Detekcija objekata

Tehnička dokumentacija

Verzija <1.0>

Studentski tim: Ivana Babić

Filip Jakov Bulić

Borna Feldšar

Josip Gatjal

Ivan Grubišić

Lucija Šikić

Mentor: Izv. prof. dr. sc. Siniša Šegvić

SADRŽAJ

1. UVOD

Od najranijih je početaka ljudske civilizacije čovjekovo djelovanje bilo usmjereno tehnološkom napretku. Danas se taj napredak posebice očituje u razvoju računalnih znanosti, koje ujedinjuju znanstvenike prirodnog i društvenih usmjerenja te inženjere raznih struka čiji je posao otkrivanje, razvoj i oblikovanje novih pristupa raznim problemima. Skupu takvih znanosti pripada i računalni vid, odnosno znanstveno-tehnološka disciplina koja se bavi teorijom izrade umjetnih sustava koji dohvaćaju informacije iz slika te njihovom izradom. Temelj njegova razvoja čini mogućnost elektroničke percepcije ljudskog vida i digitalizacije slike, a djelovanje mu je usmjereno automatizaciji i integraciji širokog spektra procesa te prezentaciji vizualne percepcije. Klasični su problemi toga područja analiza pokreta, restauracija slike, rekonstrukcija događaja i prepoznavanje objekata. Potonji predstavlja problematiku ovoga projekta.

Svrha samog projekta jest programska implementacija rješenja problema detekcije i praćenja objekata koji se kreću. Kao ulazni podatak program prima videosnimku snimljenu statičnom kamerom. Metodama računalnog vida ta se snimka obrađuje, a rezultat obrade, odnosno projekta predstavljaju dvije izlazne snimke. Obrada snimke započinje njezinim pretprocesiranjem. Taj postupak čini zaglađivanje slika primjenom *Median filter*-a s ciljem reduciranja šuma, a daljnjom se obradom tako dobivenih slika postižu mnogo bolji rezultati u odnosu na nezaglađene. Pretprocesuirana snimka potom se koristi za modeliranje pozadine na dva načina – metodom histograma te Gaussovim modelom mješavina, a njihovi su rezultati prikazani u dvjema zasebnim izlaznim snimkama. Izgradnjom pozadinske slike omogućena je detekcija prednjeg plana, odnosno objekata koji se kreću u odnosu na pozadinu. Odvojeni pikseli pozadine i pikseli prednjeg plana omogućili su izgradnju binarne slike. U tu je svrhu pikselima pozadine pridružena crna, a pikselima koji predstavljaju prednji plan bijela boja. Daljnji je postupak obrade slike uključivao ostvarivanje praćenja kretanja objekata prednjeg plana te detekciju lica ljudi koji u skupu objekata prednjeg plana. Dodatno, omogućena je funkcionalnost slanja *e-mail*-a zadanom korisniku prilikom detekcije objekta na snimci. U privitku bi se takvog *e-mail*-a nalazila slika s označenim detektiranim objektom.

Programsko ostvarenje rješenja ovog projektnog problema uključivalo je uporabu niza metoda nad ulaznom snimkom. Zbog izrazito velike standardne knjižnice koja sadržava kvalitetno dokumentirane module te nerijetkoj praksi njegove uporabe u problematici sličnoj projektnom zadatku, za jezik smo implementacije izabrali *Python 2.7*. U testiranju i prevođenju programskog koda korišteno je razvojno okruženje *JetBrains PyCharm*. U samoj se implementaciji koda koristila biblioteka programskih funkcija *OpenCV*, *open source* biblioteka često korištena u metodama računalnog vida te obradi slike. U našem je kodu ta biblioteka korištena u postupku pretprocesiranja ulazne snimke i detekciji objekata. Nadalje, dijelovi programskog koda koji predstavljaju implementaciju algoritama za obradu slika jasno su modularizirani, što olakšava potencijalnu uporabu istih u budućnosti, ali i doprinosi samoj kvaliteti izrađene programske potpore.

Sljedeća poglavlja sadrže opisan rad na projektu, proces testiranja te zaključke o ostvarenoj programskoj implementaciji. U drugom je poglavlju opisan postupak pribavljanja i izrade videosnimki. Poglavlja 3-7 sadrže opis korištenih metoda, algoritama te načine njihove implementacije po fazama projekta. U 8. poglavlju prikazani su rezultati verifikacije i validacije programskog produkta, a u 10. poglavlju dane su upute za njegovo korištenje. U 9. je poglavlju iznesen zaključak, a u 11. poglavlju nalazi se popis korištene literature.

2. POSTUPAK IZRADE VIDEOSNIMKI

U ovom projektu video se pribavlja iz 5 megapiksela OV5647 kamere. Uz uvjet da je kamera pravilno integrirana s *Raspberry Pi-jem* te inicijalizirana, koristeći već gotove funkcije biblioteke *OpenCV*, moguće je pribaviti video te sliku po sliku iz istog videa (koja kasnije ide na obradu). Službeni kod sa stranica *OpenCV*-a nalazi se u nastavku te jasno prikazuje jednostavnost postupka pribavljanja videa i slika iz istog (http://docs.opencv.org/master/dd/d43/tutorial_py_video_display.html#gsc.tab=0).

```
1  import numpy as np
2  import cv2
3
4  cap = cv2.VideoCapture(0)
5
6  while(True):
7      # Capture frame-by-frame
8      ret, frame = cap.read()
9
10     # Our operations on the frame come here
11     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
12
13     # Display the resulting frame
14     cv2.imshow('frame',gray)
15     if cv2.waitKey(1) & 0xFF == ord('q'):
16         break
17
18 # When everything done, release the capture
19 cap.release()
20 cv2.destroyAllWindows()
```

Slika 1. Opencv v3 - getting started with videos

3. PRETPROCESIRANJE SLIKE

Kako bi postupak obrade slike dao što bolje rezultate, potrebno ju je adekvatno pretprocesirati. U konkretnom slučaju, ali i najčešće u praksi, poželjno je ukloniti moguće šumove na slici prije njezine obrade. U tu smo svrhu koristili *Median filter*, poznatu tehniku zaglađivanja slika. Uporaba te tehnike česta je u algoritmima za detekciju rubova jer pronalazi rubove uklanjajući šumove. Naime, prilikom obrade nezaglađenih slika moguća je pojava velike količine bridova koji mogu narušiti ili otežati kasnije prepoznavanje objekata.

Filtriranje po srednjoj vrijednosti (engl. *Median filtering*) metoda je iz skupine nelinearnih tehnika odstranjivanja šuma. Ideja je te metode vrijednost piksela prepoznatog kao šum zamijeniti srednjom vrijednošću vrijednosti piksela njegove okoline. Spomenuta okolina sadrži piksele u zadanom radijusu oko promatranoga, a u našoj je implementaciji korišten radijus 7 jer se testiranjem iste pokazalo kako daje najbolje rezultate. Rezultati filtriranja prikazani su na Slikama 1 i 2.

Slika 2. Bez Median Blur-a

Slika 3. Nakon korištenja Median Blur-a

4. MODELIRANJE POZADINE

Modeliranje pozadine tehnika je korištena u obradi slike i računalnom vidu, a koristi se u svrhu izdvajanja slike pozadine koja će biti korištena u daljnjoj obradi. Konkretno, u ovom se projektu obrada bavi problemom detekcije objekata prednjeg plana. Taj se problem nastoji riješiti usporedbom trenutačno promatrane slike s naučenim modelom pozadine. Rješenje se temelji na procjeni razlike vrijednosti piksela tih dviju slika. U ovisnosti o načinu izrade modela pozadine te načinu vrednovanja prethodno spomenute razlike obrađujemo dva modela. Prvi model temelji se na prikazu vrijednosti pojedinog piksela histogramom, dok drugi model svaki piksel predstavlja Gaussovom mješavinom s proizvoljnim brojem komponenata. Objašnjenja principa rada oba modela dana su u nastavku.

Važno je napomenuti da se u sklopu ovoga projekta ostvareno generiranje modela pozadine u realnom vremenu (prilikom izvršavanja), a dodatno je dana i mogućnost zasebnog provođenja učenja modela, spremanja njegova rezultata i kasnijeg učitavanja radi obrade.

4.1 Model najčešće vrijednosti u histogramu

Jedan od najjednostavnijih pristupa modeliranju pozadine uključuje izgradnju histograma za svaki piksel slike. Za svaki se piksel iz njemu pripadajućeg histograma uzima vrijednost koja se najčešće pojavljivala te se ista proglašava vrijednošću tog piksela u modelu pozadine.

Analiza se ulaznog skupa slika vrši na tri načina. U prvom se slučaju zadaje parametar odstupanja. On omogućava računanje minimalne i maksimalne vrijednosti koju pojedini piksel u analizi smije poprimiti da bi još uvijek pripadao pozadinskom skupu piksela. Ukoliko se ne nalazi unutar izračunata intervala, proglašava se pikselom prednjeg plana. Drugi način uključuje zadavanje parametra koji predstavlja postotak. Pri samoj analizi uzimaju se stupci histograma i to na način da se stupcima, odnosno pikselima pozadine proglašava najmanji broj piksela čiji se stupci nalaze oko stupca najveće visine (uključujući i njega), a da omjer njihove površine i površine cijelog histograma bude veći od zadanog parametra. Treći način optimizacija je prvoga. U njemu je programski ostvareno češće ažuriranje modela pozadine, čime nam je omogućena obrada i snimaka u kojima nije stalno osvjetljenje.

4.2 Model Gaussove mješavine. EM algoritam

Postupak modeliranja pozadine mješavinom Gaussovih razdioba provodi se na razini piksela i to tako da se vrijednost intenziteta svakog piksela modelira pomoću *n* Gaussovih razdioba u mješavini. Takva

mješavina čini probabilistički model, odnosno predstavlja linearnu kombinaciju *n* funkcija gustoće vjerojatnosti kojima opisujemo vrijednost intenziteta pojedinog piksela. Drugim riječima, funkcija miješane gustoće određena je sljedećim izrazom:

$$p(x) = \sum_{k=1}^{n} \pi_k \, p(x|\mu_k, \sum_k) \tag{4.1}$$

gdje su

$$p(x|\mu_k, \sum_k) = \frac{1}{4\pi^2\sqrt{|\sum_k|}} e^{-\frac{1}{2}(x-\mu_k)^T \sum_k^{-1}(x-\mu_k)} \tag{4.2}$$

gustoće od *x* uz odabranu komponentu *k*, a π_k apriorna vjerojatnost odabira komponente *k*.

Pri modeliranju pozadine mješavinom Gaussovih razdioba uzeta je pretpostavka da se na ulaznom skupu slika najčešće pojavljuju pikseli pozadine, koji će stoga imati veću apriornu vjerojatnost pojave u odnosu na piksele prednjeg plana. Za optimizaciju parametara modela miješane gustoće korišten je EM-algoritam, koji je objašnjen u nastavku. Konačni, optimirani parametri $(\pi_k, \mu_k, \sum_k)$ koriste se u postupku odvajanja pozadinskih od piksela objekata prednjeg plana. Uz pretpostavku da je *T* zadani omjer pozadinskih piksela i svih piksela slike, pri analizi vrijednosti određenog piksela na nekoj slici minimalan broj *B* Gaussovih razdioba najvećih težina čija je suma veća od *T* čini Gaussove razdiobe pozadine. Preostale razdiobe sadrže piksele prednjeg plana. Princip odluke o pripadnosti pojedinom od ta dva skupa jest sljedeći. Ako vrijednost piksela na određenoj slici s vjerojatnošću većom od 99.9999% pripada Gaussovoj razdiobi pozadine, tada on pripada skupini piksela pozadine te slike. Ako to nije slučaj, piksel se proglašava pikselom prednjeg plana.

4.2.1 Primjena EM-algoritma na model miješane gustoće

EM-algoritam primjenjuje se na optimizaciju parametara modela miješane gustoće. Provodi se iterativno, a uključuje uvođenje latentnih (skrivenih) varijabli. Takve varijable opisuju vezu između primjera i grupa, odnosno koriste se kako bismo odredili koji primjer pripada kojoj grupi. Njihov naziv oslikava činjenicu da izvana ne vidimo koji primjer pripada kojoj grupi. U našem je programu implementirano rješenje toga problema za proizvoljne K i N, gdje je K broj grupa, a N broj primjera sadržanih u skupu. Označimo taj skup kao $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$, a skup latentnih varijabli koje koristimo u algoritmu s $\mathcal{Z} = \{\boldsymbol{z}^{(i)}\}_{i=1}^{N}$. Sam algoritam sastoji se od dva koraka, a prije njegova provođenja potrebno je inicijalizirati početne parametre Gaussove mješavine $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$. Prvi korak naziva se još E-korak (engl. *Expectation step*). U njemu računamo očekivanje funkcije logaritamske izglednosti uzimajući u obzir trenutačne vrijednosti parametara $\theta^{(t)}$. To radimo po sljedećoj formuli, za svaki primjer $\boldsymbol{x}^{(i)}$ i svaku komponentu k:

$$\mathcal{Q}\left(\theta|\theta^{(t)}\right) = \sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)}\left(\ln \pi_k + \ln p(x^{(i)}|\theta_k)\right) \tag{4.3}$$

gdje je

$$h_k^{(i)} = \frac{p(x^{(i)}|\mu_k,\Sigma_k)\pi_k}{\sum_{j=1}^{K} p(x^{(i)}|\mu_k,\Sigma_k)\pi_j} \tag{4.4}$$

Veličinu h_k nazivamo odgovornost, a iskazuje kolika je vjerojatnost da primjer $\boldsymbol{x}^{(i)}$ pripada k-toj komponenti. Nakon toga krećemo u drugi korak, odnosno M-korak (engl. *Maximization step*). U njemu računamo procjene parametara temeljem trenutnih odgovornosti za svaku komponentu k i to prema formulama:

$$\mu_k = \frac{\sum_i h_k^{(i)} \boldsymbol{x}^{(i)}}{\sum_i h_k^{(i)}} \tag{4.5}$$

$$\Sigma_k = \frac{\sum_i h_k^{(i)} (\boldsymbol{x}^{(i)} - \mu_k)(\boldsymbol{x}^{(i)} - \mu_k)^T}{\sum_i h_k^{(i)}} \tag{4.6}$$

$$\pi_k = \frac{1}{N}\sum_{i=1}^{N} h_k^{(i)} \tag{4.7}$$

Slijedi računanje trenutne vrijednosti logaritamske izglednosti:

$$\ln \mathcal{L}(\theta|\mathcal{D},\mathcal{Z}) = \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \, p(x^{(i)}|\mu_k,\Sigma_k) \tag{4.8}$$

Iteracija, odnosno izmjena E-koraka i M-koraka prestaje postizanjem konvergencije logaritamske izglednosti ili parametara mješavine.

5. ODVAJANJE OBJEKATA PREDNJEG PLANA OD POZADINE

Kao što je već spomenuto u uvodu, nakon izgradnje modela pozadine provodi se analiza ulaznog skupa slika. Prema opisanim kriterijima u prethodnom poglavlju pikseli se pojedine slike svrstavaju ili u piksele pozadine ili u piksele objekata prednjeg plana. Piksele smo prednjeg plana označili bijelom, a piksele pozadine crnom bojom. Takav postupak nazivamo generiranjem binarne slike, a njegovo je objašnjenje dano u nastavku.

5.1 Izgradnja binarne slike

Binarnom slikom nazivamo sliku čiji pikseli sadrže jednu od dvije vrijednosti. U izrađenoj smo programskoj potpori uveli njezinu izgradnju kako bismo si pojednostavili postupak detekcije objekata prednjeg plana. Tako smo pozadini pridružili crnu, a prednjem planu bijelu boju. Primjer izgradnje dan je na Slici 2.

Slika 4. Binarna slika

5.1.1 Uklanjanje šumova na binarnoj slici

Postupak uklanjanja šumova na binarnoj slici potrebno je provesti kako bi se olakšala detekcija objekata prednjeg plana. U konkretnom slučaju, šumovi predstavljaju piksele bijele boje koji na pripadnoj ulaznoj slici zapravo nisu objekt, ali su bijelo obojeni zbog pogreške pri analizi. Više o pogreškama bit će rečeno kasnije. Takvi su pikseli najčešće izolirani ili su u skupinama od nekoliko njih. Njihovo je uklanjanje iz tog razloga lako provesti, a mi ga provodimo opcionalno, i to koristeći već spomenuti *Median Blur* s radijusom 7. Izgled izlazne slike programa prije i poslije korištenja toga filtera vidljiv je na Slikama 4 i 5.

Slika 5. Uz korištenje Median Blur-a

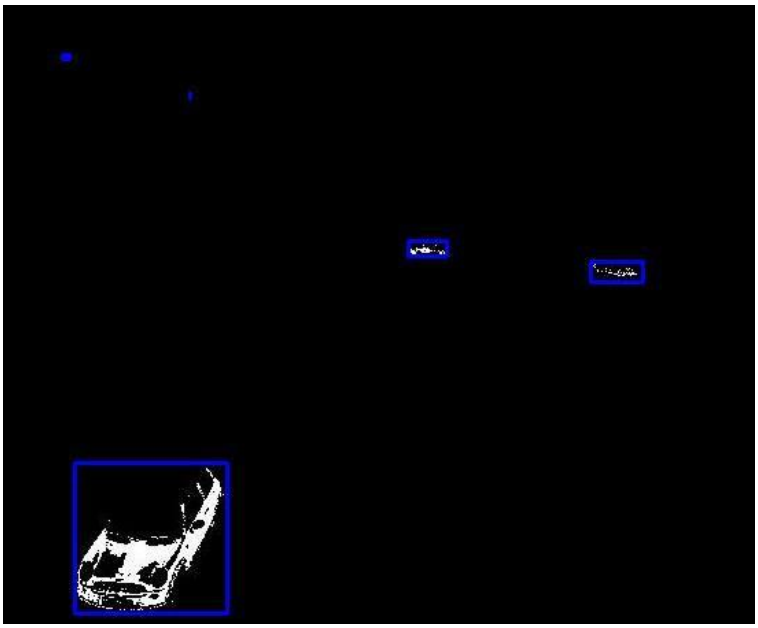
Slika 6. Prije korištenja Median Blur-a

6. DETEKCIJA OBJEKATA PREDNJEG PLANA

Nakon generiranja binarne slike i uklanjanja šuma, može započeti postupak detekcije objekata prednjeg plana. Korišteni je algoritam prikazan sljedećim pseudokodom:

```
za svaki piksel {
        odredi_kojem_objektu_točka_pripada()
}
za svaki objekt {
        pronađi_najveći_pravokutnik()
        za svaku točku objekta {
                ako se područja točaka preklapaju s najvećim pravokutnikom {
                        proširi_najveći_pravokutnik()
                }
                obriši_stare_točke()
        }
        spremi_novi_pravokutnik()
}
```

Rezultati rada tog algoritma prikazani su na Slici 1.



Slika 7. Rezulati detekcije objekata prednjeg plana

6.1 Prepoznavanje lica

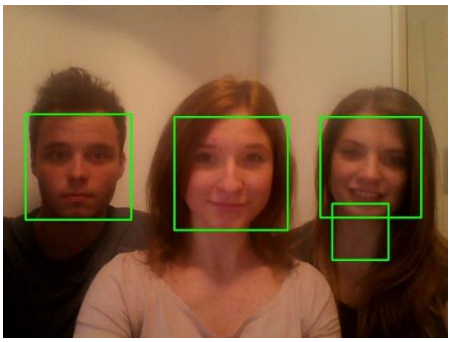
Ukoliko korisnik programa poželi implementirani sustav koristiti za detekciju uljeza, on to može odabrati prilikom pokretanja istoga. Dodatno mu je omogućeno postavljanje *e-mail* adrese na koju će mu program u realnom vremenu poslati sliku na kojoj je detektirao čovjeka.

Kako bismo mogli omogućiti tu funkcionalnost sustava, za detekciju lica koristili smo gotovu implementaciju Viola Jones detektora unutar biblioteke *OpenCV*. Budući da je spomenuti algoritam Viola Jones objavljen još 2001., postoji mnogo različitih već gotovih klasifikatora za detekciju objekata kao što su lice, oči, usne, itd. Za naš zadatak kao najprikladniji je bio klasifikator *haarcascade_frontalface_default.xml* koji omogućava uspješnu detekciju lica na slici uz zadovoljena dva uvjeta :

1) Lice koje želimo da bude detektirano mora se u potpunosti nalaziti unutar slike. Dakle, ako se na slici vidi samo npr. polovica lica neke osobe, ono neće biti detektirano.
2) Uspješna detekcija je zagarantirana tek ako su na slici vidljiva oba oka na licu koje se detektira. Dakle, ako je na slici prikazana osoba iz profila, njeno lice neće biti detektirano.

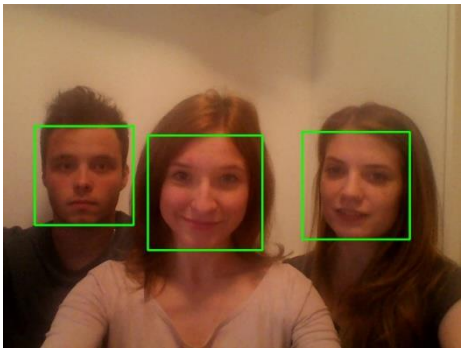
Prvi uvjet se može naći i u literaturi o načinu implementacije i korištenja klasifikatora *'frontalface'*. Drugi uvjet, međutim, nismo pronašli u literaturi već smo do njega došli pri ispitivanju funkcionalnosti koda.

Pozivanjem metode *'detectMultiScale'* počinje obrada slike i detekcija objekta. Metodu je bitno spomenuti iz razloga što njeni argumenti utječu na sam proces detekcije, tj. pomoću njih algoritam detekcije prilagođavamo slikama koje obrađujemo. Loše podešeni parametri mogu dovesti do neispravnog rada klasifikatora :



Slika 8 . Pogreška pri detekciji lica

Nakon podešavanja kritičnih parametara (*scaleFactor* na vrijednost 1.1, *minNeighbors* na 5 i *minSize* na 5x5) algoritam je uspješno detektirao sva lica:

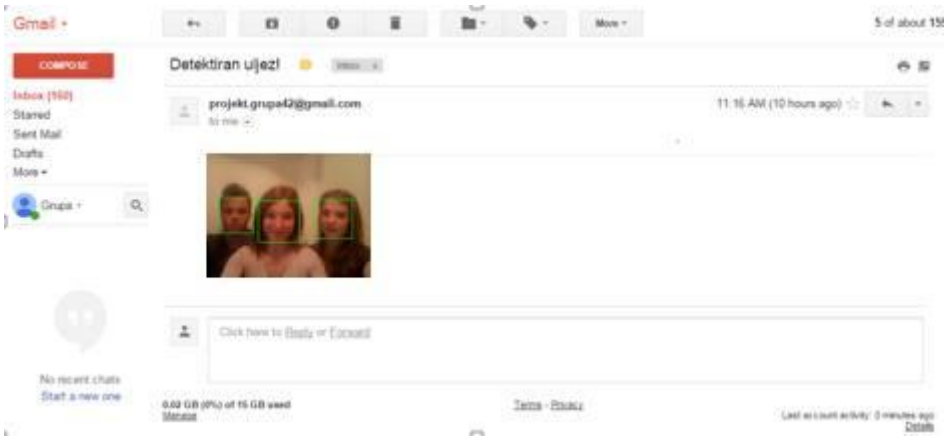


Slika 9. Ispravna detekcija svih lica

7. SLANJE *E-MAIL*-A SA SLIKOM DETEKTIRANOG ULJEZA

Nakon detekcije uljeza vrši se postupak slanja slike e-poštom na sljedeći način: funkciji za slanje mail-a kao argument šalju se relativna putanja slike na kojoj se nalazi uljez te mail adresa primatelja (koja je navedena pri početku korištenja programa).

Koriste se moduli *MIMEImage* te *MIMEMultipart* na sljedeći način: uz pomoć modula *MIMEMultipart* stvara se okvir mail-a koji će biti poslan (Subject: Detektiran uljez, mail primatelja, mail pošiljatelja...), a uz pomoć modula *MIMEImage* prilaže se slika detektiranog objekta u mail koji će se poslati. Nakon toga, uz pomoć biblioteke *smtplib* pristupa se (npr.) Google-ovom mail poslužitelju na dobro poznatim vratima (u ovom slučaju 'smtplib.gmail.com:587') te se uz pomoć korisničkog imena i lozinke, pristupa se računu elektroničke pošte i šalje prethodno formirani e-mail.



Slika 10. Primjer primljenog *e-mail*-a

8. EVALUACIJA I REZULTATI

8.1 Detekcija lica

Korištenje gotovih klasifikatora i funkcija iz biblioteke funkcija za računalni vid iz *OpenCV*-a dalo je jako dobre rezultate. Tzv. *'cascade methods'* koje čine okosnicu implementacije Viola Jones algoritma u *OpenCV* bibliotekama imaju prednost velike brzine i uz dobro zadane argumente funkcije *'detectMultiScale'* odrade jako dobar posao detekcije objekata!

Međutim, te metode imaju ograničenje da u potpunosti ovise o podešavanju argumenata za metodu *'detectMultiScale'*. Čak i ako uspješno podesimo argumente metode za detekciju objekata na tipu slika koje ćemo većinom koristiti u analizi, nemoguće je garantirati da će određeni set vrijednosti argumenata biti dobar za svaku sliku.
Dakle, u situacijama kada je potrebno odraditi detekciju objekata na setu slika na kojima veličine objekata koje detektiramo (koje ovise o udaljenosti objekta snimanja od kamere) dosta variraju, jako je teško podesiti argumente spomenute metode tako da se pri detekciji neki objekt ne detektira više puta ili da se ne detektira uopće.

Alternativa za Viola Jones algoritam danas ima dosta, ipak je taj algoritam star gotovo 15 godina. Međutim, većina tih alternativnih načina, kao npr. *'Histogram of Oriented Gradients method'* zahtjeva da se prvo odradi faza strojnog učenja na jako velikom skupu pozitivnih i negativnih slika (slika koje sadrže / ne sadrže objekte koje želimo detektirati), što zahtjeva dosta vremena, a za naš zadatak veća preciznost detekcije od one koju pružaju funkcije iz biblioteka *OpenCV*-a zaista nisu bile potrebne.

8.2 Vrednovanje točnosti modela na istom videu na kojem je naučen

Program za vrednovanje točnosti modela kao parametre prima vrstu modela pozadine, korijensku mapu s dvije podmape i dodatne parametre threshold i putanja do datoteke u koju će biti spremljen model izgenerirani model pozadine. Korijenska mapa sadrži podmape *input* i *truth*. Model pozadine se gradi učitavanjem slika iz *input* mape te nakon izgrađenog modela isti je i prikazan korisniku. Zatim slijedi izgradnja binarne slike, tj. dekekcija prednjeg plana i vrednovanje pojedinacne binarne slike sa istom slikom u *truth* direktoriju. Nakon završetka vrednovanja program ispisuje točnost modela, odnosno postotak istih piksela binarnih slika generiranih od strane modela pozadine naspram slika u *input* direktoriju.

Točnost svakog modela je vrednovano na uzorku od 1500 slika sa zadanim parametrima. Za *DistanceBackgroundModel* zadani *threshold* je 0.2, za *IntervalBackgroundModel* 0.98, a za *GaussBackgroundModel* 0.85, *k* iznosi 1, a *tolerance* 0.001. Najbolje rezultate s postotkom točnosti 99.754% je pokazao *DistanceBackgroundModel* koji je ostvaren metodom histograma. *IntervalBackgroundModel* je ostvaren također metodom histograma, a točnost modela je 99.569%. *GaussBackgroundModel* je model pozadine ostvaren pomoću Gaussove metode, točnost mu iznosi 99.215%.

9. ZAKLJUČAK

Opisana implementacija programske podrške za detekciju i praćenje objekata na videosnimci sa statičke kamere dala je dobre rezultate. Ispitivanje programa pokazalo je da se 'uljez', odnosno lice na snimci, detektira jako brzo i sa dosta velikom točnošću te da je vremenski razmak od ulaska objekta (osobe) u kadar statičke kamere do trenutka pristizanja maila sa slikom detektiranog uljeza jako kratak, što znači da će korisnik ovog sustava, u slučaju detekcije uljeza, biti jako brzo obavješten.

Svi modeli pozadine daju rezultate sa točnošću preko 99%, što garantira ispravan rad sustava neovisno o korisnikovom izboru modela pri pokretanju sustava. Međutim, najbolje rezultate s postotkom točnosti od čak 99.754% je pokazao *DistanceBackgroundModel*.

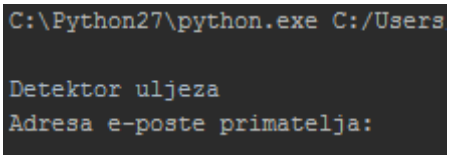
Nakon što se model pozadine izgradi, uspješno se detektiraju objekti prednjeg plana koji se kreću u odnosu na pozadinu. Postojanjem objekata prednjeg plana, slika na kojoj su detektirani šalje se na obradu da se provjeri da li je detektirani objekt zapravo osoba (lice) , tj. da li je korisnika ovog sustava potrebno obavijestiti mailom da je detektiran uljez. Ako dotični objekt nije osoba(lice), korisnika sustava se ne obavještava o detekciji tog objekta, dakle ne šalje se mail, jer se uljezima u slici smatraju samo ljudi.

Sama detekcija lica obavila se ispravno u velikom broju slučajeva i tada je korisnik sustava bio jako brzo obavješten o postojanju uljeza. Međutim, u situacijama kada je u videosnimci lice uljeza vidljivo samo djelomično dolazi do propusta, objekt prednjeg plana neće biti prepoznat kao lice i stoga korisnik sustava neće dobiti obavijest mailom. To proizlazi iz ograničenja u radu korištenog klasifikatora iz OpenCV biblioteke funkcija računalnog vida, *haarcascade_frontalface_default.xml* koji kao uvjet uspješne odrade detekcije navodi da se lice mora u cijelosti nalaziti unutar kadra. Dakle, ako videosnimka koju obrađujemo ima ispoštovan spomenuti uvjet '*frontalface*' klasifikatora, sustav radi s velikom točnošću brzinom. U suprotnom, pouzdanost sustava drastično pada jer je nemoguće sa sigurnošću tvrditi da će uljez uvijek biti uspješno detektiran.

10. UPUTE ZA KORIŠTENJE

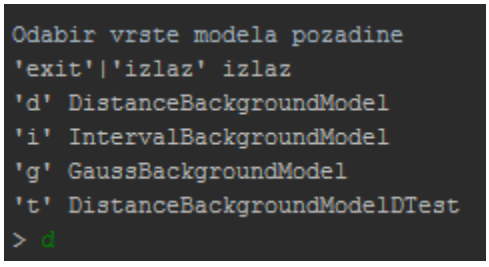
Prije korištenja same programske potpore, nužno je osigurati okruženje za njegovo izvođenje. S ciljem ostvarivanja njene potpune funkcionalnosti, korisniku se preporuča uporaba *Raspberry PI 2*, prethodno uključena na napajanje i spojena s kamerom. Također, on treba imati ostvarenu Internet vezu te, dodatno, vezu na korisničko sučelje. Važno je i osigurati da je u njegovoj trajnoj memoriji spremljen ne samo naš program, već i u prethodnim točkama spomenuta popratna programska potpora koja omogućuje njegovu izvedivost.

Kako bismo pokrenuli sustav, potrebno je pokrenuti datoteku *intruder_detector.py*. Nakon njezina pokretanja, pojavljuje se opcija u kojoj možemo navesti primatelja elektroničke pošte koji će, ako sustav detektira uljeza, primiti njegovu sliku (Slika 12). Nakon što korisnik unese valjani račun elektroničke pošte, odabir se izvrši klikom na *Enter*.



Slika 11. Mjesto upisivanja primatelja e-pošte

Sada je korisniku omogućeno prilikom upisa jedne od navedenih opcija odabrati model pozadine koji želi koristiti. Konkretno, upisuje se navedeni znak koji se nalazi uz željeni model ('d', 'i','g' ili 't') ili mod izlaza te se odabir potvrdi *Enter*-om.



Slika 12. Odabir vrste modela pozadine

U daljnjem je radu sa sustavom korisniku omogućena promjena vrijednosti parametra. Pojedini je parametar predstavljen ključnom riječju ('thr' (prag), 'buf' (buffer) ili 'med' (medijan)) te je pored njega naveden interval koji predstavlja dozvoljene vrijednosti za odgovarajući parametar. Uz napisanu promjenu potrebno je kliknuti *Enter*. Korisniku je u ovom koraku omogućeno i vratiti se na prethodne opcije odabira modela (navedena ključna riječ 'r' i *Enter*) ili nastaviti dalje (navedena ključna riječ 'c' i *Enter*), kao što se vidi na Slici 13.

```
Postavljanje parametara
'r' povratak
'c' potvrda i nastavak
'thr <broj>' prag = broj iz [0,1], (=0.2)
'buf <broj>' velicina buffera = broj, (=32)
'med <broj>' dimenzija medijana, (=7)
> c
```

Slika 13. Dodatne opcije prije pokretanja

Unošenjem i potvrdom svih parametara, omogućili smo pokretanje rada sustava. Nakon što program izgradi model pozadine, spreman je prepoznati uljeza (ako se takav pojavi). Sustav nudi i dodatne opcije koje su navedene na priloženoj slici (Slika 14).

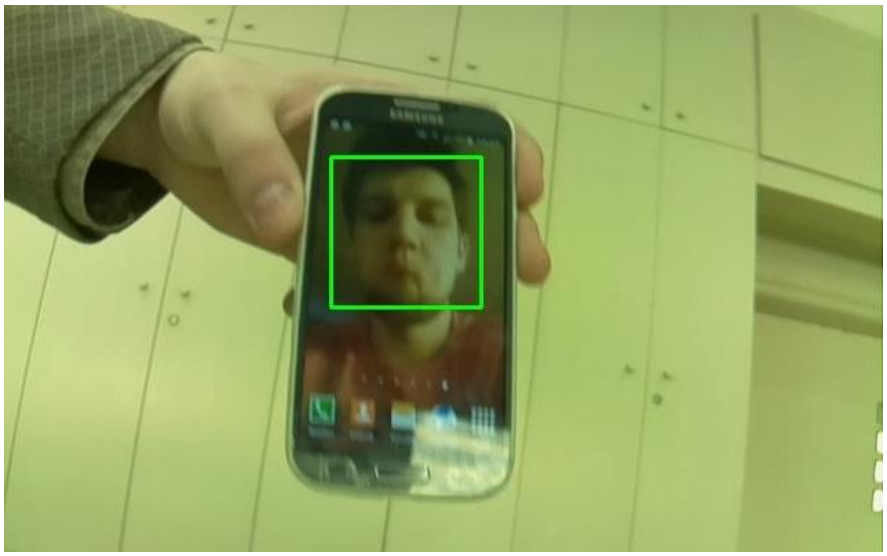
```
Inicijalizacija modela pozadine
32/32

Inicijalno racunanje modela pozadine

Cekanje uljeza
[ESC] zatvaranje programa
[r] povratak u izbornik
[n] ponovna inicijalizacija
[d] ukljucivanje/iskljucivanje prikaza
[u] ukljucivanje/iskljucivanje prilagodavanja promjenama pozadine
[e] ukljucivanje/iskljucivanje prepoznavnja lica i slanja e-mailom
[arrow key GORE]/[arrow key DOLJE] povecavanje/smanjivanje praga
[s] ukljucivanje/iskljucivanje ugladivanja
[arrow key DESNO]/[arrow key LIJEVO] povecavanje/smanjivanje ugladivanja
[CTRL]+[K] spremanje postavki u datoteku
[CTRL]+[L] ucitavanje spremljenih postavki iz datoteke
Foreground ratio:0.04
```

Slika 14. Dodatne opcije

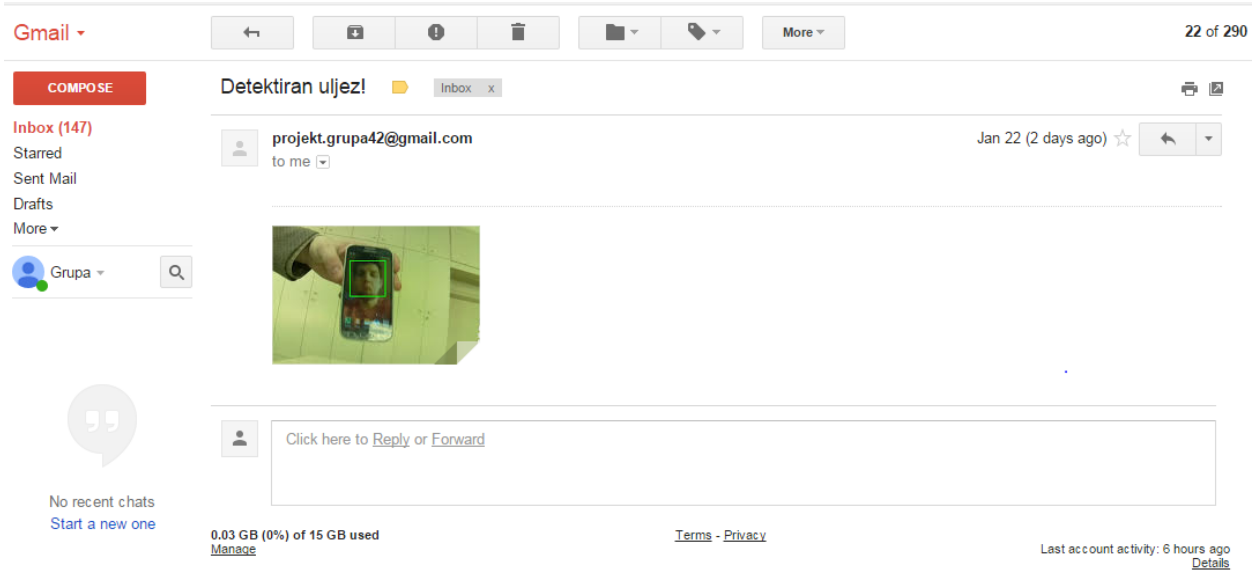
Ako imamo izlazno sučelje, moguće je vidjeti trenutnu sliku koju snima kamera (objekt – svijetliji pikseli, pozadina – tamniji pikseli), što je prikazano na Slici 15.

Slika 15. Pozadina i označeni objekt

Ako sustav na detektiranu objektu prepozna ljudsko lice, na prvotno zadani *e-mail* račun stiže odgovarajući *e-mail* sa slikom detektiranog lica uljeza (Slika 16 i Slika 17).



Slika 16. Prepoznat uljez



Slika 17. Mail sa slikom uljeza u privitku

11. TABLICA SLIKA

12. LITERATURA

[1] https://en.wikipedia.org/wiki/Background_subtraction

[2] https://en.wikipedia.org/wiki/Histogram

[3] https://en.wikipedia.org/wiki/Mode_%28statistics%29

[4] https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithmh

[5] http://docs.opencv.org/doc/tutorials/core/mat_the_basic_image_container/mat_the_basic_image_container.html#matthebasicimagecontainer

[6] http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_core/py_basic_ops/py_basic_ops.html

[7] https://en.wikipedia.org/wiki/Histogram

[8] https://en.wikipedia.org/wiki/Mode_%28statistics%29

[9] http://bmc.iut-auvergne.com/?page_id=24

[10] http://research.microsoft.com/en-us/projects/i2i/data.aspx

[11] http://research.microsoft.com/en-us/um/people/jckrumm/WallFlower/TestImages.htm

[12] http://www.vis.uni-stuttgart.de/forschung/informationsvisualisierung-und-visual-analytics/visuelle-analyse videostroeme/sabs.html

[13] https://en.wikipedia.org/wiki/Mixture_model

[14] https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

[15] http://scikit-learn.org/stable/modules/mixture.html

[16] http://www.ai.mit.edu/projects/vsam/Publications/stauffer_cvpr98_track.pdf