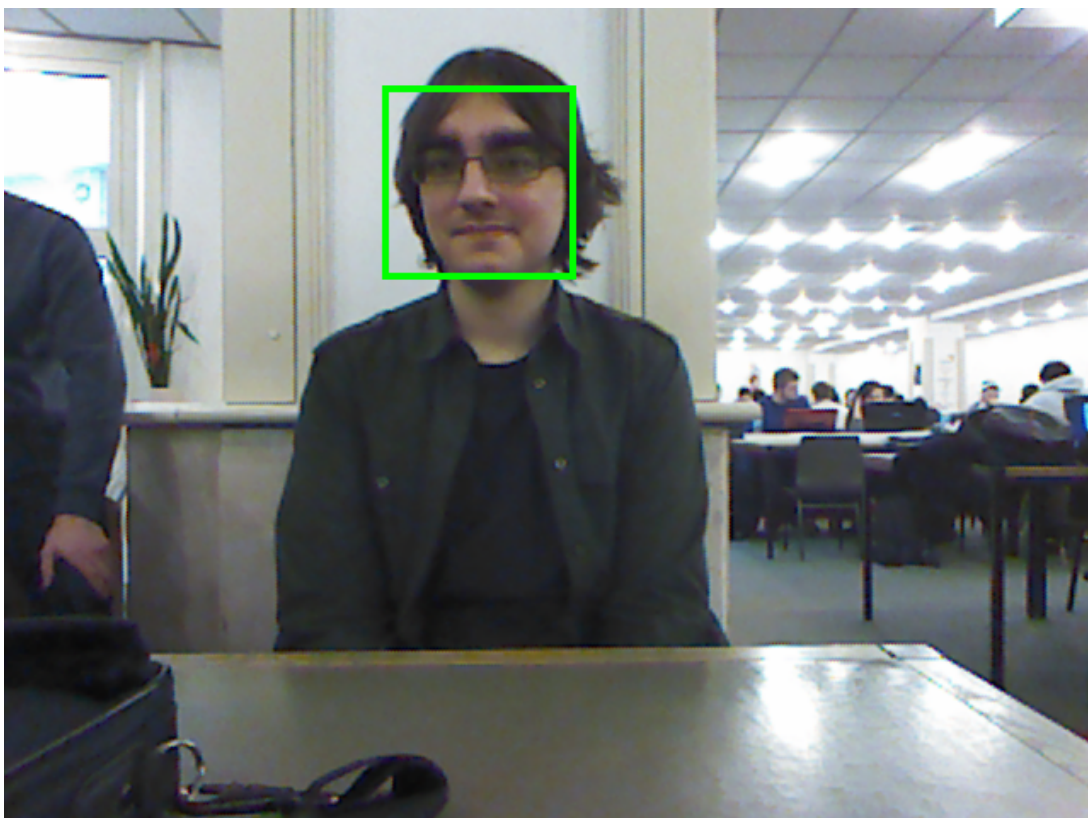
3. Ispitivanje rješenja

3.1 Ispitna baza

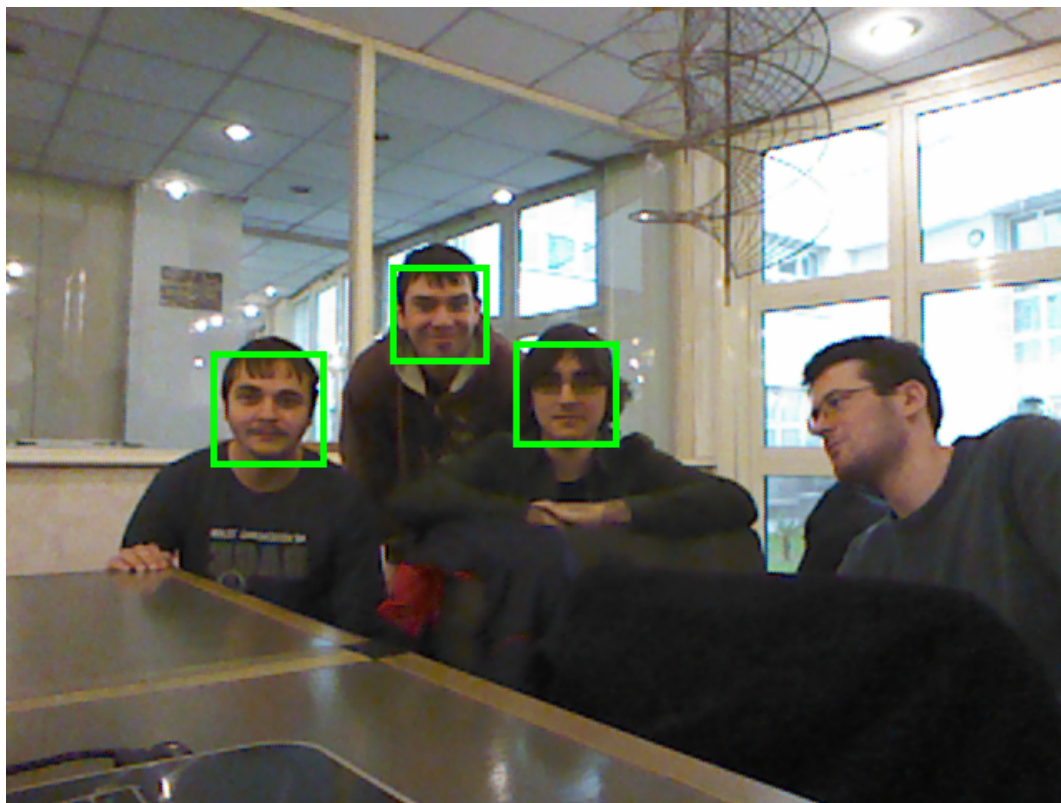
Ispitna baza se sastoji od 9 videozapisa koje smo kreirali pomoću Kinecta. 7 ih ima trajanje od 6 sekundi, 1 od 5 sekundi, i 1 od 10 sekundi. Svi su RGB-D videozapisi sa *framerateom* od 25fps.

Među zanimljivijima su sljedeći:

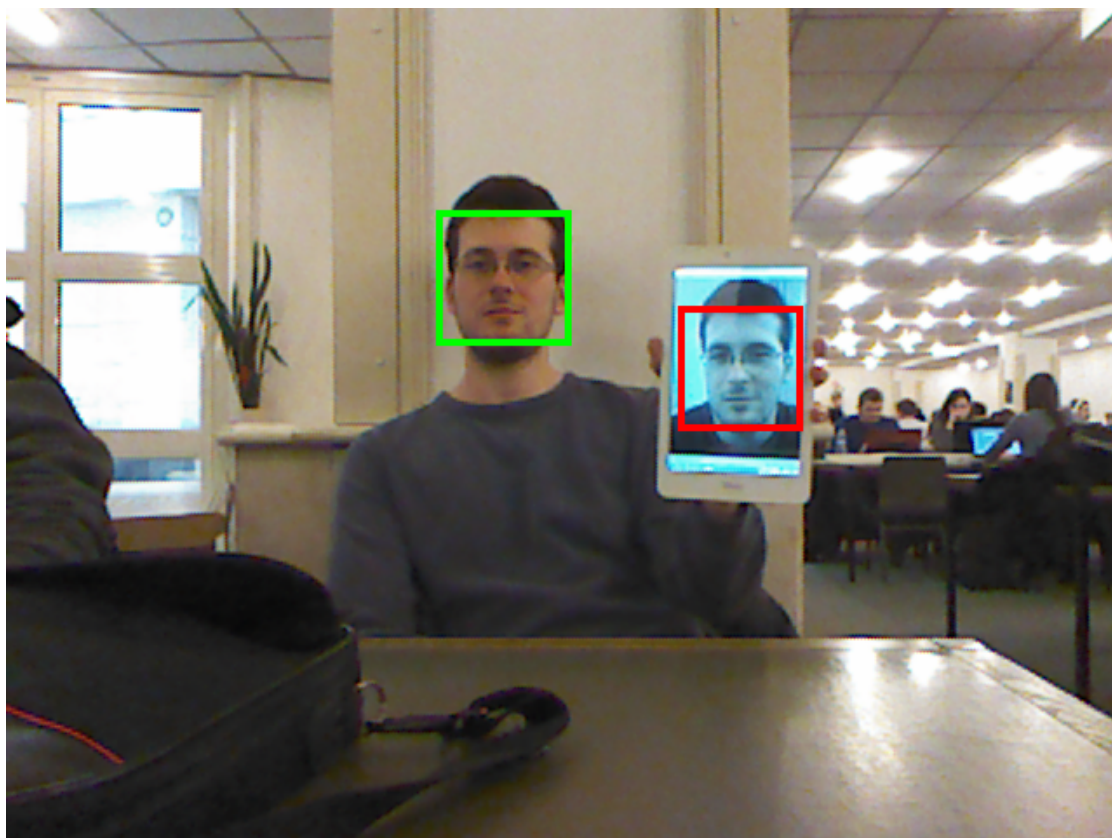
1. jednostavna snimka lica



2. snimka više ljudi, uključuje prekrivena i okrenuta lica



3. snimka pravog lica i dvodimenzionalne slike lica



3.2 Rezultati učenja i ispitivanja, analiza rezultata

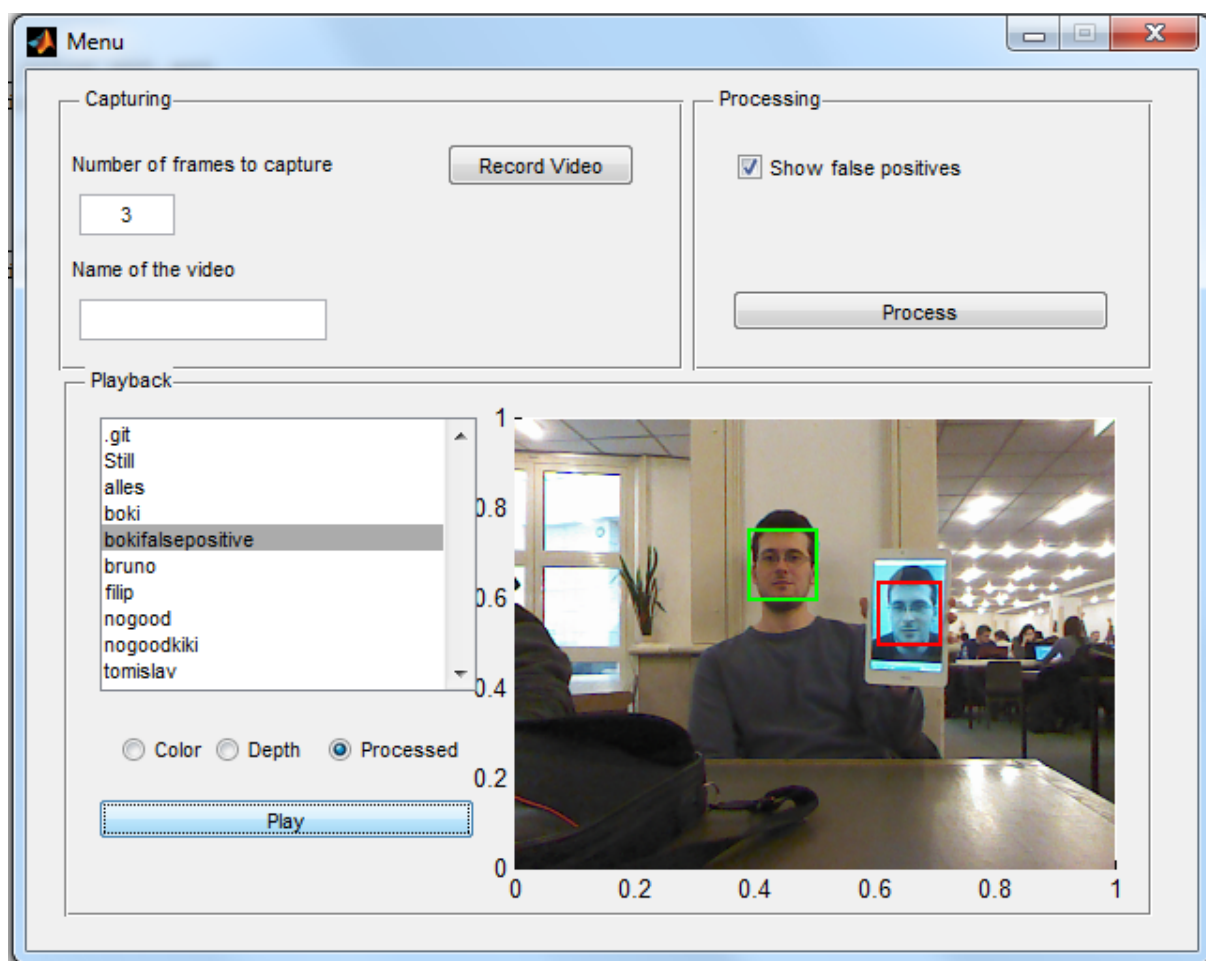
Zbog vrlo male testne baze, nije moguće generalizirati dobivene rezultate.

Na *depth* slikama jednog od videozapisa je natreniran SVM. Na danim primjerima (kao i na ostalim, koji se mogu naći u podacima koji prate dokumentaciju) su jasno vidljivi slučajevi gdje SVM ispravlja loše klasificirana lica. Također su vidljivi nedostaci osnovnog Viola-Jones postupka (ne-frontalna i ne-uspravna lica). Vremenska postojanost (kroz *frameove*) je dobra.

4. Opis programske implementacije rješenja

Programsko rješenje je implementirano u Matlabu. Korištenje zahtijeva Matlab (program) i Matlab CV toolbox. Kompletan programski kôd je raspoređen u 10 .m skripti; pokretanje početne skripte (Menu.m) pokreće grafičko sučelje. Grafičko sučelje pruža mogućnost snimanja videozapisa, pokretanja procesiranja videozapisa, te reprodukcije RGB, dubinskog, i procesiranog videozapisa.

Implementacija očekuje sve skripte u istom direktoriju, sa svakim RGB-D videozapisom u poddirektoriju, naprimjer, myDir. Videozapisi su ./myDir/color_out.avi, ./myDir/depth_out.avi i ./myDir/P_color_out.avi za RGB, dubinski, i procesirani zapis. Slike su u direktoriju ./myDir/images. Generiranje podataka skriptama producira upravo očekivanu strukturu.



Slika 6: Grafičko korisničko sučelje

5. Zaključak

Implementacija je rudimentarna, ali funkcionalna. Rezultati koje postiže, prema provedenom ispitivanju, pokazuju da je naša varijanta koja kombinira SVM sa Viola-Jones postupkom bolja nego osnovni Viola-Jones postupak. Potrebno je provesti testiranje na većim, po mogućnosti standardnim bazama kako bi se moglo propisno generalizirati i uspoređivati efektivnost i efikasnost našeg rješenja sa drugim rješenjima. Također bi bilo dobro proširiti postojeće rješenje da radi na proizvoljnim RGB-D videozapisima u .avi formatu (manja modifikacija). Značajke koje SVM u našoj implementaciji koristi su također relativno jednostavne; još bolji rezultati bi se možda mogli postići ekstrakcijom značajki više razine za ulaz SVM-u.

6. Literatura

1. G. P. Meyer, S. Alfano, and M. N. Do, *Improving Face Detection with Depth* u *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016.
2. Yi-Qing Wang, *An Analysis of the Viola-Jones Face Detection Algorithm*, u *Image Processing On Line* 4, p.128-148, 2014
3. P. Viola and M. J. Jones, *Robust real-time face detection*, u *International Journal of Computer Vision* 57(2), p137-154, 2004.