

# Soft kompjuting

- Projekti prethodnih generacija

# Softverski sistem za optičko prepoznavanje i određivanje starosti lica

- Andrijana Stefanović i Marko Žužić
- <https://github.com/ftn-ai-lab/sc-2016-e2/issues/71>
- Materijali prezentovani ovde preuzeti su iz diplomskih radova autora

# Zadatak

---

- Pronaći lice na video snimku
  - Rad u realnom vremenu
- Klasifikovati lice u jednu od tri predefinisane starosne kategorije:
  - Deca
  - Mlađi odrasli
  - Stariji odrasli

# Motivacija

---

- Automatsko određivanje starosti osobe moglo bi da ima ulogu u
  - Kontroli pristupa
  - Interakciji čovek-mašina
  - Identifikaciji osoba
  - Reklamiranje
- Lak zadatak za čoveka, ali veliki izazov za računare

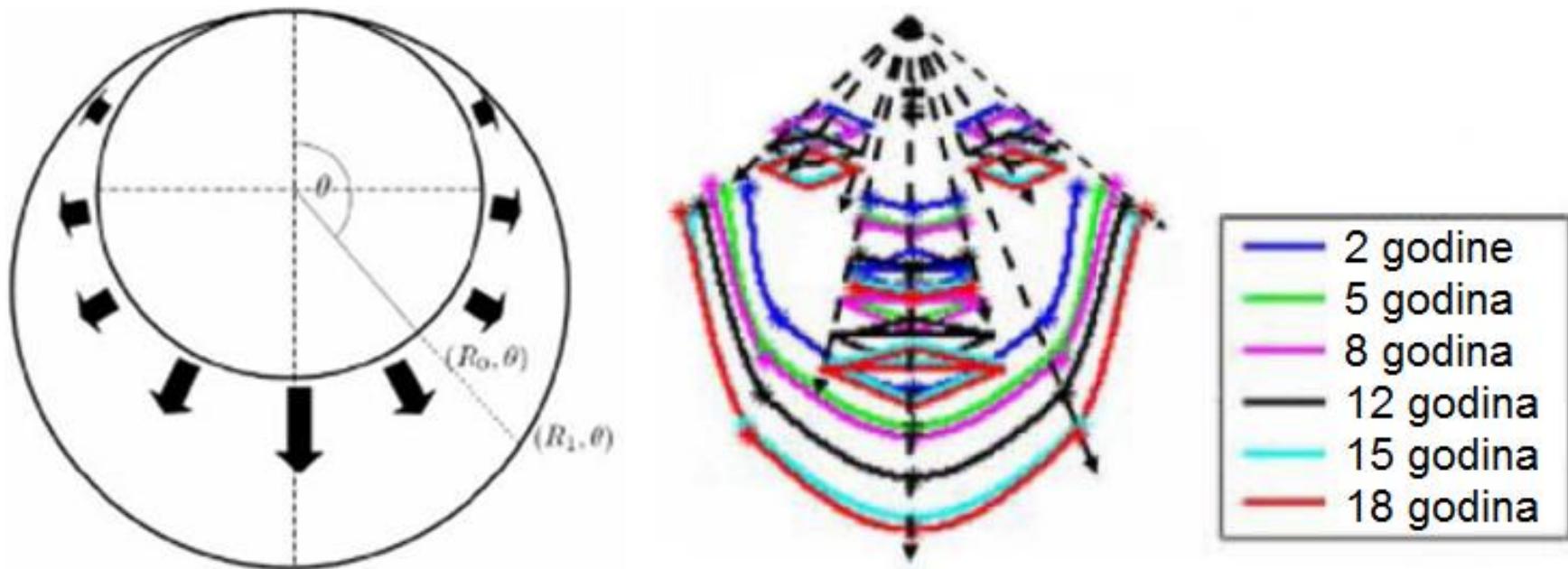
# Klasifikacija u starosne kategorije

---

- Potrebna nam je efektivna reprezentacija slike
  - Sliku predstavljamo vektorom obeležja
  - Koja obeležja da upotrebimo?
- Uticaj starenja [1]
  - Deca → Mlađi odrasli: pomeranje i rast kostiju lica
  - Mlađu odrasli → Stariji odrasli: promene u teksturi kože usled nastanka bora i smanjenja snage mišića

# Razlikovanje dece od odraslih

- Menja se oblik lica usled kraniofacijalnog rasta
  - Čelo zauzima sve manji deo površine lica, obrazi se šire, a brada postaje sve izražajnija
  - Razvijen je matematički model koji opisuje rast čovekove glave
  - Ovaj rast može biti vizuelizovan kao serija rastućih krugova koji imaju zajedničku tangentu na vrhu glave



# Razlikovanje dece od odraslih

---

- Sa slike ćemo izdvojiti lice i odrediti karakteristične odnose crta lica:

- $F1 = \frac{\text{udaljenost između unutrašnjih uglova očiju}}{\text{udaljenost oka i vrha nosa}}$
- $F2 = \frac{\text{udaljenost između unutrašnjih uglova očiju}}{\text{udaljenost oka i usne}}$
- $F3 = \frac{\text{udaljenost oka i vrha nosa}}{\text{udaljenost oka i vrha brade}}$
- $F4 = \frac{\text{udaljenost oka i vrha nosa}}{\text{udaljenost oka i usne}}$

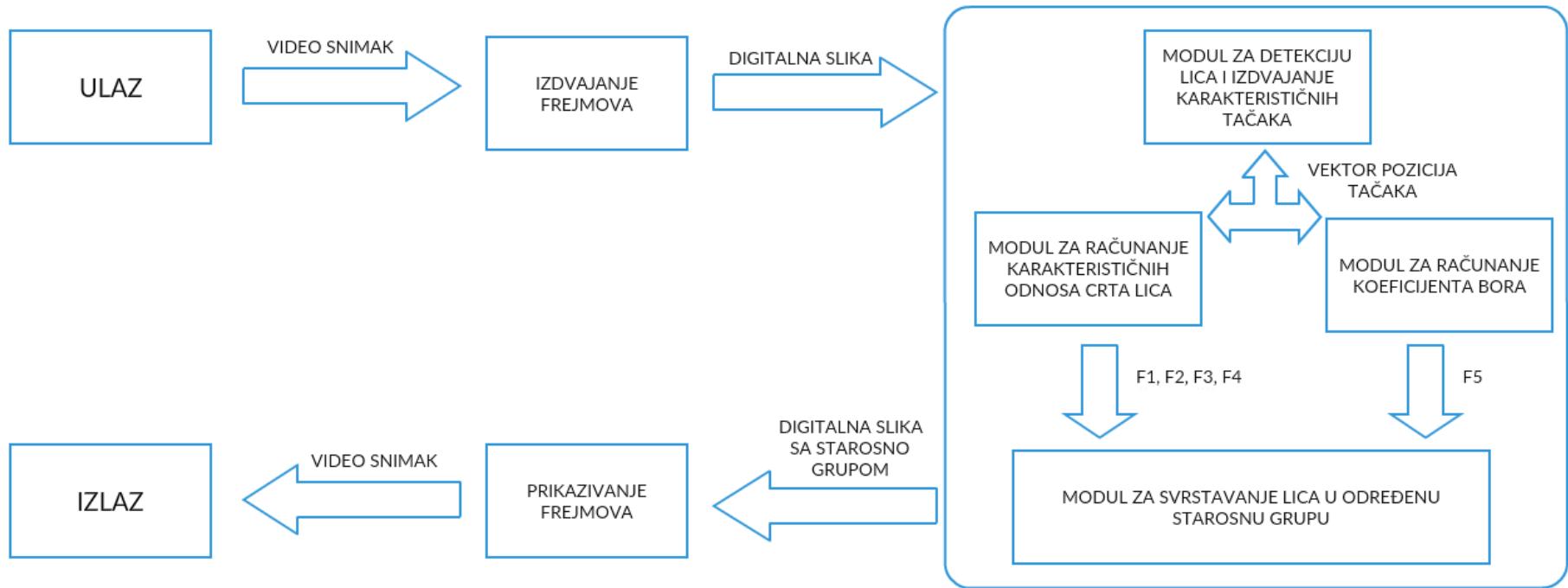
# Razlikovanje starijih i mlađih odraslih

- U odrasлом добу одвijaju se promene kože. Ona postaje tanja i manje elastična, što uzrokuje nastanak bora



- Računaćemo obeležje  $F_5$  koje predstavlja procenat bora na licu

# Konceptualni model sistema

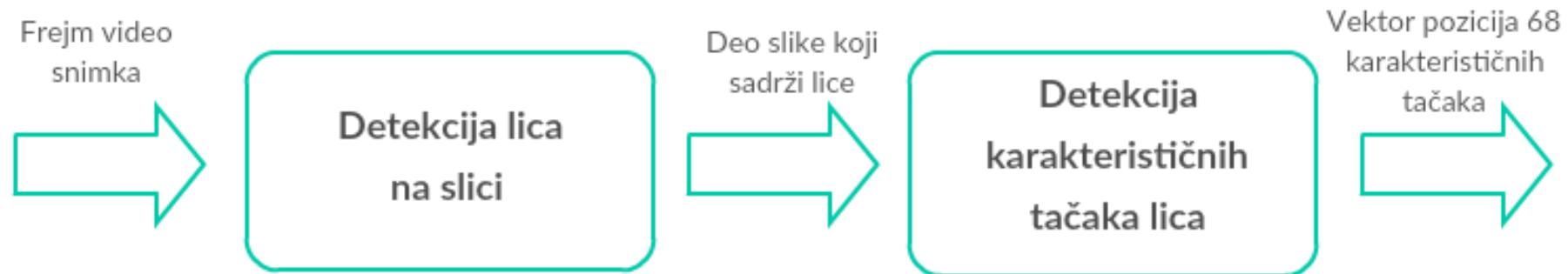


# Implementacija sistema

---

- Programski jezik *Python*
  - Biblioteka *OpenCV* → rukovanje veb kamerom i video sadržajem
  - Biblioteka *dlib* → detekcija lica i pronalaženje karakterističnih tačaka
  - Biblioteka *Keras* → implementacija neuronske mreže

# 1. Modul za detekciju lica i ponalaženje karakterističnih tačaka



- Izlaz procesa detekcije lica je granični okvir (pravougaoni isečak u kojem se nalazi detektovano lice)
  - Ovaj isečak zatim prolazi kroz detekciju karakterističnih tačaka
  - U jednom frejmu se može nalaziti više lica. U tom slučaju, za svako pronađeno lice vrši se detekcija 68 karakterističnih tačaka

# Detekcija lica na slici

---

- Detektor iz *dlib* biblioteke
- Ovaj detektor se oslanja na:
  - HOG (Histogram Orijentisanih Gradijenata)
  - Piramida slike (*image pyramid*)
  - Pomerajući prozori (*sliding window*)

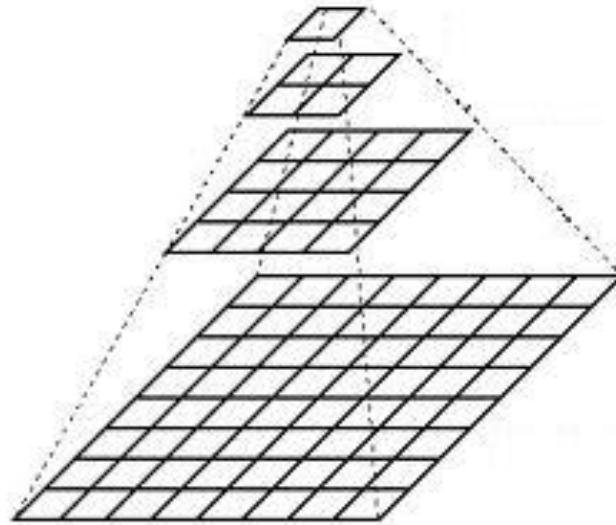
# HOG (Histogram orijentisanih gradijenata)

---

- HOG se koristi za detekciju objekata na slici
1. Predstaviti sliku kao vektor obeležja
  2. Primjeniti klasifikator (npr. SVM) koji klasificuje u dve kategorije: na slici postoji objekat/na slici ne postoji objekat
    - Svaka slika je reprezentovana HOG deskriptorom
    - Trening skup:
      - grupa slika na kojima traženi objekat postoji (pozitivni primeri) i
      - grupa slika na kojima objekat ne postoji (negativni primeri)

# Piramida slike i pomerajući prozori [3]

- Sa HOG deskriptorom imamo mogućnost da za određeni isečak slike odredimo da li on sadrži lice ili ne
- Potrebno je da pronađemo na kojoj se poziciji na slici lice nalazi
  - Pomeraćemo „prozor“ određene veličine od gornjeg levog do donjeg desnog ugla slike i za svaki testirati da li sadrži lice
- Lica mogu biti različitih veličina – koristićemo piramidu slike, tj. višenivovsku reprezentaciju slike, gde je prvi nivo originalna slika, a svaki sledeći nivo predstavlja istu sliku, promjenjene veličine



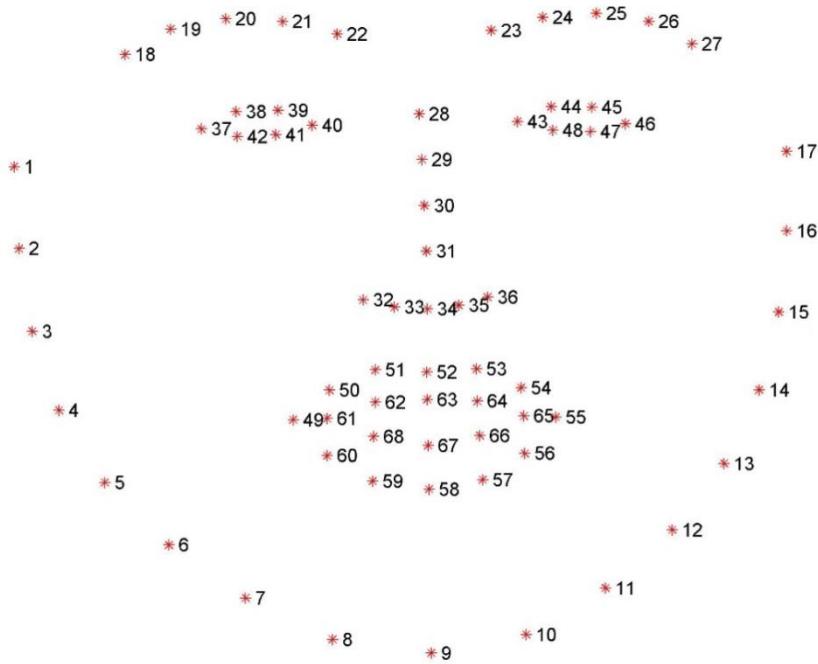
# Detekcija lica na slici

---

- Za svaku sliku iz test skupa, i za svaku veličinu slike, potrebno je primeniti tehniku pomerajućeg prozora
- Za svaki prozor potrebno je izdvojiti HOG deskriptore i primeniti klasifikator
- Ukoliko klasifikator detektuje objekat sa dovoljno velikom verovatnoćom, potrebno je sačuvati njegovu poziciju i veličinu

# Detekcija karakterističnih tačaka lica

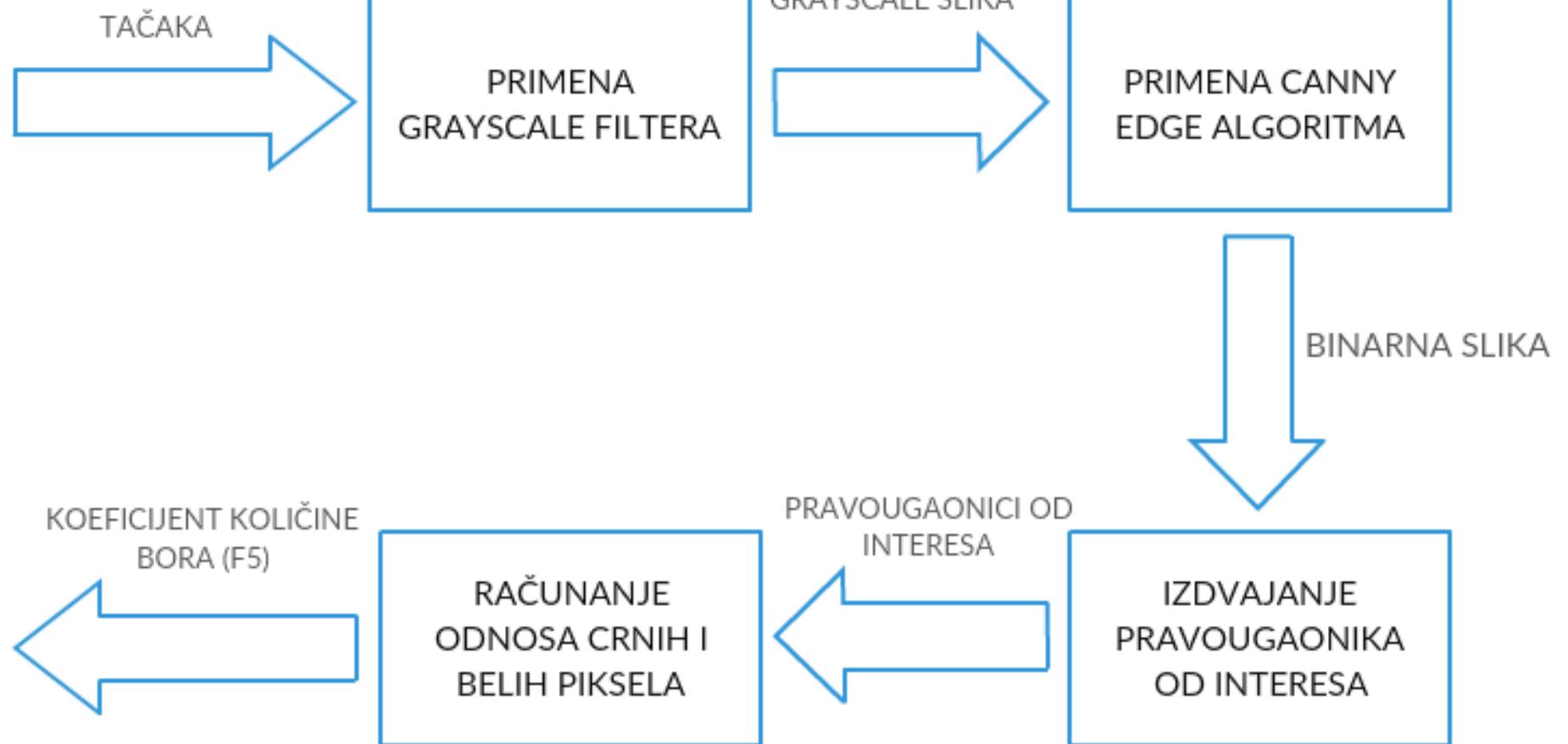
- Za ovaj korak potrebno je obezbediti granični okvir u kome se nalazi lice
- Za detekciju karakterističnih tačaka koristi se *dlib* detektor
  - Bazira se na treniranju modela na osnovu obučavajućeg skupa koji se sastoji iz slika na kojima su ručno označene koordinate regiona svake ključne strukture lica



Na osnovu dobijenih tačaka možemo lako izračunati obeležja neophodna za razlikovanje dece od odraslih (antropometrijski odnosi  $F_1 - F_4$ )

# Modul za računanje koeficijenta bora

VEKTOR POZICIJA 68  
KARAKTERISTIČNIH  
TAČAKA



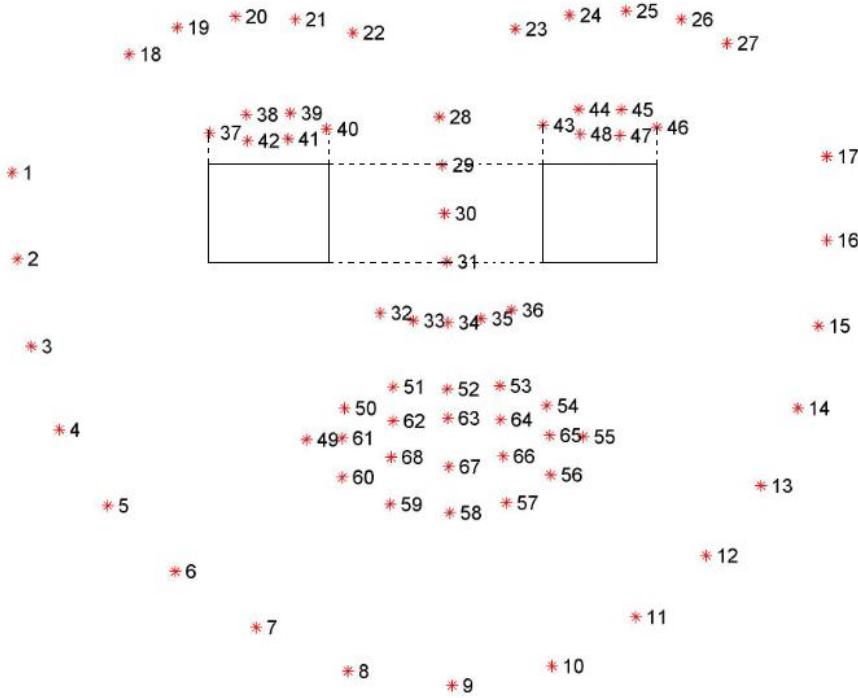
# Modul za računanje koeficijenta bora

- Ilustracija razlike u količini bora:



# Modul za računanje koeficijenta bora

- Odabrani region za računanje koeficijenta bora su obrazi
  - U ovoj regiji se u priličnoj meri manifestuju bore
  - Širina pravougaonika od interesa odgovara širini očiju, a dužina pravougaonika odgovara dužini nosa



Koeficijent se računa po formuli:

$$F5 = \frac{\text{broj belih piksela}}{\text{ukupan broj piksela}}$$

# Modul za svrstavanje lica u starosnu grupu

---

- Isprobani su neuronska mreža i  $K$ -NN algoritmi
- Svaka slika je reprezentovana karakterističnim odnosima lica ( $F_1 - F_4$ ) i koeficijentom bora ( $F_5$ )
- Izlaz predstavlja jedna od tri starosne grupe

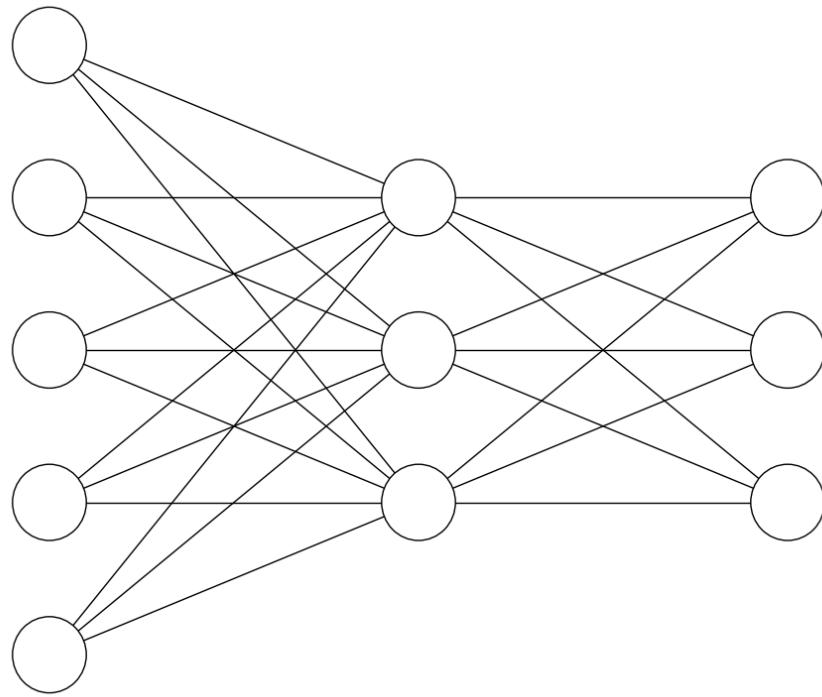
# Kreiranje obučavajućeg skupa i verifikacija

---

- Vršilo se prikupljanje slika zadovoljavajućeg kvaliteta, koje nisu retuširane
  - Bitno je da slike nisu obrađene jer su bore i neravnine na licu krucijalne za tačnu klasifikaciju
- Prikupljene slike su ručno razdeljene u odgovarajuće kategorije
- Test skup se sastojao od 24 slike, po 8 slika za svaku od kategorija

# Neuronska mreža

---



- Neuronska mreža je rezultovala tačnošću od 66.67%

# K-NN

- K-NN je rezultovao tačnošću od 71%
  - Algoritam je svu decu svrstao u odgovarajuću kategoriju
- Analiza grešaka sistema:



1a)

1b)

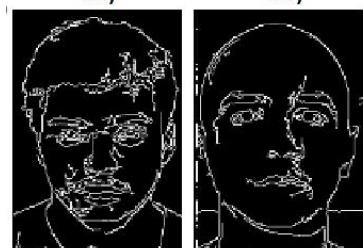
1c)



2a)

2b)

2c)



3a)

3b)

3c)

- Prve dve osobe su mlađi odrasli svrstani u kategoriju dece
  - Njihov vrh brade je na višoj poziciji od proseka, što čini njihove antropometrijske odnose bliske dečijim
  - Sistem je tačno odredio malu količinu bora u regiji obraza
- Treća osoba je klasifikovana kao mlađi odrasli
  - Razlog je zaklonjenost lica bradom što je uticalo na računanje antropometrijskih odnosa – vrh brade je označen niže nego što jeste
  - Količina bora je dobro određena

# Pravci daljeg razvoja

---

- Korišćen je relativno mali obučavajući skup
- U radu je pretpostavljeno da su lica okrenuta kameri, što čini računanje odnosa jednostavnim
- Primećuje se da rezolucija snimka ima uticaj na tačnost

# Pravci daljeg razvoja

---

- Potrebno je uzeti u obzir slučajeve kada su delovi lica zaklonjeni
  - Npr. nošenje tamnih naočara, poveza, brade, brkova,...
- Ožiljci na licu i bore nastale usred facialne ekspresije bi mogli biti pogrešno protumačeni kao bore nastale usred gubitka elastičnosti kože
- Region čela bi takođe mogao biti uzet u obzir za računanje bora
  - Problem je razlikovanje kože od kose

# Odabрана literatura

---

1. Geng, Xin. Fu Yun, Smith-Miles Kate. *Automatic Facial Age Estimation*. Monash University, Australia. Southeast University China, University at Buffalo, USA, 2010
2. *Histogram of Oriented Gradients and Object Detection*.  
<http://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/>. (Poslednji put pristupljeno 1.9.2017.)
3. *Histogram of Oriented Gradients*.  
<http://www.learnopencv.com/histogram-of-oriented-gradients/>. [Poslednji put pristupljeno 1.9.2017.]

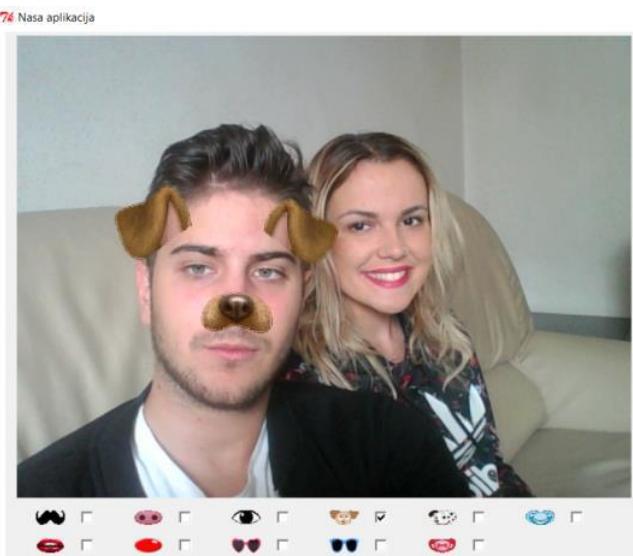
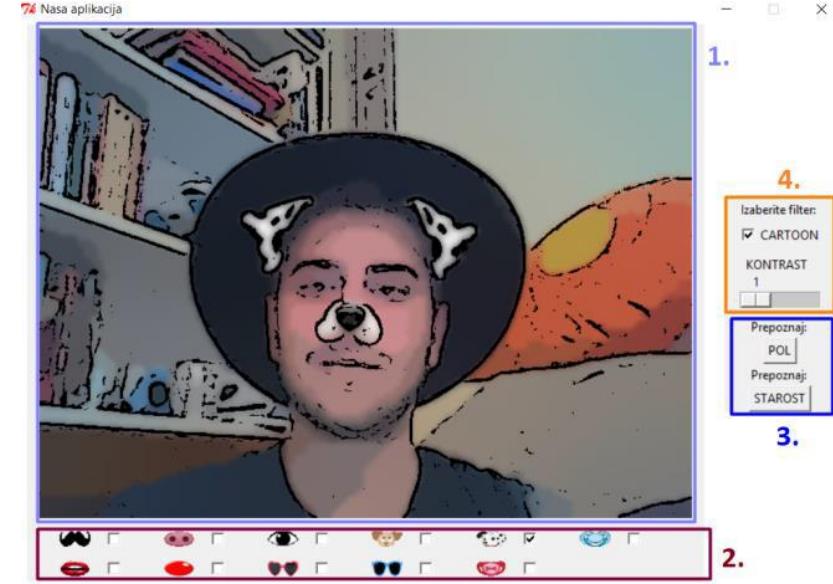
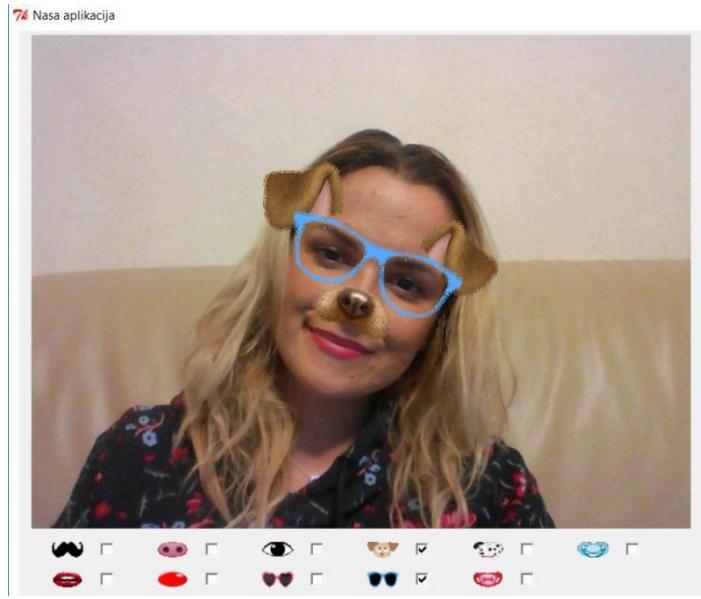
# Sličan projekat

- Tri diplomska rada:
  - Određivanje starosti i pola na osnovu lica sa videa i dodavanje efekata i stikera (po uzoru na Snapchat)
  - Nebojša Sarvan, Milica Nikolić i Nina Manojlović
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/44>



- U odnosu na prethodni projekat, autori su aproksimirali dodatne tačke na licu kako bi rešili problem rotacije lica
- Rešavan je i problem različite veličine lica usred različite udaljenosti od kamere
- Rešavan je i problem određivanja pola: korišćene su morfološke razlike muškog i ženskog lica

# Sličan projekat



- Problem:
- Više lica na kameri
- Brzina reprodukcije videa

# Praćenje kretanja igrača na video snimku fudbalske utakmice

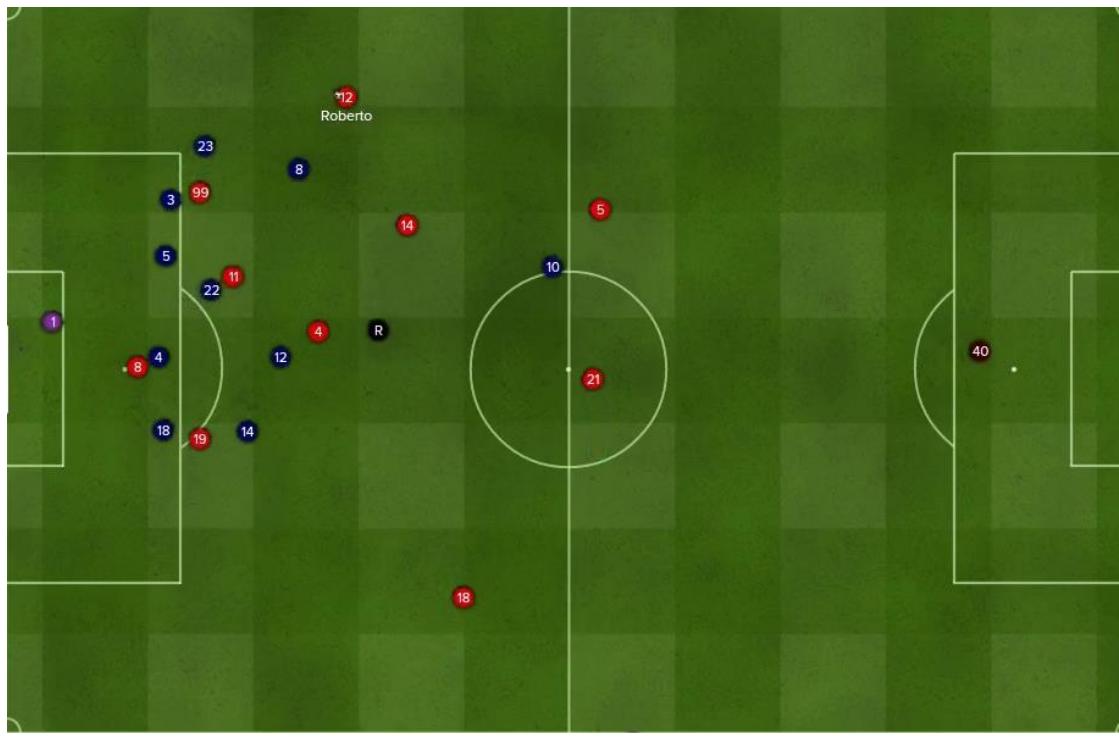
- Đorđe Marjanović
- Materijali prezentovani ovde preuzeti su iz diplomskog rada autora

# Motivacija

---

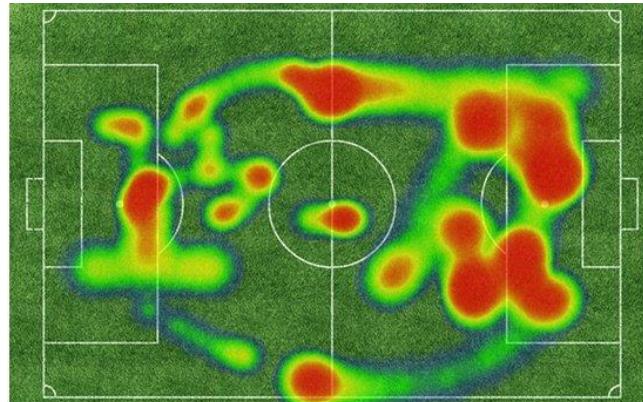
- Prikupljanje i analiza statističkih podataka predstavlja osnovu za praćenje učinka igrača kao pojedinca, ali i celog tima
  - U fudbalu se za timsku statistiku beleži posed lopte, broj kornera, broj faulova, broj šuteva na gol, broj dodavanja, pretrčana kilometraža,...
- Nekada, statističke podatke su unosili treneri ili statističari
- Sa razvojem tehnologije, pojavila se mogućnost i potreba da se unapredi proces prikupljanja statističkih podataka
  - Na primer, sportistima se na opremu postavljaju senzori ili se na stadionima ugrađuju sistemi video kamera koji prate kretanje igrača, njihovu brzinu, poziciju na terenu,...

# Zadatak



Obraditi video snimak fudbalske utakmice – detektovati igrače i pratiti njihovo kretanje

- igrica *Foodball manager*



Rezultat rada aplikacije je grafički prikaz prosečne pozicije igrača u formi *heat-mape*

# Konceptualni pregled sistema



# Obrada digitalnih slika i video sadržaja

---

- Prvi korak je da se iz video snimka izdvoje slike (frejmovi)
- Na svakoj izdvojenoj slici se detektuju igrači:
  - $RGB \rightarrow grayscale \rightarrow$  binarizacija
  - Konačno, vrši se segmentacija slike – slika se deli na segmente među kojima se izdvajaju oni sa brojem
    - Iskorišćen je *Connected-components labelling* postupak [1]

# Prepoznavanje broja sa slike

---

- Najbitniji deo – tačnost sistema direktno zavisi od toga da se broj tačno prepozna
- Za obučavanje modela korišćen je *MNIST* skup podataka  
<http://yann.lecun.com/exdb/mnist/>
  - 70 000 slika cifara
  - Slike su binarne, dimenzije  $28 \times 28$
- U svrhe treniranja modela, skup podataka je podeljen na:
  - Trening skup – 50 000 primera
  - Validacioni skup – 10 000 primera
  - Test skup – 10 000 primera

# Prepoznavanje broja sa slike

---

- Korišćene su neuronske mreže jer su u domenu klasifikacije slika pokazale veoma dobre rezultate [2]
- Arhitektura neuronske mreže sa propagacijom signala unapred:
  - *Fully connected layer* – 512 neurona, *relu* aktivaciona funkcija
  - *Dropout layer* [3] – koeficijent 0.2
  - *Fully connected layer* – 512 neurona, *relu* aktivaciona funkcija
  - *Dropout layer* – koeficijent 0.2
  - *Fully connected output layer* – 10 neurona, *softmax* aktivaciona funkcija [4]
- *Softmax* aktivaciona funkcija na izlaznom sloju daje vektor dužine 10, koji predstavlja verovatnoću svake klase
- Arhitektura je određena na osnovu rada [3]

# Prepoznavanje broja sa slike

---

- Optimizacija na validacionom skupu je služila za podešavanje parametara mreže
  - *learning rate*: 0.001, *rho*: 0.9, *epsilon*:  $10^{-8}$ , *decay*: 0.0, ...
  - Optimizovan je i broj epoha (najbolje se pokazalo 10 iteracija)
- Dobijena je tačnost od 98.7% na *MNIST* skupu podataka

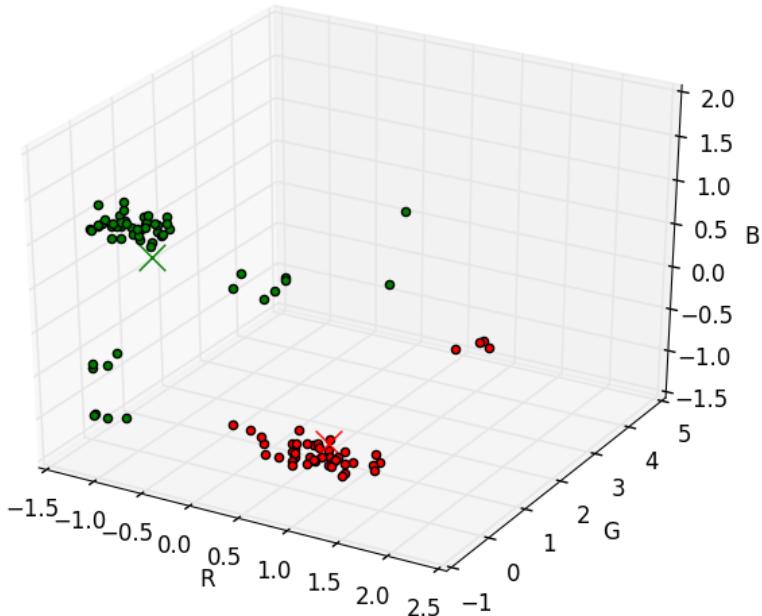
# Prepoznavanje broja sa slike

---

- Sve slike koje je potrebno klasificirati su skalirane na dimenzije  $28 \times 28$
- Slike su predstavljene kao vektori intenziteta piksela, gde su vrednosti vektora bile realne u opsegu  $[0,1]$

# Određivanje pripadnosti timu

- Identifikovani su brojevi svakog od igrača
- Ali broj je jedinstven samo unutar jednog tima
- U svrhu razvrstavanja igrača u timove, primenjena je *k-means* klasterizacija sa 2 klastera (koja predstavljaju 2 tima)
- Igrači se prepoznaju prema boji dresova – obeležja su pikseli sa informacijom o boji (vektor *RGB* vrednosti)
- Podatke je neophodno normalizovati, a kao metrika rastojanja je korišćena *Euklidska udaljenost*



Problem: algoritam greši kod golmana  
čij se dresovi razlikuju

# Skup podataka

---

- Korišćen je skup podataka od 7 video snimaka
- Svaki snimak predstavlja simulaciju fudbalske utakmice iz igrice *Foodball manager*
  - <https://www.dropbox.com/s/lryox4s2l3a5q0q/FMtekme.zip?dl=0>

# Zaključak

---

- Rezultat: za svakog igrača se čuvaju koordinate na frejmu i rezultati se prikazuju putem *heat mape*
- Mogući pravci razvoja:
  - Predikcija daljeg kretanja igrača
  - Analizirati putanju lopte
  - Računati dodatne statističke podatke: prosečna brzina, pretrčana kilometraža, broj dodavanja,...

# Odabraná literatura

---

1. R. Fisher; S. Perkins; A. Walker; E. Wolfart (2003). "Connected Component Labeling,,
2. Ciresan, Dan Claudiu, et al. "Deep big simple neural nets excel on handwritten digit recognition", 2010
3. Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of machine learning research 15.1 (2014): 1929-1958
4. <http://cs231n.github.io/linear-classify/>

# Prepoznavanje i prebrojavanje pešaka koji su prešli pešački prelaz, dok je upaljeno crveno svetlo na semaforu

- Jelena Janković i Tamara Mrkšić
- <https://github.com/ftn-ai-lab/sc-2016-e2/issues/90>
- Materijali prezentovani ovde preuzeti su iz diplomskog rada  
autora Tamare Mrkšić

# Motivacija

---

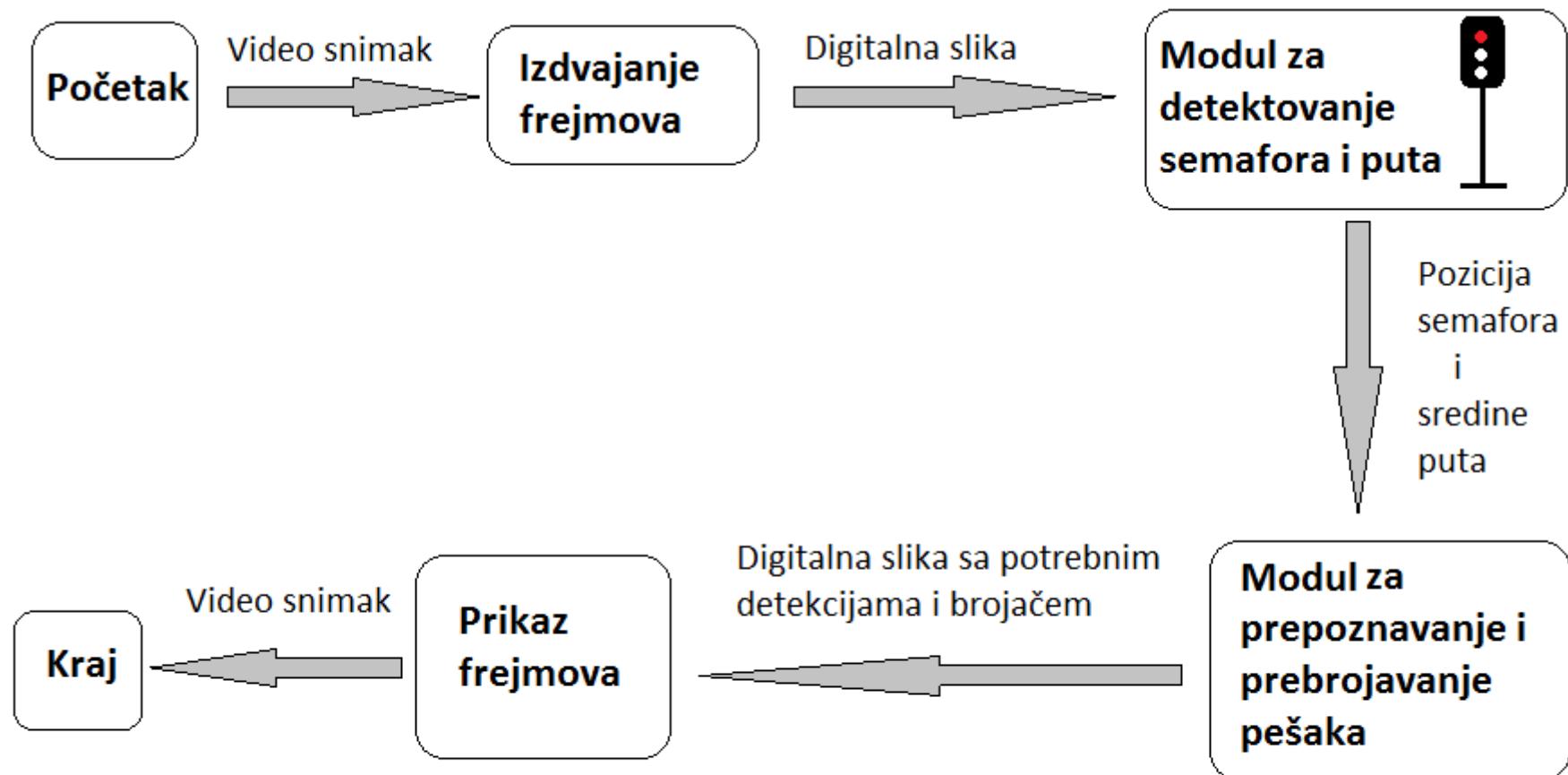
- Algoritam bi se mogao primeniti u saobraćaju sa ciljem povećanja bezbednosti

# Rešavani problem

---

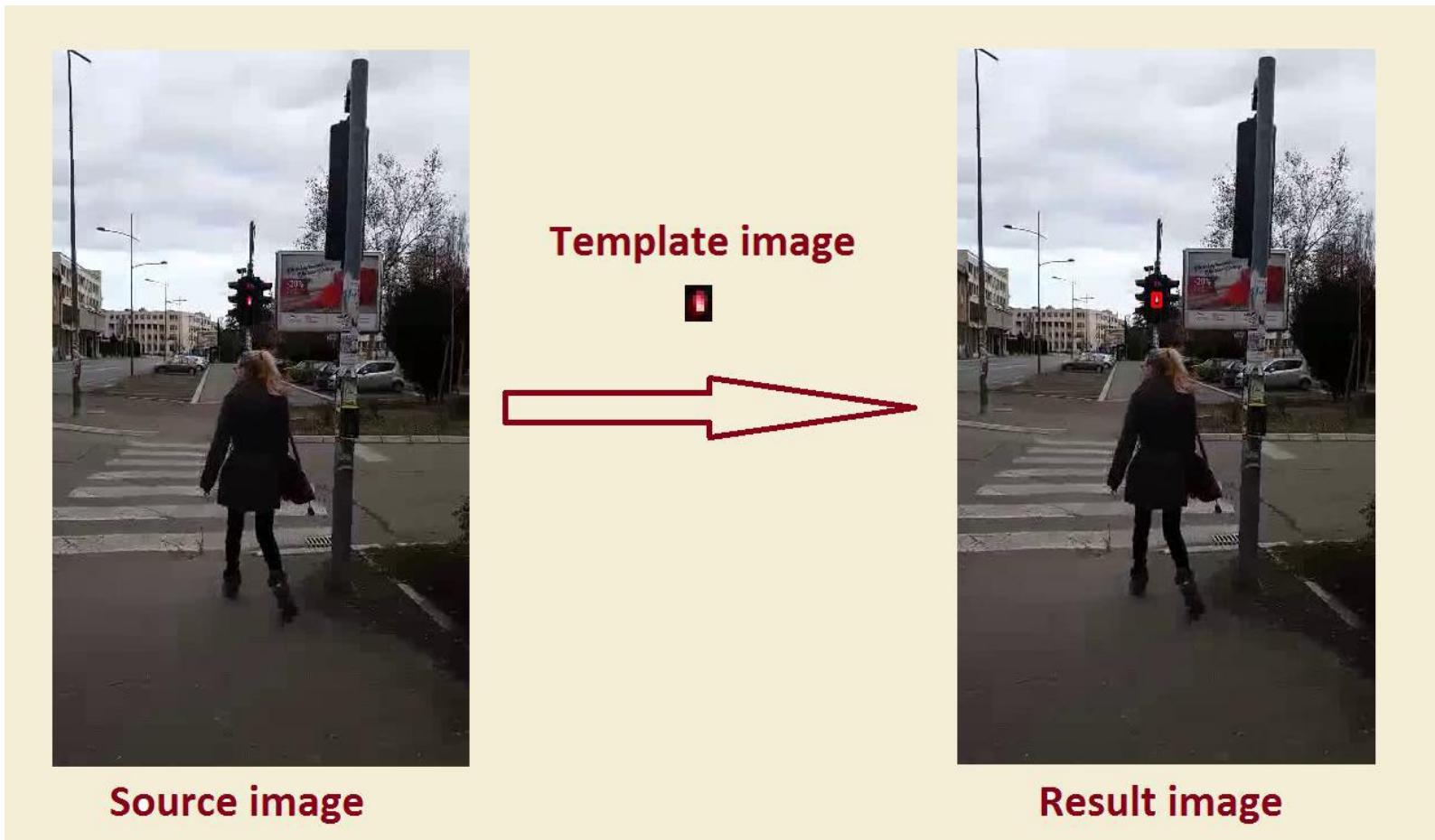
- Detektovanje semafora na snimku
- Detektovanje najjačeg svetla na semaforu, odnosno onog koje je uključeno
- Detektovanje granica puta
- Detektovanje ljudi koji su prešli granice
- Prebrojavanje ljudi koji su prešli granice na crveno svetlo

# Konceptualni pregled sistema



# Detekcija semafora sa upaljenim crvenim svetлом

- Primenjen je *Template matching* – tehnika za pronalaženje dela slike koji se najviše podudara sa uzorkom (koji se na slici pronalazi)
  - Pomeramo templejt po slici i za svaku poziciju računamo metriku koja pokazuje koliko dobro se templejt poklapa sa slikom na datoј lokaciji

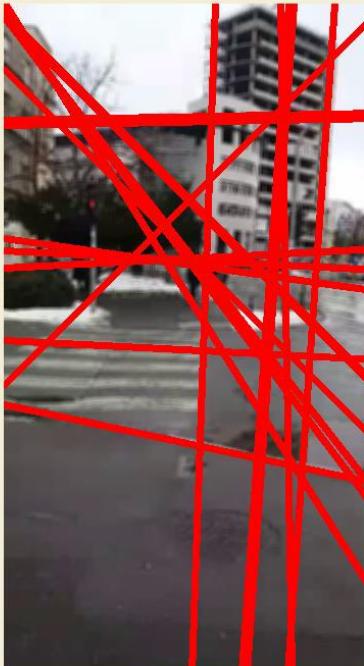


# Detekcija puta

1. Učitava se slika u boji
2. Pomoću *Canny edge detection* tehnike su detektovane ivice – rezultat je binarna slika
3. Primenjena je *Hough transformacija*
4. Postavljen je prag za detekciju svih linija



Hough  
Transformacija



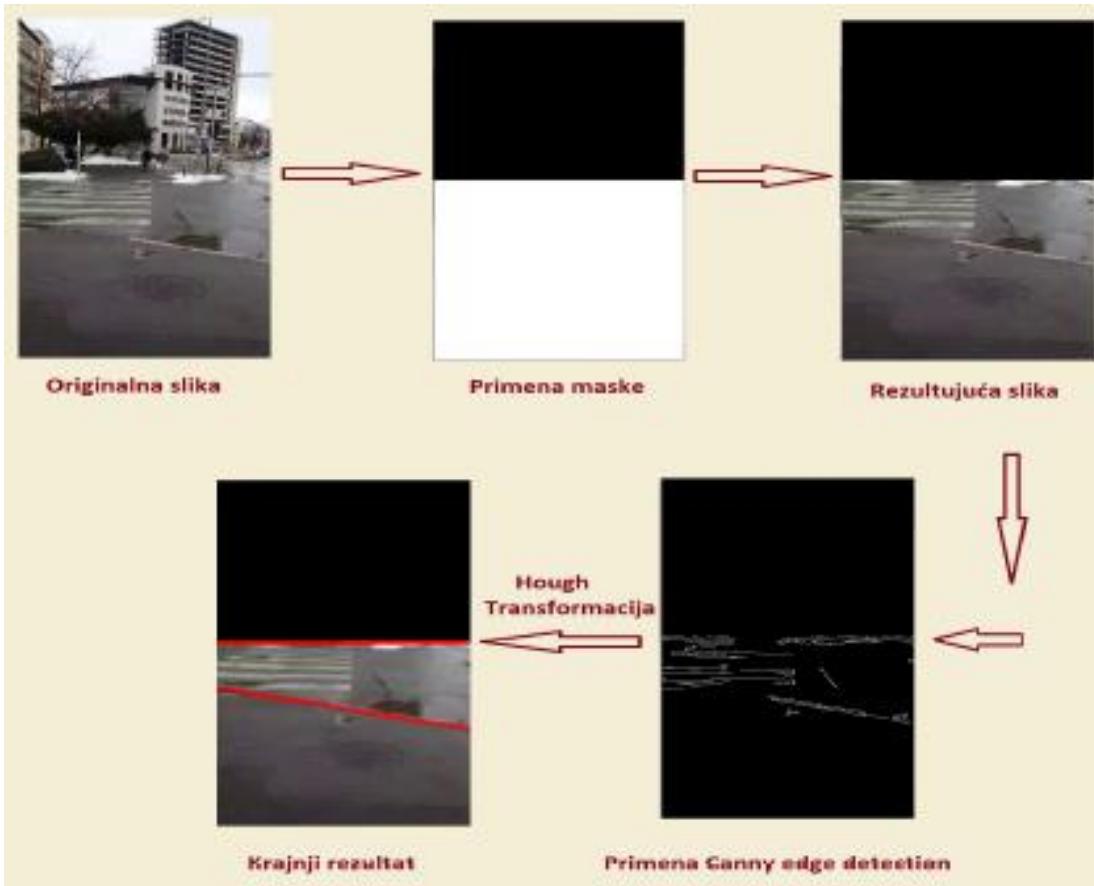
Originalna slika

Rezultujuća slika

rezultati nisu bili zadovoljavajući, jer je previše ivica prepoznato

# Detekcija puta

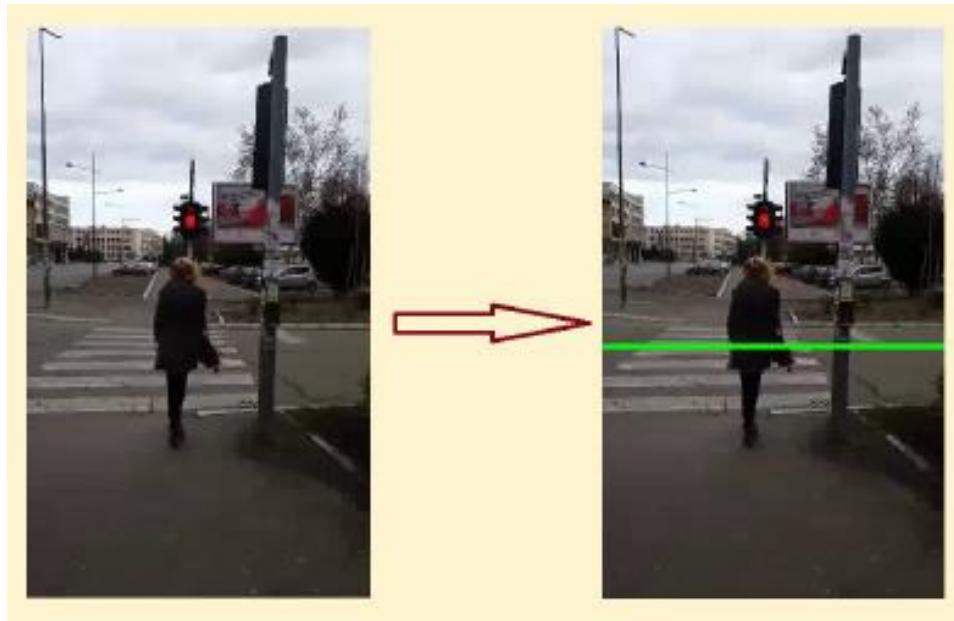
- Rešenje: koristiti masku kako bi se uklonili nepotrebni delovi slike



Radilo je bolje, ali rezultati nisu uvek bili zadovoljavajući

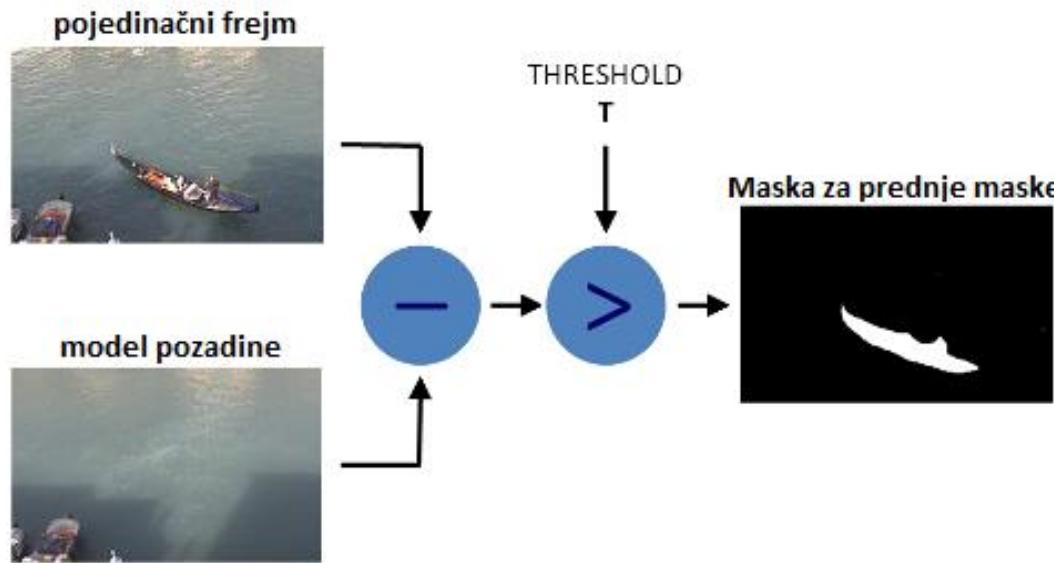
# Detekcija puta

- Snimci su napravljeni sa mesta, koje je približno udaljeno od semafora kao i pešačkog prelaza, koji se posmatra
  - Uz korišćenje te informacije i pozicije semafora (detektovan na slici pomoću *template matching*), određena je približna pozicija sredine pešačkog prelaza
  - Na osnovu toga, dobijena je jednačina prave, koja predstavlja sredinu pešačkog prelaza koju pešaci moraju preći kako bi prešli ulicu



# Modul za prepoznavanje pešaka

- Potrebno je da izdvojimo pešaka od pozadine
  - Oduzimamo trenutni kadar od pozadinskog modela (statički deo scene)



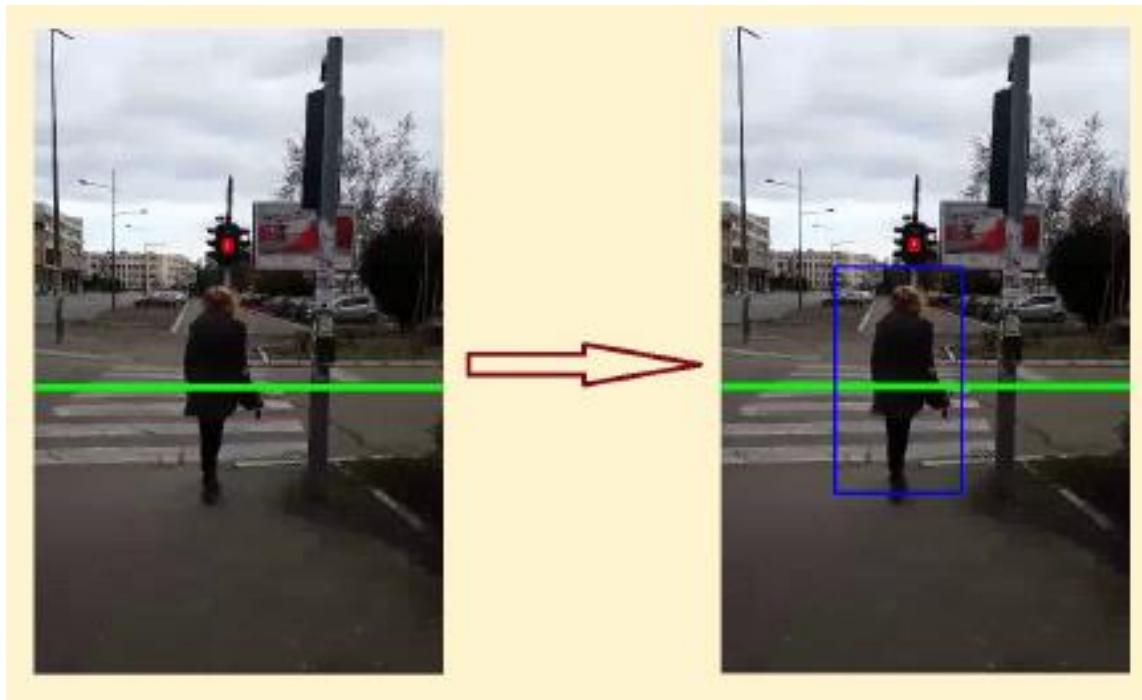
- Najpoznatiji algoritmi ovog tipa su MOG, MOG2 i GMG
  - [http://docs.opencv.org/trunk/db/d5c/tutorial\\_py\\_bg\\_subtraction.html](http://docs.opencv.org/trunk/db/d5c/tutorial_py_bg_subtraction.html)

# Modul za prepoznavanje pešaka

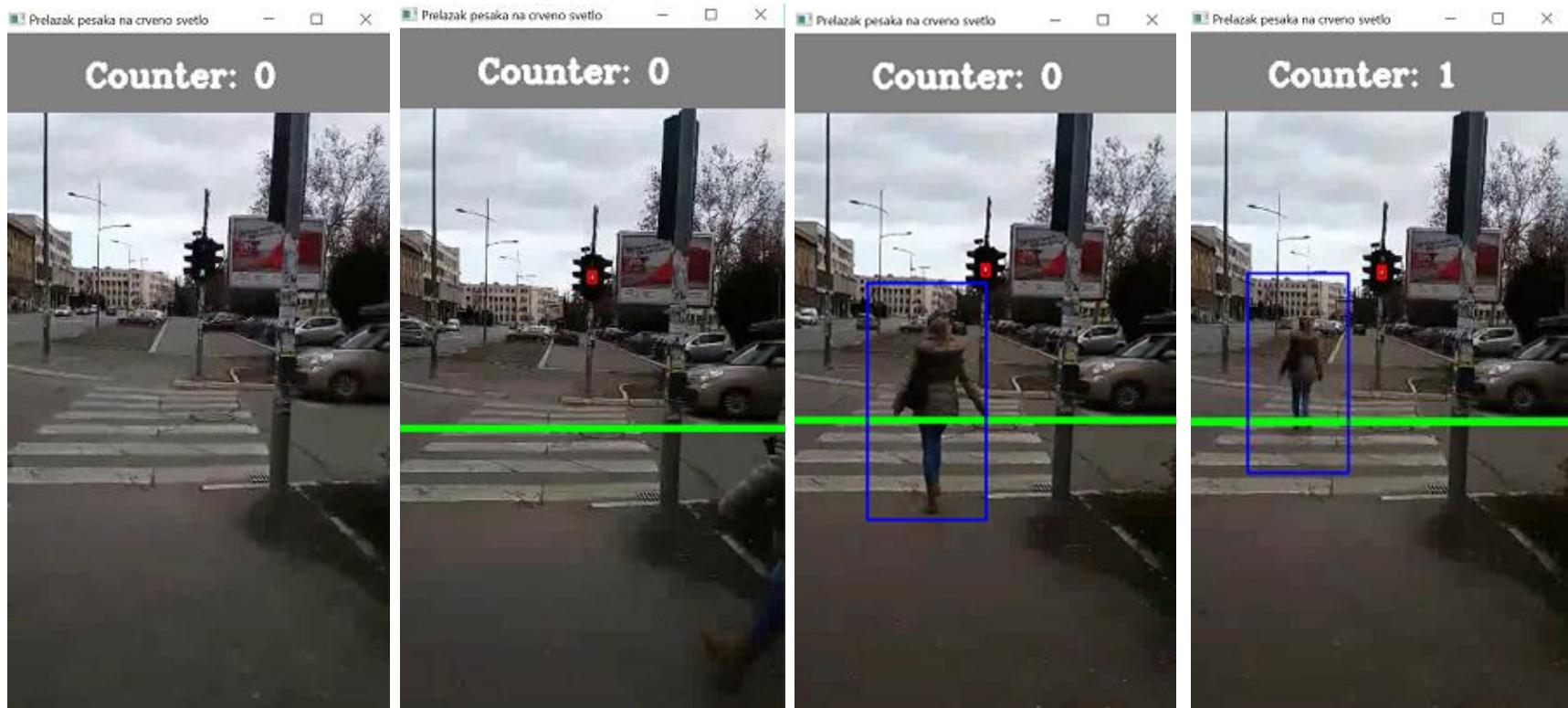
- Problem: snimci nisu napravljeni pomoću stacionarne kamere već ručno putem mobilnog telefona
  - Sve je prepoznavano kao pokret
- Zbog toga je umesto MOG primenjen *Haar Cascade Classifiers*
  - Metod detekcije objekata zasnovan na mašinskom učenju
  - Trening skup: pozitivne slike (koje sadrže pešake) i negativne slike (koje ne sadrže pešake)
  - U zadovoljavajućoj meri je prepoznato kretanje pešaka uz korišćenje klasifikatora za prepoznavanje celog tela
  - [http://docs.opencv.org/3.1.0/d7/d8b/tutorial\\_py\\_face\\_detection.html](http://docs.opencv.org/3.1.0/d7/d8b/tutorial_py_face_detection.html)

# Modul za prebrojavanje pešaka

- Za ovaj korak je upotrebljena izračunata jednačina prave sredine pešakog prelaza i pozicija prepoznatog pešaka
- Kada pozicija pešaka predstavlja deo jednačine prave, to bi značilo da je pešak prešao sredinu puta i stoga je brojač za njihovo prebrojavanje uvećan



# Verifikacija



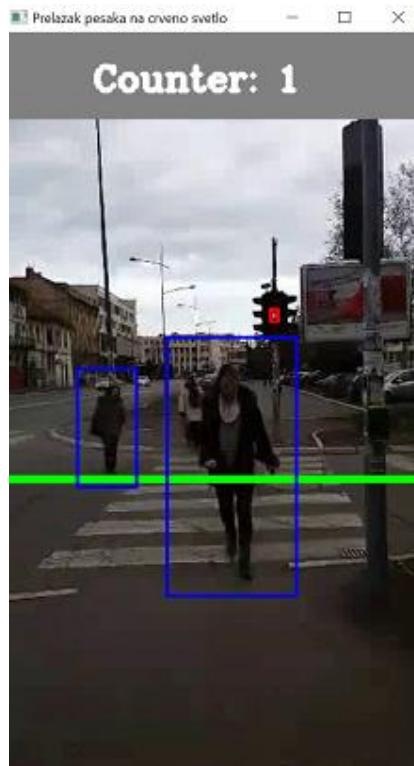
# Nedostaci u rešavanju ovog problema

---

- Pozicija kamere
  - Semafor ne sme biti previše blizu niti daleko (neće se videti svi pešaci)
  - Kamera ne sme biti suviše visoko zbog klasifikatora koji prepoznaju telo pešaka
- Stacionarnost kamere
  - Stacionarna kamera bi znatno pobojšala rezultate
  - Pomogli bi i algoritmi i tehnike za dodatnu stabilizaciju video snimka
- Osvetljenje
  - Različito ponašanje u zavisnosti od doba dana
  - Različiti uslovi za sunčano i kišovito vreme
  - Primena algoritama i tehnika za automatsko prilagođavanje u zavisnosti od uslova osvetljenosti

# Nedostaci u rešavanju ovog problema

- Grupa pešaka
  - Ako mnogo pešaka odjednom prelazi na crveno, pogoršala bi se tačnost prepoznavanja pešaka
- Smer kretanja pešaka
  - Pretpostavka sistema je da se pešaci kreću ortogonalno u odnosu na smer ulice
- Pešak iza pešaka



Prepoznata su dva umesto tri pešaka

# Sistem za segmentaciju i evaluaciju mamografskih snimaka

- Ivan Radosavljević i Aleksandra Mitrović
- Materijali prezentovani ovde preuzeti su iz diplomskog rada autora Ivana Radosavljevića

# Rešavani problem i motivacija

---

- Mamografija je neinvazivna metoda pregleda dojke koja se vrši u cilju ranog otkrivanja kancera dojke
- Pregled se vrši vizuelnom inspekcijom rendgenskog snimka dojke načinjenog pomoću mamografa
- Inspekciju mamograma, odnosno sam pregled, vrše posebno obučeni radiolozi

# Rešavani problem i motivacija

---

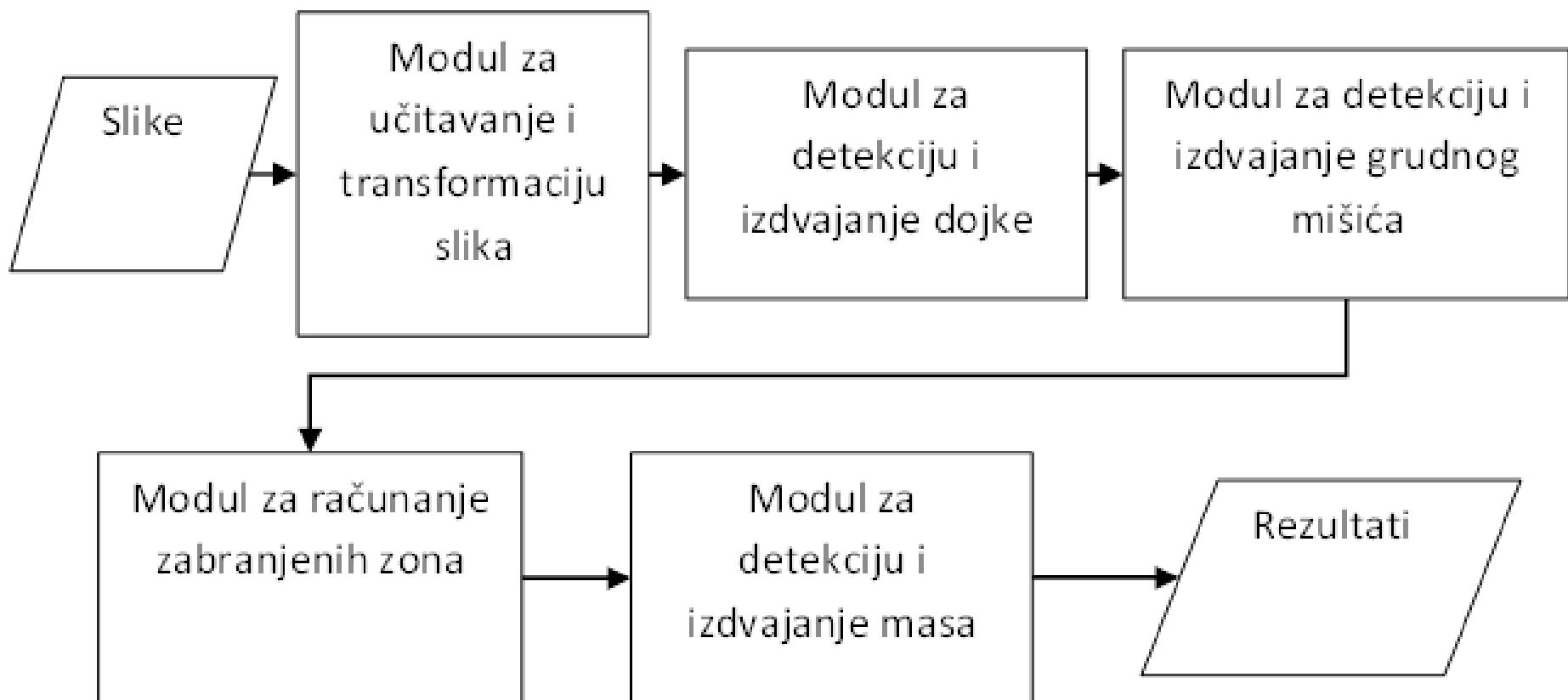
- Proces snimanja i pregleda dobijenog snimka uobičajeno traje do sat vremena
  - Ovo može biti problematično u slučaju kada je potrebno u kratkom vremenskom roku pregledati mnogo pacijenata
- Cilj: omogućiti automatsku detekciju svih potencijalno zločudnih efekata i za svaki od njih dati procenu rizika

# Metod rešavanja

---

- Za procenu rizika potencijalno zločudnih defekata ne postoji egzaktan matematički model
- Neophodno je modelovati rad radiologa koji u toku davanja procena defekte najčešće opisuje kvalitativno
  - npr., veoma malo, veliko, gusto, vrlo blizu grudnog mišića, daleko od ruba dojke itd.
- Zbog ove osobine procena rizika će biti vršena pomoću fazi kontrolera, kojima je ove nepreciznosti i neodređenosti moguće eksploratisati

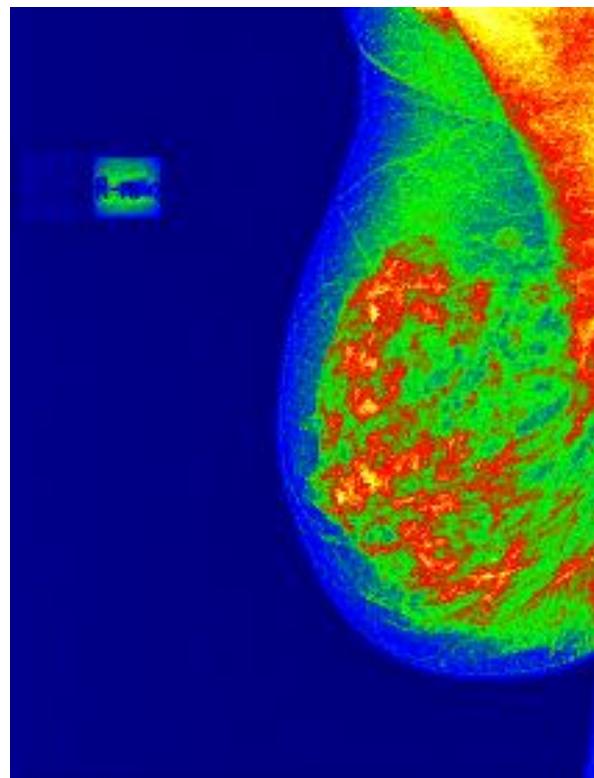
# Konceptualni model rešenja



# Modul za učitavanje i transformaciju slika

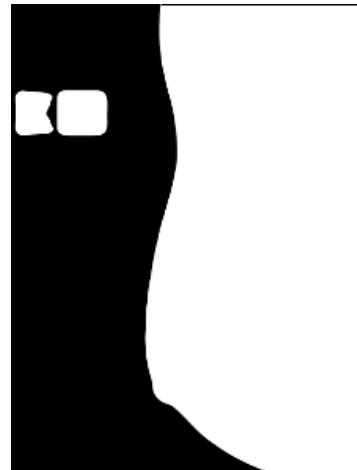
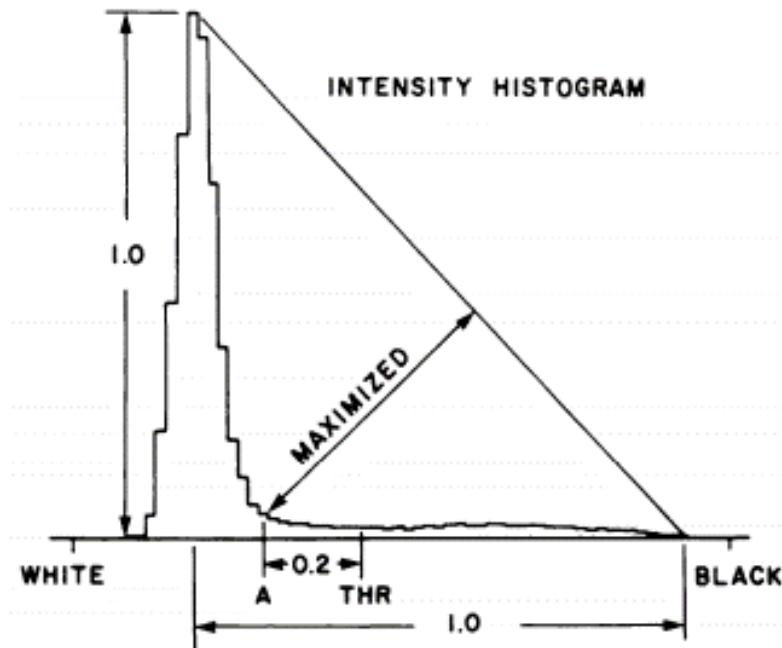
---

- Uklanjanje šuma postavkom minimalnog i maksimalnog praga
- Prikaz slike pomoću toplotne mape

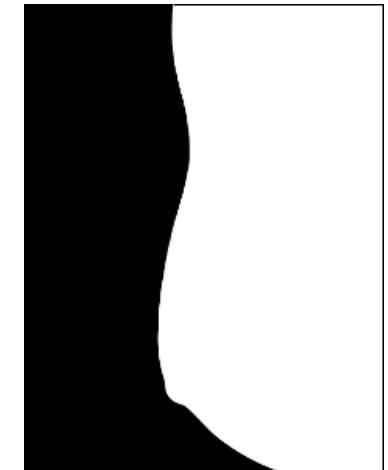


# Modul za detekciju i izdvajanje dojke

- Uklonjen šum visoke frekvencije upotrebom Gausovog filtera
- Određivanje praga na osnovu histograma



Rezultat

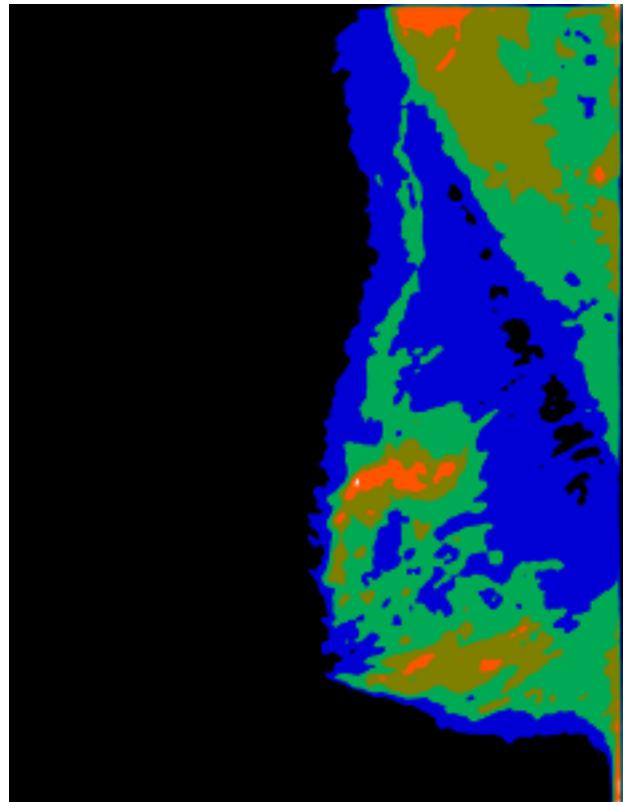


Pronalaženje  
povezanih  
regiona i  
selekcija  
najvećeg

W. E. Rogers, G. W. Zack i S. A. Latt, „Automatic measurement of sister chromatid exchange frequency,” *The Jurnal of Histochemistry and Cytochemistry*, t. 25, br. 7, pp. 741-753, 1977

# Modul za detekciju i izdvajanje grudnog mišića

1. Uklanjanje šuma visokih frekvencija upotrebom Gausovog filtera
2. Binarna slika iz prethodnog koraka se primenjuje kao maska
3. Od dobijene slike formira se histogram intenziteta piksela
  - Histogram se deli na 8 jednakih delova
4. Za svaki piksel originalne slike se proverava kojoj particiji histograma pripada i dodeljuje mu se vrednost intenziteta koja odgovara rednom broju particije
  - Efekat je da smo regije sličnih intenziteta grupisali u velike regije istog intenziteta
  - Granice između ovih regija su oštре



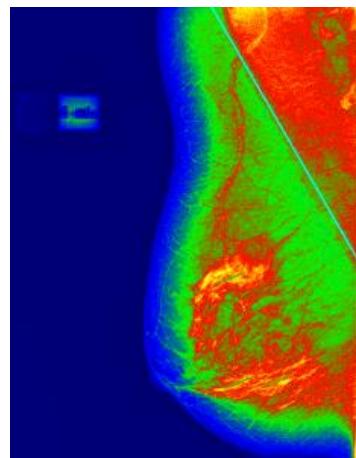
# Modul za detekciju i izdvajanje grudnog mišića

## 5. Detekcija ivica (Sobel)



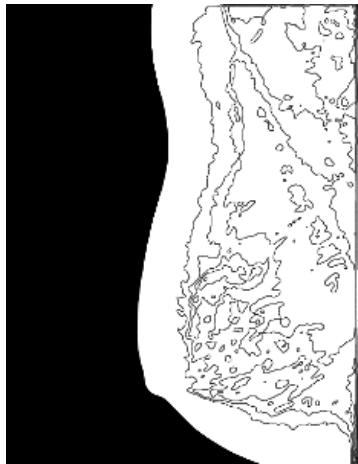
- Izdvojene su sve ivice, a mi želimo samo ivice grudnog mišića. Rešenje: odrediti kriterijum na osnovu kojih se tražena ivica izdvaja od ostalih
- Traženu ivicu moguće približno opisati kao najdužu duž koja se nalazi pod odgovarajućim uglom

## 6. Hough transformacija – pronalaženje tražene linije



# Modul za detekciju i izdvajanje grudnog mišića

## 7. Izdvajanje regionala od kojih se sastoji grudni mišić



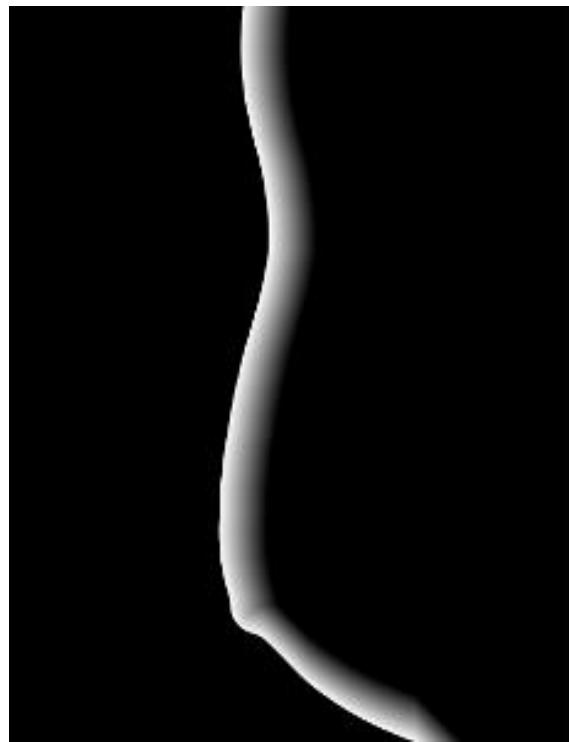
- Od binarne slike dojke oduzima se binarna slika ivica
- Sa ove slike se izdvajaju regioni
- Za svaki region se proverava da li je sa dobre strane prave koja definiše mišić
- Primjenjuje se dilacija da se ukone ivice sa ove slike



Rezultat

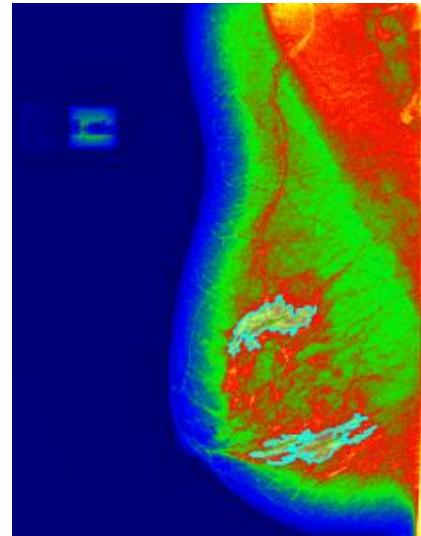
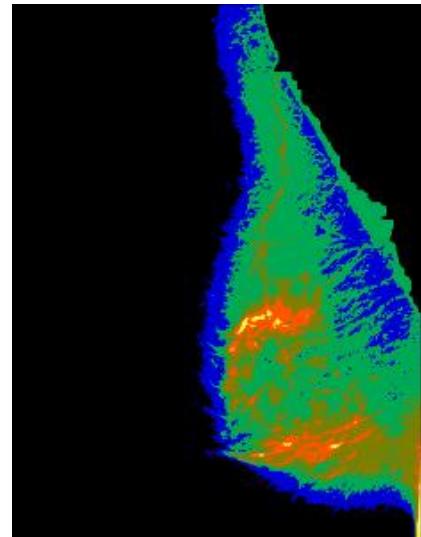
# Modul za računanje zabranjenih zona

- Zabranjene zone: tačke koje su od spoljašnje ivice mišića ili od unutrašnje ivice dojke udaljene najviše 1,5cm
  - Procenjeno je da je ovo približno 300 piksela



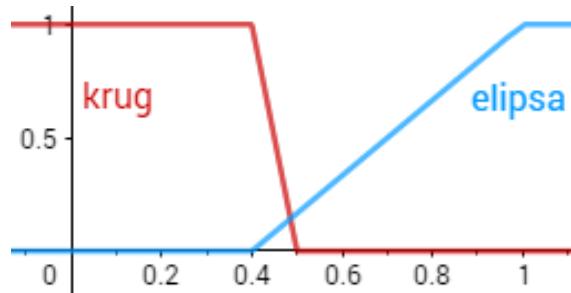
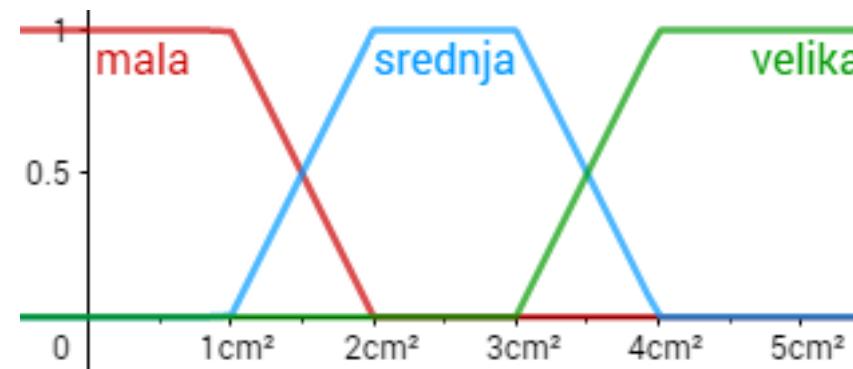
# Modul za detekciju i izdvajanje masa

1. Uklanjanje šuma Gausovim fiterom
2. Primenom maski dojke i grudnog mišića se izdvaja regija od interesa
3. Regioni sličnih intenziteta grupišu se u 8 regionala istog intenziteta
4. Segmentacija na osnovu fiksnog praga
  - Pikseli veći od 4 se smatraju gustim tkivom – ove regije se izdvajaju sa snimka
5. Za svaki region se računaju parametri od interesa
  - Površina, centar mase, udaljenost od ivice dojke,...



# Obrađivač rezultata

- Programski jezik *Python*
- *Fuzzy Logic Controller Toolkit (FLCT)*



# Provera tačnosti

---

- Vizuelna inspekcija
  - Segmentacija je ispravna ako su sa snimka ispravno izdvojene sve regije od interesa (ROI)
  - Defekti: nedostajući delovi ROI, spajanje dve ROI u jednu, proglašavanje pogrešnih regija za ROI i nedetektovanje ROI
  - Pronađeno 6 (od ukupno 25) snimaka sa defektima
  - Najčešći problem su bile regije koje predstavljaju mase (uglavnom zbog gustine slične gustini pozadinskog tkiva)
- Provera tačnosti procene rizika
  - Sistem uglavnom precenjuje rizike

# Moguća pobjošanja

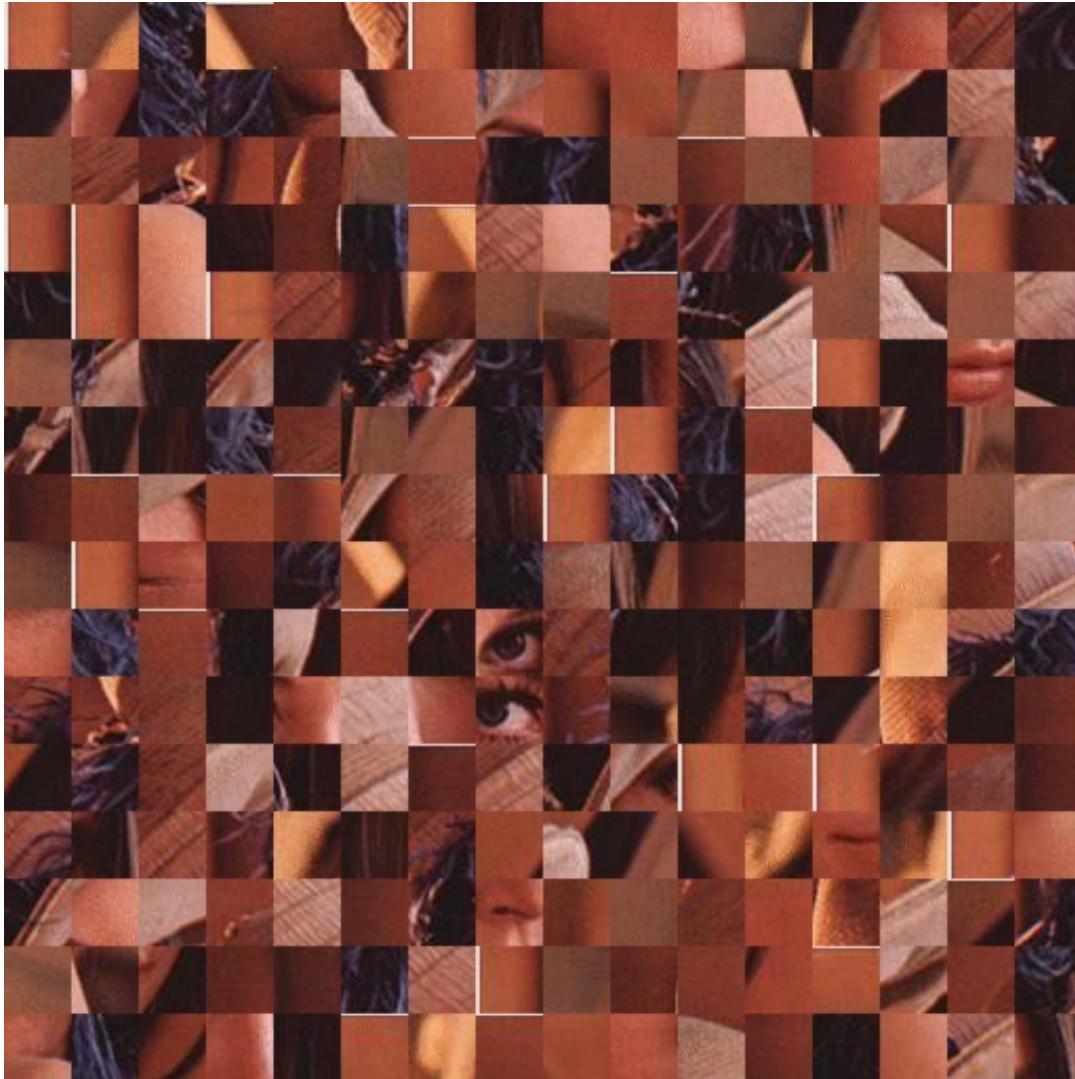
---

- Unapređivanje metoda segmentacije regija visoke gustine primenom metoda za izdvajanje blobova
- Unapređivanje metode za izdvajanje grudnog mišića tako da se umesto prave kao model ivice mišića koriste krive proizvoljnog stepena.
- Dodavanje novih ulaznih parametara za procenu rizika, npr.: razlika u gustini izdvojenih masa, razlika između prosečne gustine dojke i gustine izdvojene mase, itd.
- Uvođenje dodatnih pravila za procenu rizika

# Automatsko rešavanje puzli upotrebom genetskih algoritama

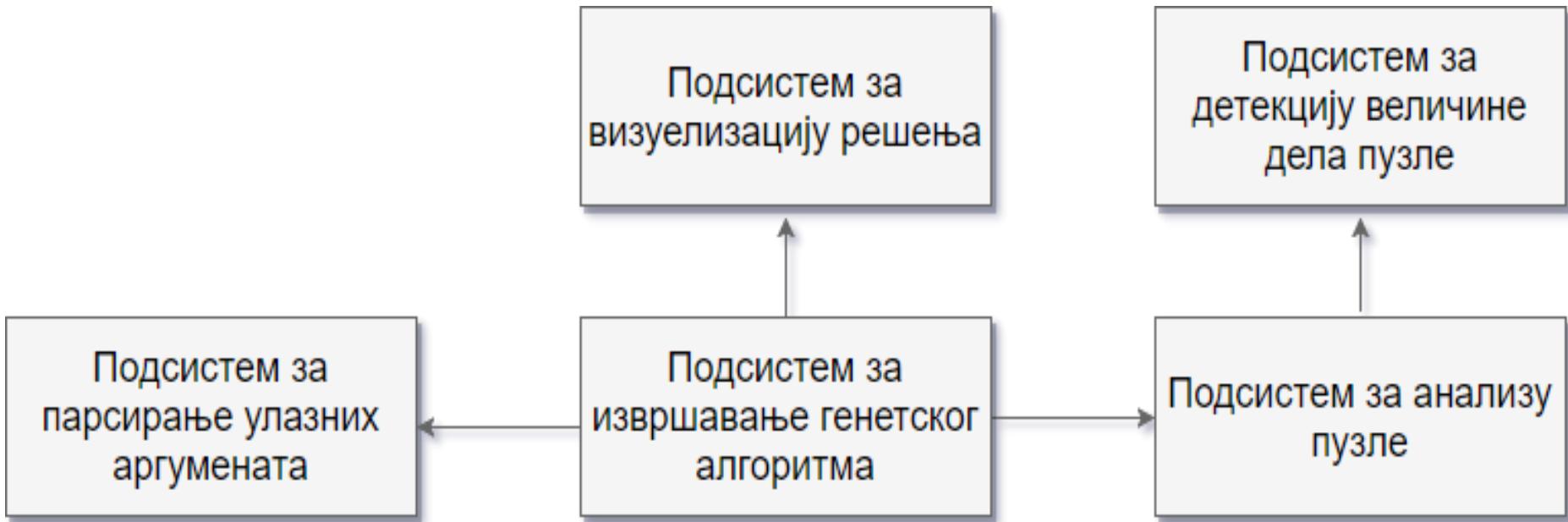
- Nemanja Milićević
- <https://github.com/ftn-ai-lab/sc-2016-e2/issues/87>
- Materijali prezentovani ovde preuzeti su iz diplomskog rada autora

# Problem



- Slike sa interneta, podeljene na delove. Delovi se izmešaju
- Ograničenje: delovi puzzle moraju biti kvadratnog oblika

# Metodologija



- Argumenti: putanja do slike koja predstavlja puzlu, parametri GA
- Analiza puzzle:
  - formira se *lookup* tabela sa metapodacima koja drastično ubrzava vreme izvršavanja GA
  - Detektuje se veličina dela puzzle
- Nakon svake generacije GA, najbolje rešenje se vizualizuje

# Detekcija veličine dela puzzle

---

- Obrada slike
  - Slika se transformiše u *grayscale*
  - Vrši se segmentacija pomoću globalnog praga kako bi se dobila binarna slika (brza segmentacija, nezavisna od regiona)
  - Uklanjanje šuma: dilacija i erozija
- Izdvajanje kontura
  - *Border flowing* algoritam
    - S. Sazuki / K. Abe, *Topological Structural Analysis of Digitized Binary Images by Border Following*, Shizouka University, Japan, 1985.
  - Filtriranje kontura prema graničnom pravougaoniku i površini – traže se konture slične kvadratu
- Pod pretpostavkom da su delovi puzzle kvadratnog oblika (uz poznavanje dimenzija slike) znamo moguće velicine delova puzzle
  - Kao rezultat se vraća vrednost za koju imamo najviše kontura-kandidata

# *Fitness funkcija*

---

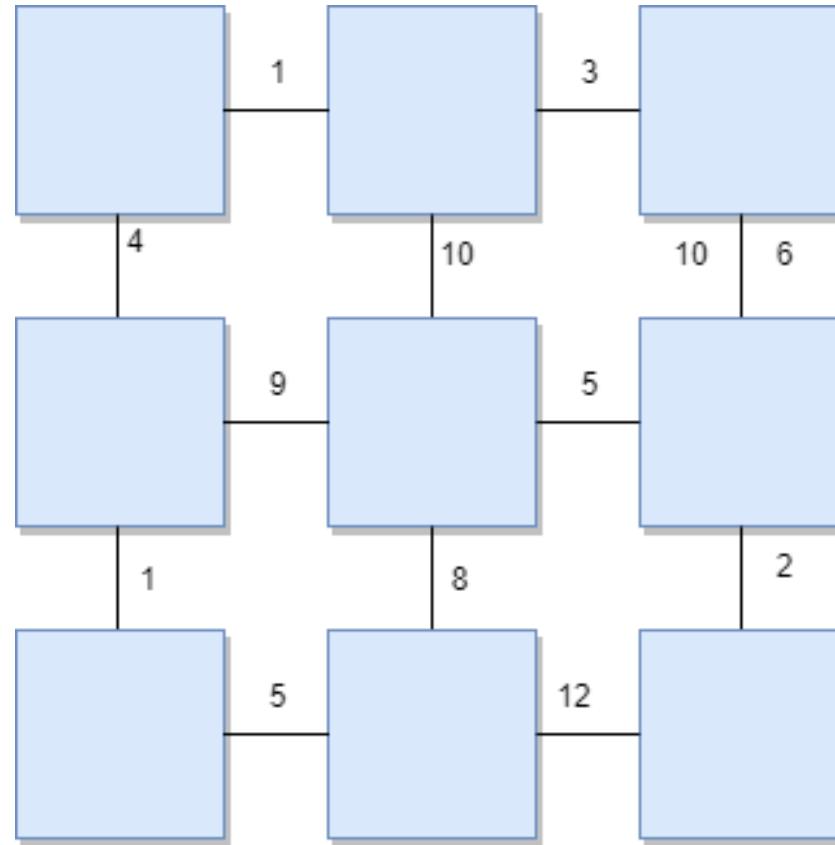
- Mera kompatibilnosti susednih delova puzzle
  - Pretpostavka: susedni delovi u originalnoj slici dele slične boje duž zajedničke stranice
- Računanje *fitness*-a mora da bude računarski jeftino jer se ovo vrši veliki broj puta
- *Lookup* tabela
  - Čuvaju se već izračunate mere različitosti za sve parove delova puzli
  - Za svaki deo se čuvaju podaci o njemu najkompatibilnijim delovima za svaku geometrijsku konfiguraciju (iznad, pored)
  - Ogromno ubrzanje rada sistema

# Crossover

---

- Problem: treba da se preuzmu dobre osobine od oba roditelja
- Operator dodaje delove puzzle deo po deo
  - Novi delovi se mogu dodati jedino ako su susedni već postojeći
  - Zaustavljamo se kada su dodati svi delovi

# Crossover



- Puzla se posmatra kao neusmereni težinski graf. Težine grana: mera različitosti susednih delova

# Crossover

---

- Inspiracija: Primov algoritam
  - Algoritam počinje od jednog čvora i povećava stablo, čvor po čvor, birajući u svakom koraku granu najmanje težine
  - Cilj: ispoštovati ograničenja, a formirati stablo koje je bolje rešenje od jedinki roditelja
- U ovom rešenju: umesto prostog biranja grane sa minimalnom težinom, u obzir se uzimaju i roditelji
  - 1) Bira se grana koja se nalazi u oba roditelja (ako postoji)
  - 2) Ako se takva grana ne može naći, bira se *best-buddy* grana
    - Svaki deo smatra drugi deo kao sebi najkompatibilniji
  - 3) Ako nema *best-buddy* grane, bira se grana sa minimalnom težinom kao u Primovom algoritmu

# Pravci za unapređenje

---

- Podržati drugačije veličine i oblike delova puzli
- Paralelizacija rešenja

# Još dobrih projekata generacije 2016

## Prepoznavanje emocije sa slike lica

- Diplomski rad
- <https://github.com/ftn-ai-lab/sc-2016-e2/issues/33>



## Upravljanje video igrom tehnikama kompjuterske vizije

- Diplomski rad
- <https://github.com/ftn-ai-lab/sc-2016-e2/issues/8>
- Plugin za *Unity* okvir koji omogućava da se postojeća igra proširi tako da se njom može upravljati pokretima tela u prostoru ispred kamere računara



# Još dobrih projekata generacije 2016

---

- Ima još mnogo dobrih projekata koji nisu stali u ovu prezentaciju, ali vredi da ih pogledate. Svaki od njih je opisan i posterom:
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/38>
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/25>
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/12>
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/59>
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/8>
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/47>
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/18>
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/28>
  - [https://gitlab.com/jelena-b94/Movie Recommendation System/](https://gitlab.com/jelena-b94/Movie%20Recommendation%20System/)
  - <https://github.com/ftn-ai-lab/sc-2016-e2/issues/31>

# Detekcija i prepoznavanje saobraćajnih znakova upotrebom CNN

- Helena Zečević, Luka Maletin i Aleksandar Nikolić
- <https://github.com/ftn-ai-lab/sc-2017-siit/issues/9>
- Materijali prezentovani ovde preuzeti su iz diplomskog rada autora Luke Maletina, koji je doradio sistem u odnosu na originalni projekat

# Motivacija – sistemi za asistenciju pri vožnji

- Parking senzori
- Automatsko parkiranje
- Automatsko upravljanje vozilom
- Asistencija pri prestrojavanju
- Sistem za automatsko kočenje
- Sistem za navigaciju
- ...
- **Sistem za automatsko prepoznavanje saobraćajnih znakova [1]**



# Problem koji se rešava

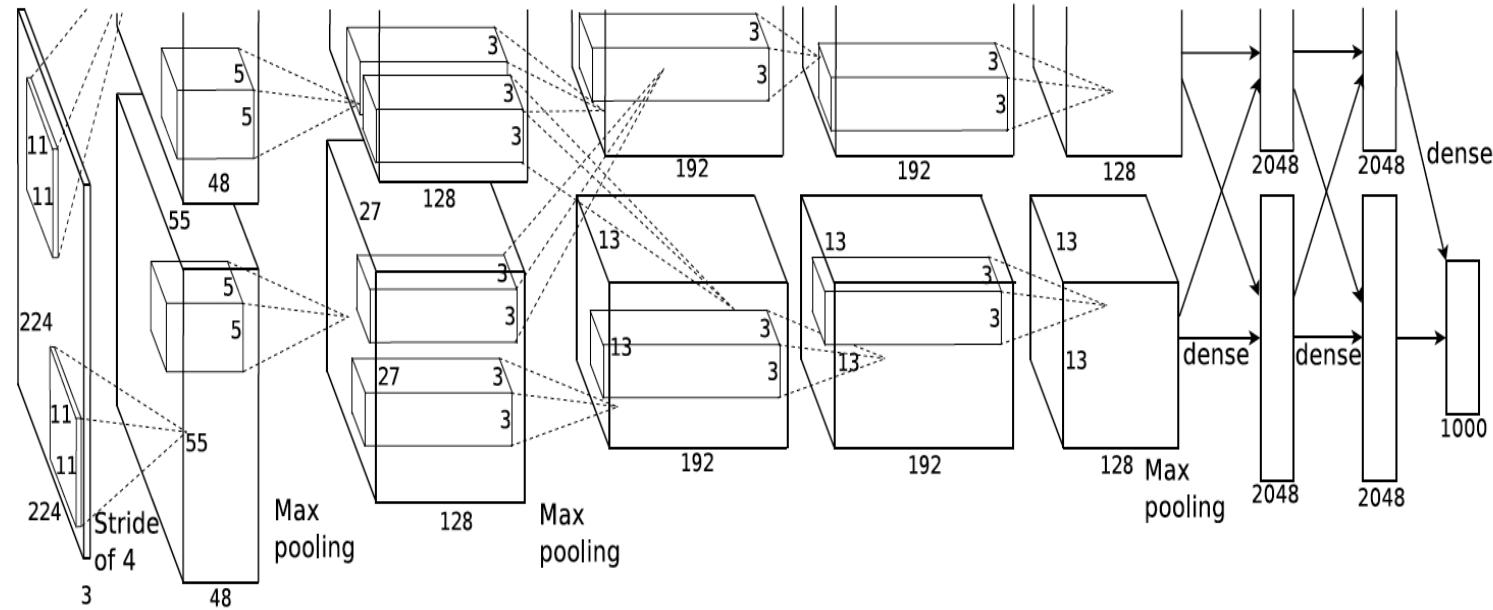
Analiza frejmova kamere u cilju:

- **Detekcije** saobraćajnih znakova
  - Procena da li se na frejmu nalazi saobraćajni znak/znaci
  - Gde se na frejmu nalaze znaci (lokalizacija)
- **Prepoznavanja** saobraćajnih znakova
  - Razvrstavanje pronađenih znakova u predefinisane kategorije



# Metodologija – CNN

- Model nadgledanog mašinskog učenja
- Zasnovan na veštačkim neuronskim mrežama
- Zadatak – klasifikacija slika



# Tok podataka u sistemu

- Ulaz je slika (a)
- Faza detekcije određuje *bounding box* svakog znaka (b)
- Faza prepoznavanja svrstava prosleđene znakove u klase (c), što predstavlja rezultat sistema



a)



b)

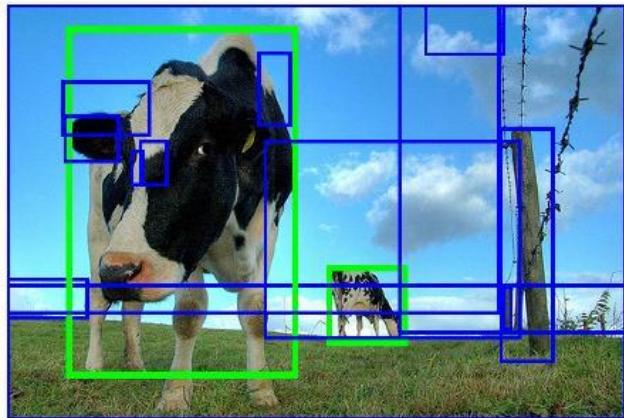
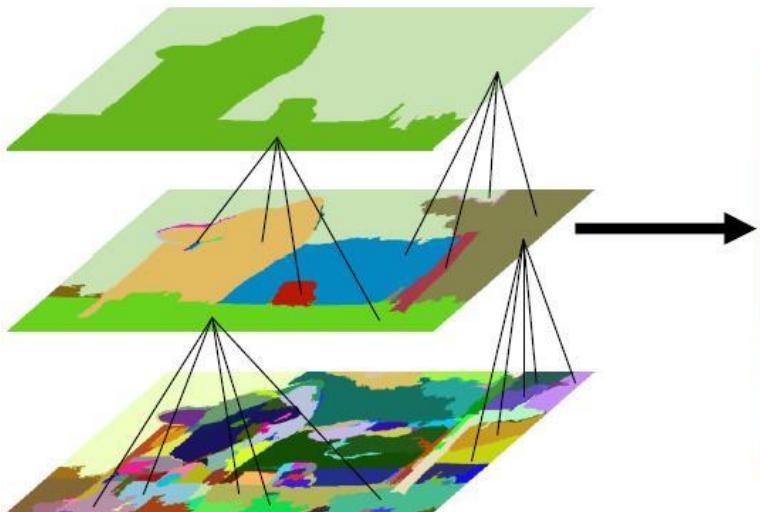
c)



Zabranjeno  
parkiranje      Uslovni  
stop

# Detekcija

1. Sliding window →
2. Region proposals ↘
3. R-CNN (*Regions with Convolutional Neural Network Features*) [2]
4. Fast R-CNN [3]
5. Faster R-CNN [4]



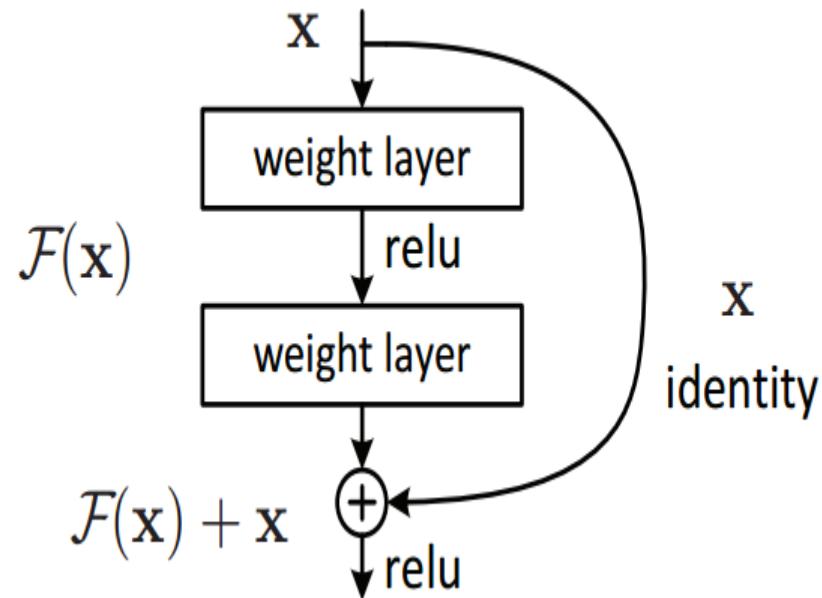
# Detekcija

---

- Za fazu detekciju korišćen je *Faster R-CNN* metod [4] sa konvolucionom mrežom *AlexNet* arhitekture
- *State-of-the-art* rešenje
  - Najbolji zvanični rezultat na *Pascal VOC* [8] skupu podataka iznosi 0.811 *mAP* uz korišćenje *Faster R-CNN* modela, ali sa naprednjom *ResNet* [5] arhitekturom
  - U rešenju predloženom u ovom radu, zbog hardverskih ograničenja, nije korišćena *ResNet* arhitektura za model za detekciju

# Prepoznavanje

- Residual Networks [5] (*state-of-the-art* za ovaj zadatak)
- ResNet block



# Skupovi podataka

***Belgium Traffic Sign Dataset [6]***  
**3479 trening, 1612 test**



***German Traffic Sign Recognition Benchmark [7]***  
**39209 trening, 12630 test**



# Implementacija

---

- *Microsoft Cognitive Toolkit (CNTK) open-source alat*
  - Seide, F., & Agarwal, A. (2016, August). CNTK: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 2135-2135). ACM

# Rešenje i rezultati

---

- Detekcija
  - Faster R-CNN sa AlexNet (5 conv)
  - 0.6756 mAP
- Prepoznavanje
  - Residual Network (1 conv + 9 residual blocks = 19 conv)
  - 96.192%
  - *State-of-the-art*: 99.71% (rešenje prikazano u ovom radu ima nešto manje performanse usled slabijih mogućnosti odabira hiperparametara zbog hardverskih ograničenja)

# Odabraná literatura

---

1. Fu, M.Y. and Huang, Y.S., 2010, July. A survey of traffic sign recognition. In Wavelet Analysis and Pattern Recognition (ICWAPR), 2010 International Conference on (pp. 119-124). IEEE.
2. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
3. Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).
4. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).
5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
6. Timofte, R., Zimmermann, K., & Van Gool, L. (2014). Multi-view traffic sign detection, recognition, and 3D localisation. Machine vision and applications, 25(3), 633-647.
7. Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011, July). The German traffic sign recognition benchmark: a multi-class classification competition. In Neural Networks (IJCNN), The 2011 International Joint Conference on (pp. 1453-1460). IEEE.
8. Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1), 98-136.

# TDSC: Traffic Sign Detection and Classification

- Viktor Šanca
- <https://github.com/vsanca/TSDC>

# Skup podataka

---

- Skupovi podataka
  - [German Traffic Sign Recognition Benchmark – GTSRB](#)
    - 5000 znakova, 42 klase
  - [German Traffic Sign Detection Benchmark – GTSDDB](#)
    - 900 slika iz saobraćaja sa anotacijama znakova
- Augmentacija skupa podataka:
  - Na slike koje ne sadrže znakove (pozadinu) dodati znakove iz **GTSRB** skupa podataka
  - Pri tome varirati veličinu, stepen rotiranosti i osvetljenosti

# Metodologija

---

- Poređenje pristupa:
  - Sliding window + CNN
  - Haar kaskadni klasifikator
  - HOG + SVM
  - CNN koja se zasniva na [GoogLeNet-OverFeat](#) algoritmu
    - Optimalniji pristup u odnosu na sliding window za segmentaciju i detekciju objekata

# Detekcija i prepoznavanje brend logoa na slici

- Željko Drageljević, Milan Desančić, Vladimir Indjić
- <https://github.com/vladaindjic/brand-logo-detection>

# Problem

- Prepoznavanje brenda čiji je logo prikazan na slici
- Flickr Logos 27 dataset
  - Slike logoa 27 brendova (za svaki 30 slika)



Nekoliko slika iz skupa podataka  
koje sadrže logo brenda

- Slike lošijeg kvaliteta su ručno izdvojene i zamenjene slikama sa interneta
- Augmentacija: generisano je po 500 slika za svaki logo (rotiranje, isecanje, dodavanje šuma, *blur*,...)
- Pridodate su slike koje ne sadrže logoe
- Podela na skup za treniranje (70%), validaciju (10%) i testiranje (20%)



Nekoliko slika koje ne sadrže logo

# Metodologija

## 1. HOG + Neuronska mreža + *Sliding Window* + *Non-Maximum Supression* (NMS)

- Ukoliko je izdvojeno više karakterističnih regiona sa logoima, na njih se primenjuje NMS
- NMS – na osnovu površine preklapanja regiona i pouzdanosti predikcije postojanja logoa unutar njih, izdvajaju se najbolji regioni i oni predstavljaju rezultat detekcije

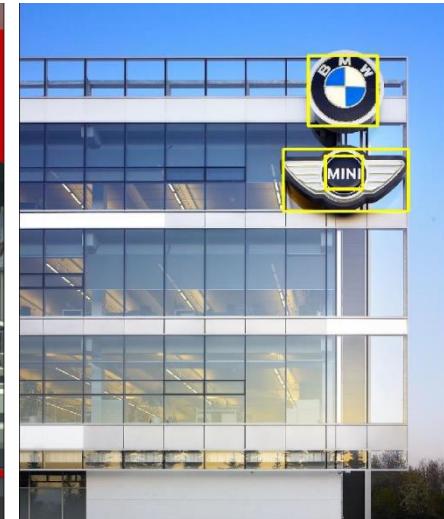


Uspešno detektovanje znaka FedEx, ali takođe i detektovanje logoa tamo gde on ne postoji

# Metodologija

## 2. CNN + Selective Search + Non-Maximum Suppression

- *Selective Search* – dostupna implementacija u *OpenCV*. Grupiše i izdvaja regione na slici koji su slični po određenim karakteristikama (boji, teksturi,...)
- Izdvojeni regioni se puštaju na ulaz CNN
- Ukoliko je izdvojeno više regiona sa logoima, na njih se primenjuje NMS



# Zaključak

---

- Oba pristupa bila su veoma uspešna pri rešavanju zadatka klasifikacije
  - NM: 96.7%, CNN: 99.8%
- Prvi metod je imao loše rezultate prilikom detekcije, pre svega zbog *sliding window* pristupa
  - Skup
  - Veličina prozora se mora predefinisati, a logoi imaju različit oblik i dimenzije
- Drugi metod je imao bolje rezultate prilikom detekcije
  - Pokazao je kako je smislena pretraga regiona važna za ovaj problem
- *State-of-the-art* za detekciju objekata: R-CNN, Fast R-CNN i Faster R-CNN

# Spajanje slika u panoramu

- Vojin Pešalj
- <https://github.com/ftn-ai-lab/sc-2017-siit/issues/18>

# Problem

---

- Spajanje više slika u panoramu
- Skup podataka:
  - <http://www.cvl.isy.liu.se/en/research/datasets/passta/>
  - Dodatne slike, slikane iz različitih uglova sa različitim osvetljenjem, kako se aplikacija ne bi testirala samo u savršenim uslovima
- Evaluacija:
  - Dobijena slika se poredi sa rezultatom *Panorama Stitcher* programa (postojeći program za spajanje slika)
  - Mera: procenat razlike među slikama

# Metodologija

- Slike se podele na dve grupe: levo i desno od centralne slike
- Pronalaženje ključnih tačaka dve slike koje treba spojiti
- Pronalaženje povezanosti ključnih tačaka dve slike korišćenjem  $k$ -NN algoritma
- Korišćenje homografije nad drugom slikom kako bi se njene ključne tačke poklopile sa onima sa prve



Spajanje dve slike u jednu:

- Traži se najveća površina koja u sebi ne sadrži pozadinu (crne piksele)
- Algoritam traži ivicu koja u sebi ima najviše crnih piksela i pomeri je za jedan piksel ka unutra
- Ovaj proces se ponavlja dok se sve ivice ne nalaze unutar slike

# Detekcija šahovskih figura i njihovog položaja na tabli i simulacija partije

- Đorđe Ilić, Nikola Đurđević, Mirjana Lakić
- <https://github.com/ftn-ai-lab/sc-2017-siit/issues/7>

# Problem

---

- Na slici detektovati šahovsku tablu
  - Odrediti koje se figure nalaze na kojim položajima na tabli
  - Nakon detekcije, aplikacija nudi izbor nekoliko različito obučenih AI-a, koji bi završili započetu partiju
- 
- Skup podataka kreiran od strane autora
    - Više različitih aparata, različiti uglovi

# Metod

---

- Detekcija ivica radi pronaleta table
  - NM za prepoznavanje figura na osnovu ivica i boje
  - Obuka AI: evolutivni algoritmi (sa različitim *fitness* funkcijama)
- 
- Problemi:
    - Kod problema pronaleta ivica, problem je prevelika sličnost boje figura i polja na tabli – rešeno korišćenjem table sa braon i krem poljima
    - Blic – lažne ivice – rešeno prelaskom u HLS model boja
    - Preklapanje figura – rešeno pažljivim slikanjem tako da se ivice ne preklapaju

# Comic reader

(Prepoznavanje lica i teksta u stripu Alan Ford)

- Vuko Jovičić
- <https://github.com/vukojovicic/comic-reader>

# Problem

---

- Napraviti algoritam koji će biti u mogućnosti da čita stripove:
  1. Prepoznavanje teksta iz oblačića
  2. Prepoznavanje glavnih likova (4 glavna lika)
- Nastavak rada na ovom projektu:
  - Pripisivanje oblačića likovima
  - *Text-to-speech* sistem koji izgovara pročitan tekst
  - „Oživljavanje“ junaka koji govori na neki način (da bi se znalo koji junak je taj koji govori)

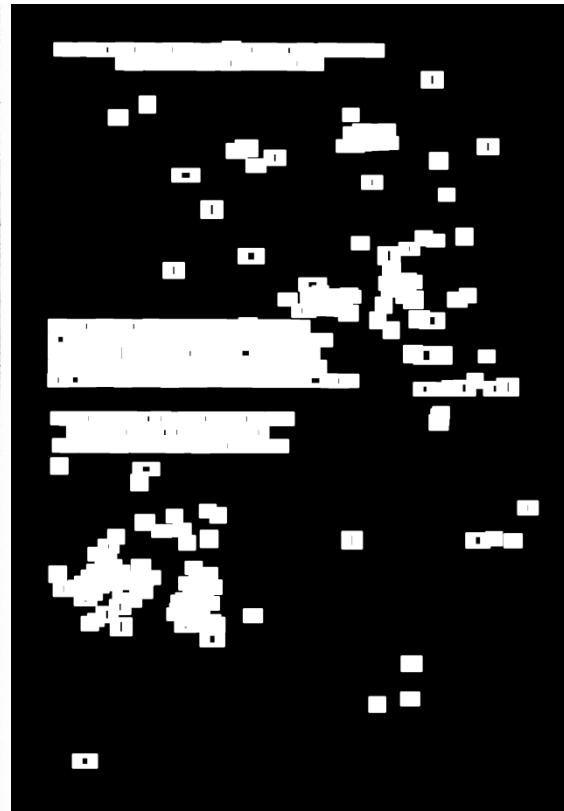
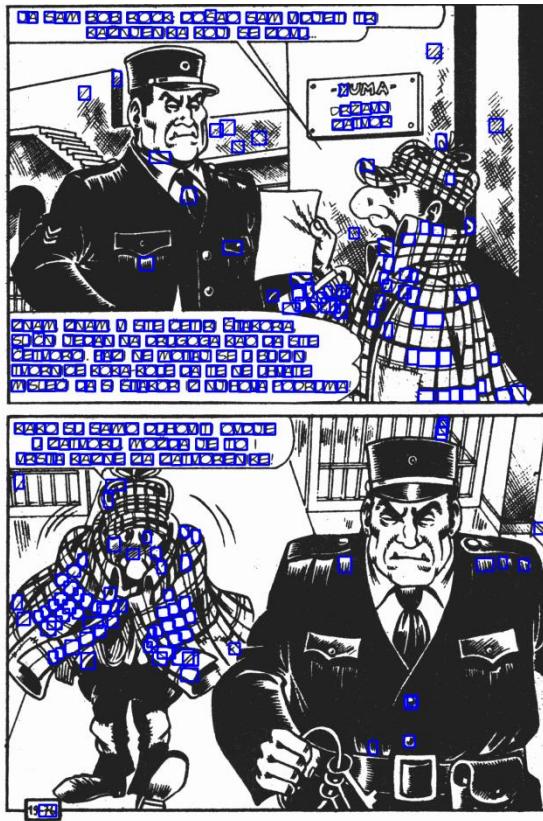
# Skup podataka

- 50 brojeva sa po ~120 stranica strip-a



# Prepoznavanje teksta u stripu

- Prvi korak: pronaći skupove slova koji zajedno čine jedan oblačić
- Pronalaze se konture čije dimenzije odgovaraju dimenzijama slova
- Da bi se izdvojena slova stopila u jednu konturu:
  - Izdvojeni su plavi pikseli sa slike, pretvoreni u bele i postavljeni na crnu pozadinu
  - Izvršena je dilacija po horizontalnoj osi u cilju gubljenja razmaka između reči i dobijanja „oblačića“



# Prepoznavanje teksta u stripu

- Prilikom pronalaženja „oblačića“ ostao je problem sa lažnim konturama čije dimenzije zadovoljavaju uslove da budu prihvачene kao „oblačić“. Dodate su još dve provere:
  1. Da li je procenat belih piksela u konturi dovoljno velik
  2. Da li je ugao pod kojim se nalazi kontura  $2k\pi$
- Slika:
  - Stvarne konture su zaokružene zelenom bojom
  - Lažne konture (koje bi bez dodatnih provera bile prepoznate) su zaokružene crvenom bojom



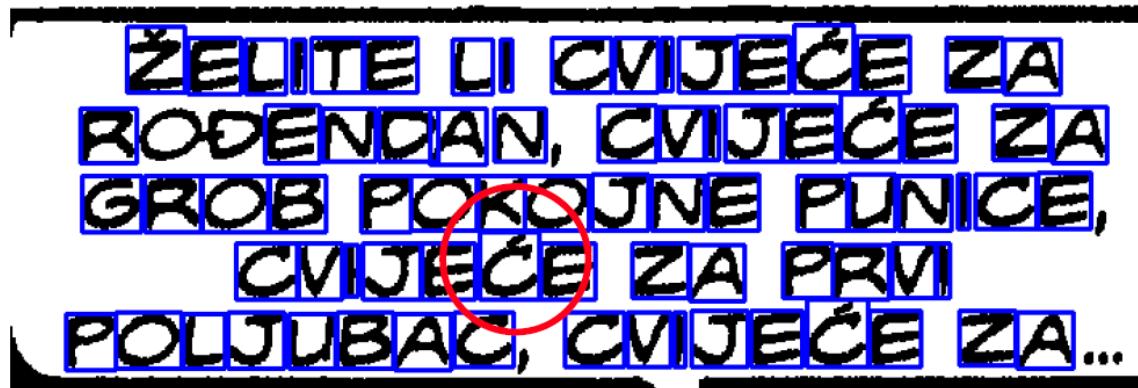
# Prepoznavanje teksta u stripu

- Nakon što su „oblačići“ uspešno izdvojeni, bilo je potrebno sortirati ih (sa leva na desno i od gore na dole)
  - Problem: u nekim slučajevima se uhvati i neželjena kontura koja oteža sortiranje



- Rešenje: *DBScan* algoritam za klasterovanje koji izdvaja slova. Klaster mora:
  1. Sadržati dovoljno veliki broj kontura
  2. Konture moraju međusobno biti dovoljno blizu
- Ovim su izdvojena samo slova, koja se mogu propustiti kroz NM

# Prepoznavanje teksta u stripu



Slika 7

Gornja granica slova Ć niža je od donje granice nekog od slova iz prethodnog reda



Slika 8

U okviru oblačića koji želimo da obradujemo našao se deo drugog oblačića zbog kojeg je nemoguće sortirati konture.

# Prepoznavanje lica

- *Yolo (You Only Look Once)* algoritam
  - Prethodnici ovog algoritma su za detektovanje objekta na slici morali svaki pojedinačni segment slike slati na predikciju u prethodno obučenu CNN i takav pristup je zahtevao veliki broj iteracija i veliki broj predikcija
  - Kod *Yolo*, slika se samo jednom provlači kroz CNN

Bob Rock



Br Jeden



Alan Ford



Šef



# Analiza grešaka modela

- Slova kod kojih dolazi do mešanja su veoma slična
  - č, č i c; š i s; o i 0
- Problem su slova pisana drugačijim fontom nad kojim algoritam nije treniran

---

I OVO BI TREBALO DA BUDE  
**GODIŠNJI ODMOR?**  
ČINI MI SE KAO DA SAM  
NA PRISILNOM RADU...



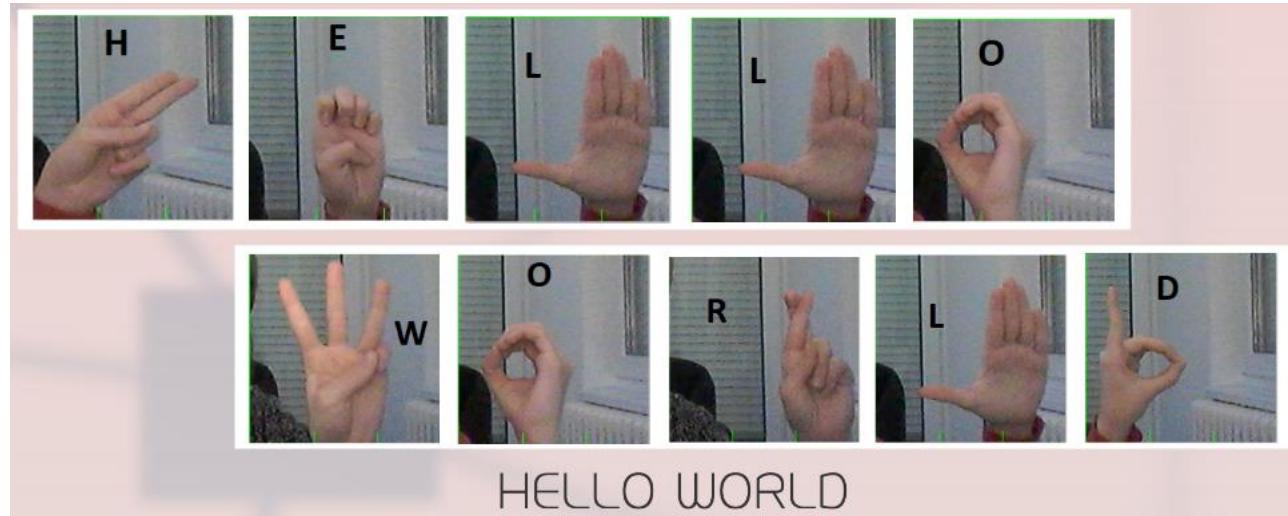
- Algoritam ne podržava prepoznavanje interpunkcijskih znakova

# Prepoznavanje znakovnog jezika za gluvoneme osobe sa slike

- Milana Bećejac
- <https://github.com/ftn-ai-lab/sc-2017-siit/issues/11>
- Materijali prezentovani ovde preuzeti su iz diplomskog rada Milane Bećejac

# Motivacija

- Gluvoneme osobe se sporazumevaju korišćenjem znakovnog jezika
- Problem se javlja prilikom komunikacije sa osobama koji ne poznaju znakovni jezik
- Sistem za prepoznavanje znakova može da reši ovaj problem



# Problem

---

- Postoje različite vrste znakovnih jezika
- Dve osnovne grupe:
  - Dinamički (gestovi)
  - Statički
- U ovom projektu:
  - Statički znakovni jezik *American Sign Language* (ASL)
  - Prepoznaju se slova američkog alfabet-a sa pojedinačne slike

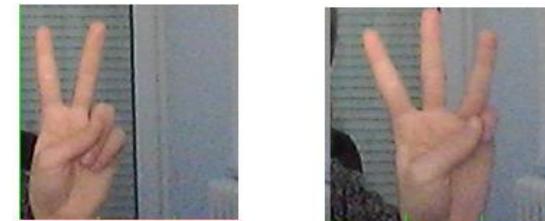
# Skup podataka

- 28 klase (A-Z), znak za razmak i neutralni znak (ne predstavlja ništa)
- Ručno kreiran od strane autorke
  - Šaka se postavi ispred kamere i snimi se određeni broj slika
- Preprocesiranje: RGB → HSV, primena pragova za izdvajanje kože od pozadine, erozija, dilacija, zamućivanje



# Metodologija

- Na konture šake (izdvojene u fazi preprocesiranja) se primenjuje CNN
  - Autorka je sama odredila arhitekturu i trenirala mrežu od početka
  - Tačnost 99.4%
- Potencijalni problemi:
  - Pozadina korisnika nije svetla ili je osvetljenje slabo
    - Možda moguće rešiti posvetljenjem fotografije
  - Šaka korisnika nije centrirana u predviđenom okviru
    - Dodati deo sistema za detekciju šake
  - Slični znaci za različita slova (slike)
  - Rotirana šaka
  - Boja kože



# Background subtraction

- Petar Radošević
- Materijali prezentovani ovde preuzeti su iz diplomskog rada Petra Radoševića

# Problem

---

- Uklanjanje pozadine sa slika na kojima se nalaze ljudski portreti
  - Sistem bi za ulaznu sliku trebao da izvrši klasifikaciju njenih piksela na piksele pozadine i piksele objekta od interesa
  - Rezultat rada sistema je maska sa označenom ljudskom figurom



# Metodologija

- CNN za klasifikaciju
  - vizuelizacijom *feature maps* skrivenih slojeva se mogu primetiti oblasti oko objekta koji se klasifikuju (slika u sredini)



- Neophodno je izbaciti *fully connected* slojeve sa kraja
- Na njihovo mesto se ubacuju *upsampling* slojevi čiji je zadatak rekonstrukcija rezolucije slike
- Rezultujuća segmentisana slika ima ublažene ivice (slika desno)

# Metodologija

---

- U radu je za CNN korišćena *DenseNet* mreža
- Obučena je nad COCO skupom podataka

# Rezultati – dobro segmentisane slike



# Rezultati – loše segmentisane slike



# Klasifikacija i autentifikacija otiska prsta

- Ivana Zeljković
- Materijali prezentovani ovde preuzeti su iz diplomskog rada Ivane Zeljković

# Motivacija

---

- Biometrički sistemi za autentifikaciju:
  - otisak prsta,
  - prepoznavanje lica
  - geometrija šake
  - prepoznavanje glasa
  - DNK...
- Većina današnjih softvera, u kojima je autentifikacija korisnika i bezbednost podataka od velikog značaja, se oslanja na biometričke sisteme koji koriste otisak prsta kao jedinstveno obeležje svake individue.

# Problem

---

- Formalna specifikacija i implementacija sistema za klasifikaciju i autentifikaciju otiska prsta.
- Implementirani sistem se sastoji iz dva podsistema:
  - Podsistem za klasifikaciju
    - Klasifikacija otiska u jednu od 5 mogućih klasa na osnovu njegove strukture
  - Podsistem za autentifikaciju
    - Izdvajanje i poređenje ključnih tačaka dva otiska u cilju utvrđivanja stepena preklapanja, tj. utvrđivanja da li pripadaju istoj osobi ili ne

# Podsistem za klasifikaciju

- Klase:
  - Luk
  - spirala
  - petlja na levo
  - petlja na desno
  - jeloviti luk
- Koraci:
  - Adaptivno izjednačavanje histograma ograničeno kontrastom (CLAHE) u cilju
    - poboljšanja lošeg kontrasta uzrokovanih visokim ili niskim stepenom osvetljenja
  - Rekonstrukcija otiska upotrebom Gabor filtera
    - Rašava probleme zašumljenih regiona, diskontinuitet u strukturi otiska,...
  - Klasifikacija otiska CNN (modifikacija *AlexNet* arhitekture)

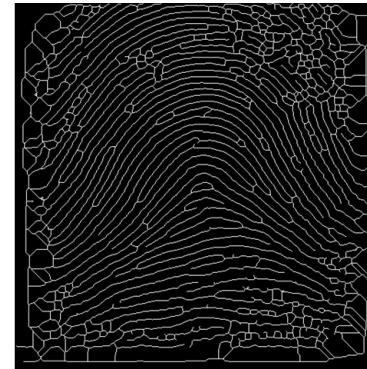


# Podsistem za autentifikaciju

- Skeletonizacija binarnog ulaza u podsistem u cilju pojednostavljivanja grebena strukture otiska



Binarna rekonstrukcija otiska



Skelet otiska

- Detekcija ključnih tačaka (završetaka i bifurkacija grebena) upotrebom algoritma *Crossing Number*
- Filtriranje skupa ključnih tačaka u cilju uklanjanja nerelevantnih
- Autentifikacija dva otiska, poređenjem skupova njihovih ključnih tačaka

# Skup podataka

- Skup podataka korišćen u sistemu predstavlja javno dostupnu bazu otiska NACIONALNOG INSTITUTA ZA STANDARDE I TEHNOLOGIJU (NIST)
- Sastoji se od 4000 instanci, od čega je 2000 korespondentnih parova otisaka napravljenih različitim metodama (elektronski skener, mastilo)
- Podela instanci po klasama je stratifikovana, što pomaže procesu treniranja klasifikatora
- Veoma varijabilan kvalitet slika



# Ostvareni rezultati – klasifikacija

---

- **Tačnost** klasifikacije je 40%.
- **Razlozi** smanjene tačnosti i predlozi za unapređenje:
  - Neadekvatna arhitektura neuronske mreže - obučavanje neuronske mreže VGG-19 kao klasifikatora
  - Nedovoljan skup faza preprocesiranja ulaza u neuronsku mrežu - redukcija zašumljenih regiona, izračunavanje orientacije i frekvencije grebena neposredno pre primene Gabor filtera
  - Nepotpuna rekonstrukcija otiska u slučajevima ulaza lošeg kvaliteta - povećanje broja filterskih kernela

# Nepotpuna rekonstrukcija otiska



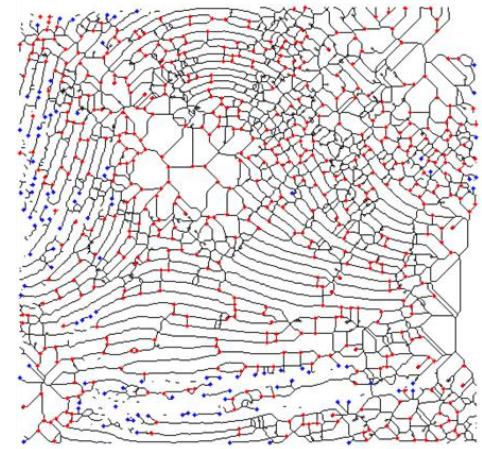
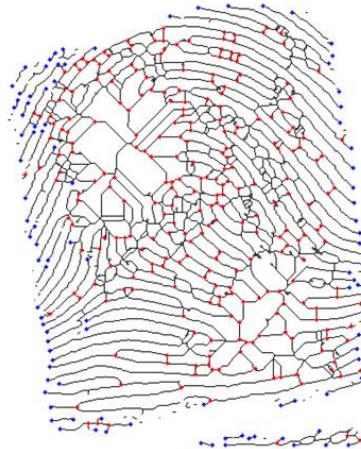
Originalni otisak sa zašumljenim  
segmentima



Loša rekonstrukcija uzrokovana šumom

# Poziciono i dimenzionalno odstupanje

- Dve različite instance identičnog otiska:
  - Razlike: veličina, orijentacija, pozicija
- Donje slike predstavljaju grafički prikaz detektovanih ključnih tačaka
- Rezultat preklapanja ključnih tačaka je 10%



# Ostvareni rezultati – autentifikacija

---

- **Tačnost** autentifikacije ostvarena u procesu poređenja identičnih instanci je **78%**, dok je u procesu poređenja različitih instanci **47%**.
- Suziti segment nad kojim se vrši skeletonizacija na centralni deo otiska (gde se nalazi najviše ključnih tačaka)
- Pobojšati algoritam provere preklapanja skupova ključnih tačaka: uzeti u obzir:
  - orijentaciju i lokaciju ključne tačke
  - frekvenciju susednih ključnih tačaka

# Finger drawing

- Nikola Kukavica
- <https://github.com/ftn-ai-lab/sc-2016-e2/issues/25>

# Problem

---

- Simulacija virtuelnog *Paint* programa
  - [https://www.youtube.com/watch?v=Z43\\_hCM74rU](https://www.youtube.com/watch?v=Z43_hCM74rU)
- Korisnik na osnovu ograničene paleta boja i predefinisanih gestikulacija vrši iscrtavanje



# Osnovna obrada slike

- Korinik nosi jednoboju crvenu rukavicu koja se izdvaja sa slike

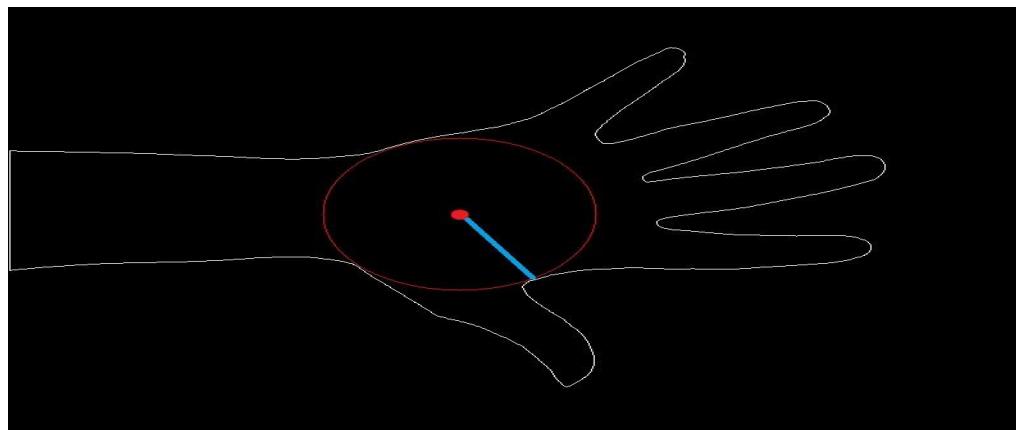
- Zamućivanje radi uklanjanja šuma
- RGB → HSV
- Erozija, dilacija



- Određivanje regiona od interesa
  - Na binarnoj slici su pronađene sve konture
  - Pronalaženje najveće i pronalazak *bounding box*

# Određivanje centra šake

- *Distance map*
  - Za svaki piksel objekta od interesa se određuje dužina do najbližeg crnog piksela
  - Centar šake je najveća površina uzastopnih belih piksela – koristimo *distance map* da pronađemo tačku u regionu koja je najviše udaljena od crnih piksela
  - Ako bismo nacrtali kružnicu oko te tačke sa poluprečnikom koje predstavlja rastojanje do najbliže crne tačke, sve tačke unutar kružnice su beli pikseli



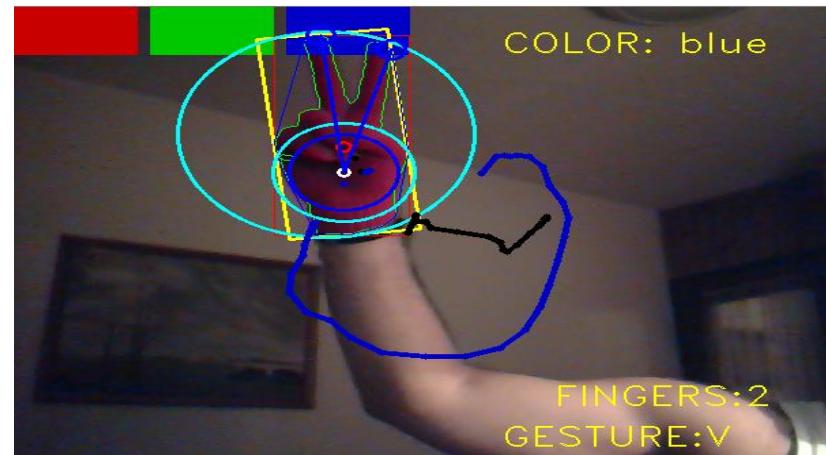
# Pronalaženje prstiju

---

- *ConvexHull* – funkcija koja vraća set tačaka koje određuju konveksnost objekta, odnosno, njegove ekstreme
  - Prsti imaju neravnine koje će se detektovati kao više tačaka na jednom prstu
  - Koristi se Euklidsko rastojanje između svake dve susedne tačke – ako je manje od zadatog praga, jedna od tačaka se eliminiše
  - Prilikom filtriranja, gleda se i rastojanje centra šake od ekstrema

# Prepoznavanje gestikulacija

- Tri stanja
  - Drawing mode – stisnuta šaka (nema ekstrema)
  - Click (slika)
    - Dva prsta ispružena, ostali skupljeni
    - Ispruženi prsti formiraju određeni ugao
  - None (ostalo)
- Jednostavan skup predefinisanih pravila

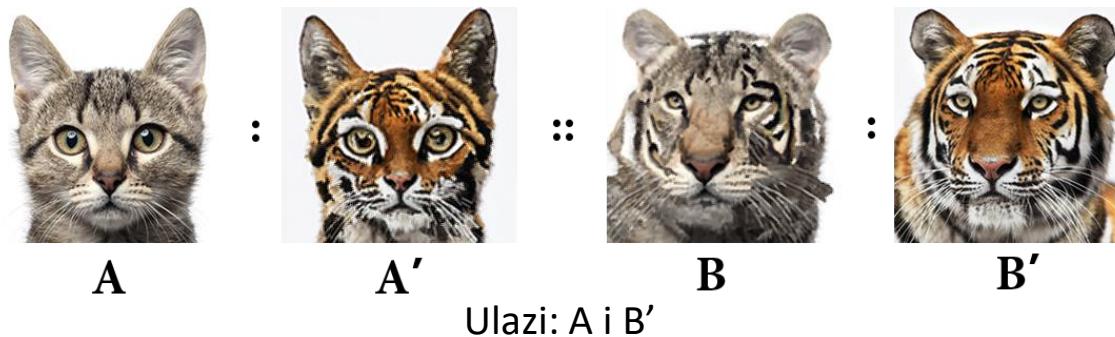


# Razmena vizualnih atributa posredstvom analogije slika (*Neural Style Transfer*)

- Michael Milinković
- Materijali prezentovani ovde preuzeti su iz diplomskog rada autora
- <https://prezi.com/view/UPga2XInloF3ISmNF8iY/>

# Rešavani problem i motivacija

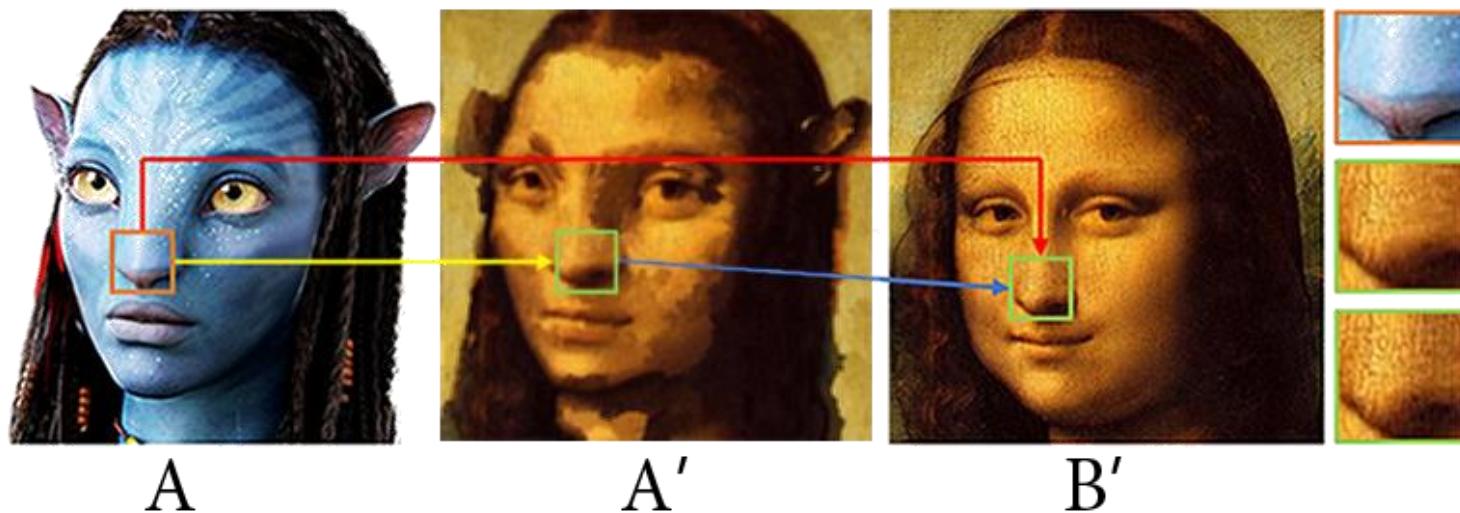
- Problem: razmena vizualnih atributa među slikama
  - Prenos vizualnih svojstava : boja, tekstura i stil
  - Slike mogu biti veoma različitog izgleda, ali slične semantičke strukture



- Popularnost u domenu društvenih mreža
- Komercijalne platforme
  - *Prisma* – aplikacija za uređivanje fotografija koja koristi neuronske mreže za transformisanje fotografija u umetničke slike

# Metodologija

- Potrebno je uspostaviti vezu između dve slike: pojava istih struktturnih karakteristika na različitim delovima obe slike
  - *Patch Match* [1]: **brzo** pronalaženje korespondencija između kvadratnih regiona slike



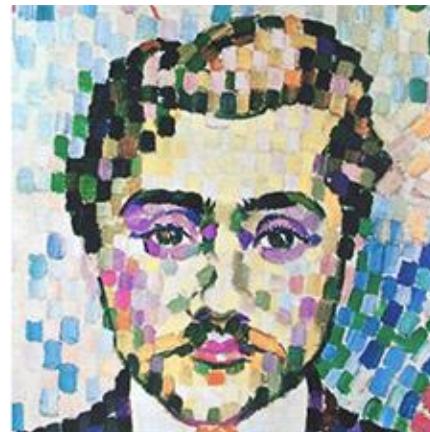
# Metodologija

- Analogija slika se ne primenjuje direktno na piksele ulaznih slika
- Koristi se CNN kako bi se konstruisao prostor obeležja
  - CNN postepeno enkodira informacije sa slike počevši od informacija niskog nivoa, sve do semantičkog sadržaja visokog nivoa
  - Koristi se trenirana VGG-19 arhitektura
- Analogije se formiraju u prostoru obeležja
  - “Duboke analogije slika” [2]
  - *Patch match* je prilagođen domenu CNN obeležja

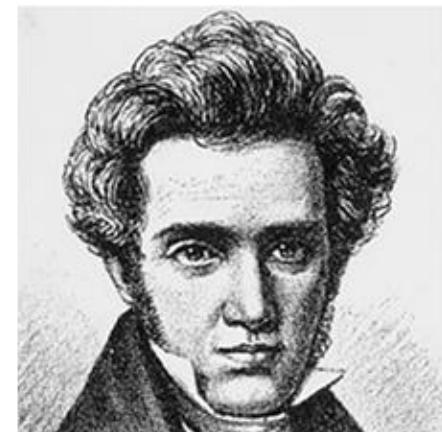


# Rezultati: stil – fotografija

- Očuvan strukturalni sadržaj fotografije (korespondencije očiju, nosa,...)
- Uspešan prenos značajnih karakteristika stila



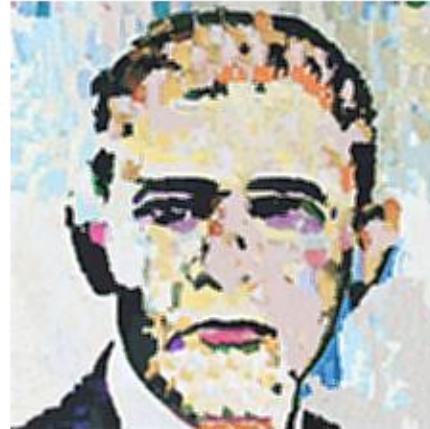
Улаз (узор)



Улаз (узор)



Улаз (извор)



Излаз



Излаз

# Poređenje sa *Pizma* rešenjem

- *Prisma* skoro u potpunosti očuvava strukturu ulazne slike, ali ne prenosi značajne informacije stila (osim boja i nijansi)
- Izlazi u ovom radu prave veći kompromis strukture i stila. Strukturalni odnosi su narušeni, ali sa ciljem uspešnog prenosa većeg broja vizuelnih svojstava



Улаз (извор)



Улаз (узор)



Улаз (узор)



Излаз



Излаз

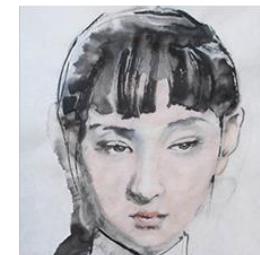


Излаз (*Prisma*)



Излаз (*Prisma*)

# Rezultati: stil-stil



A (улаz)

A' (излаz)

B (излаz)

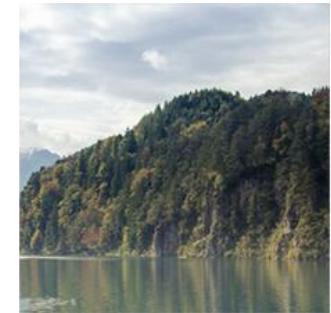
B' (улаz)

- Korespondencije između slika su većinom iskpravno određene i stil ispravno razmenjen
- Međutim, na 3. slici u 3. redu je došlo do gubitka informacija o strukturalnom sadržaju

# Rezultati: fotografija-stil



Улаз (узор)



Улаз (узор)



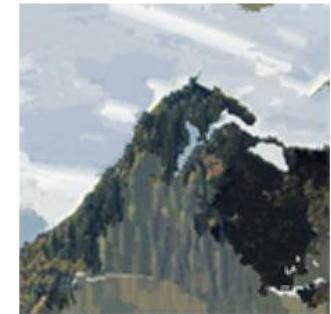
Улаз (извор)



Излаз



Улаз (извор)



Излаз

- Razmenu je teže ostvariti iz razloga što umetničkim delima često nedostaju detalji u korist kreativnosti i apstrakcije

# Razmena fotografija-fotografija



- Većina rezultata je dobra
- Jedina nepogodnost se može uočiti na 3. slici u 2. redu – loša struktura trave



A (улаz)

A' (излаz)

B (излаz)

B' (улаz)

# Ograničenja



Улаз (узор)



Улаз (извор)



Излаз 1



Излаз 2



Улаз (узор)



Улаз (извор)



Излаз

Prenos vizualnih svojstava je uspeo sa stila na fotografiju, ali u suprotnom smeru nije. Razlog za to jeste što umetničkim delima često nedostaju detalji u korist kreativnosti i apstrakcije

Algoritam koji pronalazi korespondencije između slika ne uspeva da pronađe određeni objekat (travu) sa slike izvora na slici uzora

# Literatura

---

1. Barnes, C., Shechtman, E., Finkelstein, A., & Goldman, D. B. (2009). PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3), 24.
2. Liao, J., Yao, Y., Yuan, L., Hua, G., & Kang, S. B. (2017). Visual attribute transfer through deep image analogy. arXiv preprint arXiv:1705.01088.

# Utvrđivanje istinitosti iskaza osobe sa video snimka

- Sara Perić
- Materijali prezentovani ovde preuzeti su iz diplomskog rada autora
- <https://github.com/ftn-ai-lab/sc-2017-siit/issues/20>

# Problem

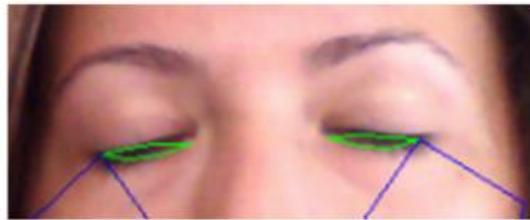
---

- Specifikacija i implementacija sistema za utvrđivanje istinitosti iskaza osobe sa video snimka ili kamere u realnom vremenu
- Tehnike kojima se koriste FBI agenti
  - Broj treptaja (5-6 uspešnih treptaja u stresnim situacijama, običajno 5-6 treptaja u minuti) – **Blink Detector**
  - Stiskanje usana – **Pursed Lips Detector**
  - Pojava rumenila u licu ispitanika – **Blushing Detector**
  - Učestalo skretanje pogleda
  - Preznojavanje
  - Odmahivanje glavom
  - Dodirivanje lica...

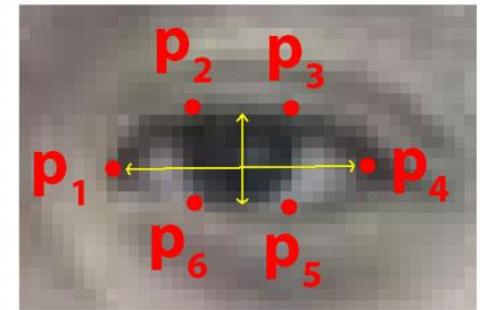
# Blink detector

- EAR vrednost – *eye aspect ratio*

Tereza Soukupova and Jan Cech, Real-Time Eye Blink Detection using Facial Landmarks, 2016, <http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>

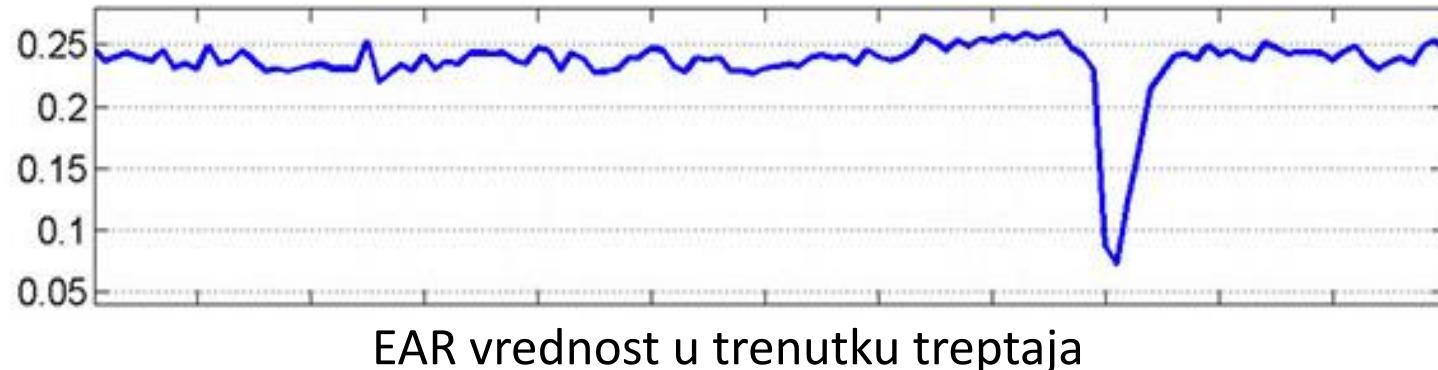


$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2 * \|p_1 - p_4\|}$$



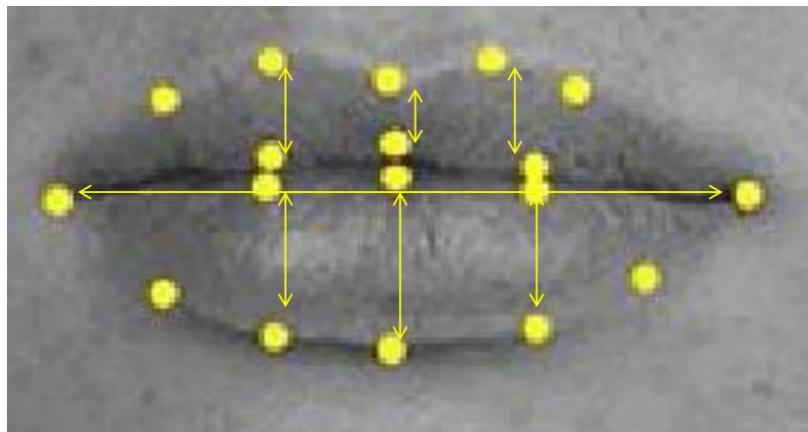
Prediktovane tačke očiju kada ih osoba drži zatvorenim. EAR vrednost iznosi 0.1230

→ Prag treptaja za svakog ispitanika

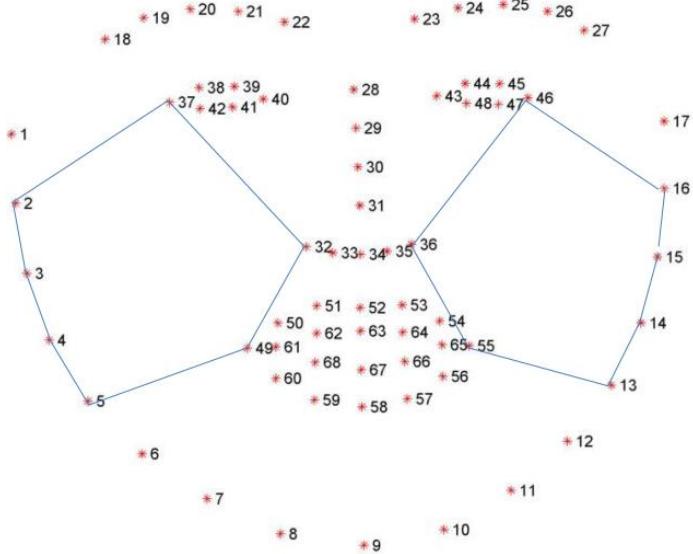


# Pursed Lips Detector

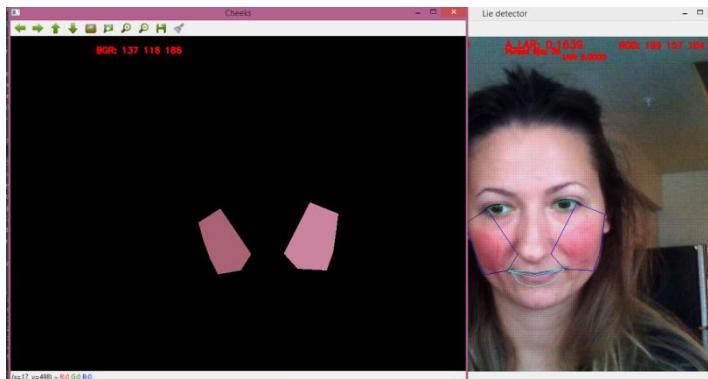
- LAR vrednost: *lips aspect ratio*
- Prag:  $0.8 * \text{LAR}$



# Blushing detector



Regioni obrazu



- Prvobitna, naivna implementacija: prilikom crvenila dolazi do rasta samo crvenog kanala RGB
- Međutim, u slučaju okretanja lica ka svetlosti, vrednosti sva tri kanala se naglo uvećavaju
- Stoga je u obzir uzet prag koji mogu da pređu vrednosti crvenog, plavog i zelenog kanala kao i ukupna promena vrednosti sva tri kanala

# Podsistem zadužen za detekciju istinitosti iskaza

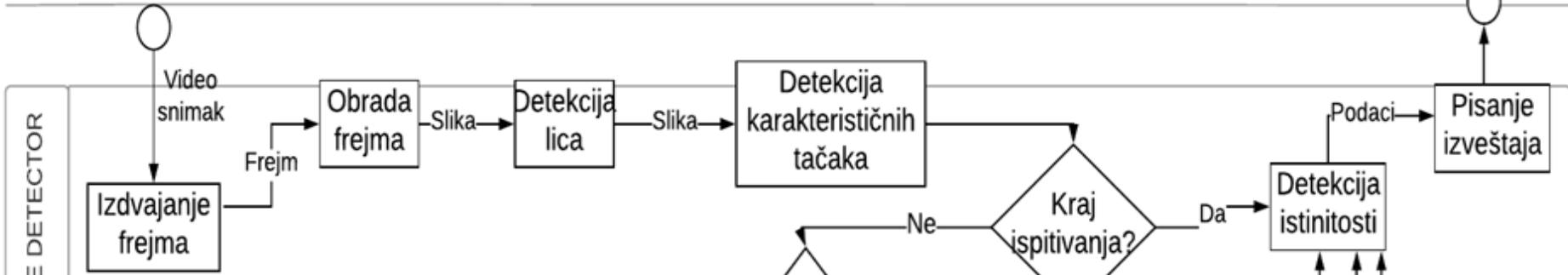
---

- k-NN algoritam
  - $k=12$
- Neuronska mreža
  - 2 skrivena sloja po 60 neurona
  - *ReLU* aktivaciona funkcija u skrivenim slojevima
  - *Sigmoid* aktivaciona funkcija u izlaznom sloju, prag 0.5

# Dijagram toka podataka

Sara | August 25, 2018

LIE DETECTOR



BLINK DETECTOR

PURSED LIPS DETECTOR

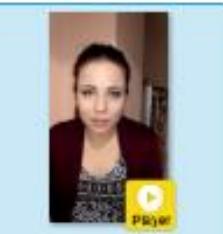
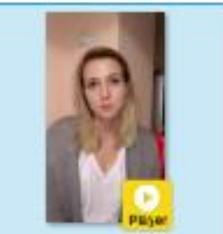
BLUSHING DETECTOR

# Skup podataka

- 8 snimaka na kojima osobe govore istinu\*
- 8 snimaka na kojima osobe govore neistinu\*
- 8 snimaka na kojima se osobe ponašaju po karakterističnim šablonima laganja

\*Ispitanici nisu bili upućeni u rad sistema prilikom snimanja istinitih i lažnih iskaza

- Video sadržaj je sniman u istim uslovima: pod istim osvetljenjem, iz istog ugla (ispitanici naspram kamere)
- Ručno izdvojeni potrebni podaci: broj treptaja po sekundi, koliko puta je osoba porumenela, koliko puta je osoba stisnula usne kao i da li je iskaz istinit ili ne
- Dodata je kolona prosečnog broja treptaja u sekundi ponaosob – neki ljudi znatno manje učestalo trepću od drugih, a to može da utiče na rezultat



# Verifikacija

---

- Empirijska verifikacija podsistema zaduženog za izdvajanje karakterističnog ponašanja
  - Problemi: neadekvatno osvetljenje
- Verifikacija podsistema zaduženog za detekciju istinitosti iskaza:
  - Empirijski
  - Podjela na trening-test skup podataka (csv datoteka, ekstrahovana obeležja šablonu laganja)

# Rezultati

---

- Nedeterminističko ponašanje
  - U zavisnosti od pokretanja (slučajne podele na trening-test skup podataka) različite postignute tačnosti
- k-NN tačnost: 33%, 66%, 100%
- Neuronska mreža tačnost: uglavnom iznad 75%
- Razlog: mali skup podataka!

# Automatsko generisanje skupa podataka za upoređivanje sličnosti poznatih ličnosti na slici sa slikama drugih ljudi

- Noemi Szabados
- Materijali prezentovani ovde preuzeti su iz diplomskog rada autora

# Motivacija

---

- Tačnost rada sistema, koji se oslanjaju na modele mašinskog učenja, u velikoj meri zavise od kvaliteta skupa podataka koji se koristi za obučavanje modela
  - Idealno, skup podataka bi trebao biti što raznovrsniji kako bi modeli obučeni nad njim imali veću moć generalizacije
  - Često se u praksi srećemo sa problemom da su dostupni podaci za treniranje modela oskudni
- Projekat: prepoznavanje sličnosti lica osobe sa licem poznate osobe
  - <https://github.com/ftn-ai-lab/sc-2017-siit/issues/17>
  - Nije postojao dovoljno dobar skup podataka

# Rešavani problem

---

- Cilj: **automatsko** generisanje skupa podataka za ovaj problem
- Slike treba da se automatski dobave i razvrstaju u različite podskupove po određenim uslovima
  - Osoba sa osmehom
  - Osoba gleda u različitim smerovima (pogled na gore, na dole , u levo, u desno i pogled pravo)
  - Osoba sa bradom
  - Osoba sa šiškama

# Metodologija

Lista imena

Pronađena na forumu

Skidanje slika sa interneta

*Google Images* (prvih 200 slika za svako ime)

Upoređivanje lica

Uklanjanje nepotrebnih slika

Filtriranjem su uklonjene:

- Nevalidne datoteke
- Slike koje ne sadrže ljudsko lice
- Slike sa više lica
- Slike na kojima lica nisu slična, odnosno, ne predstavljaju lice traženog čoveka

Generisanje skupa podataka

Prepoznavanje  
lica

Uklanjanje  
slika sa  
više osoba

Prepoznavanje  
zatvorenih  
očiju

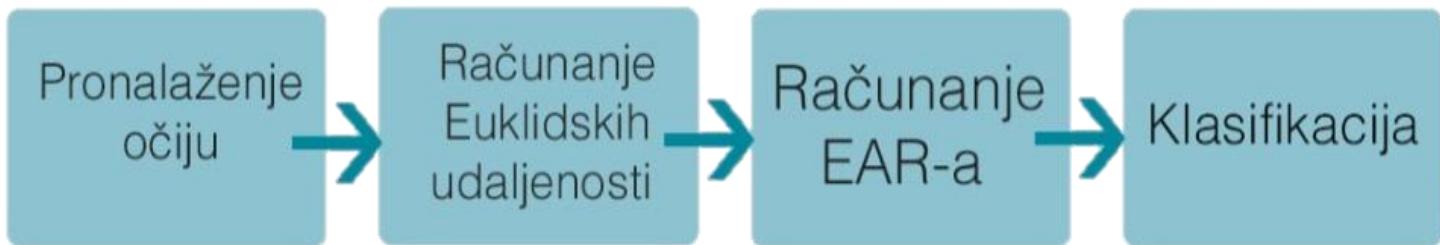
Prepoznavanje  
osmeha

Prepoznavanje  
smera  
lica

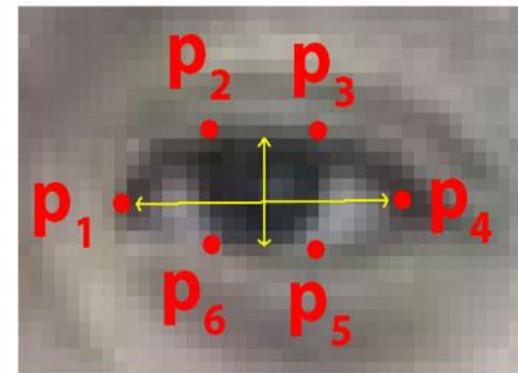
Prepoznavanje  
brade

Prepoznavanje  
šiški

# Prepoznavanje zatvorenih očiju



$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$



Soukupová, Tereza, and Jan Čech. "Real-time eye blink detection using facial landmarks." *21st Computer Vision Winter Workshop*. 2016

# Prepoznavanje osmeha

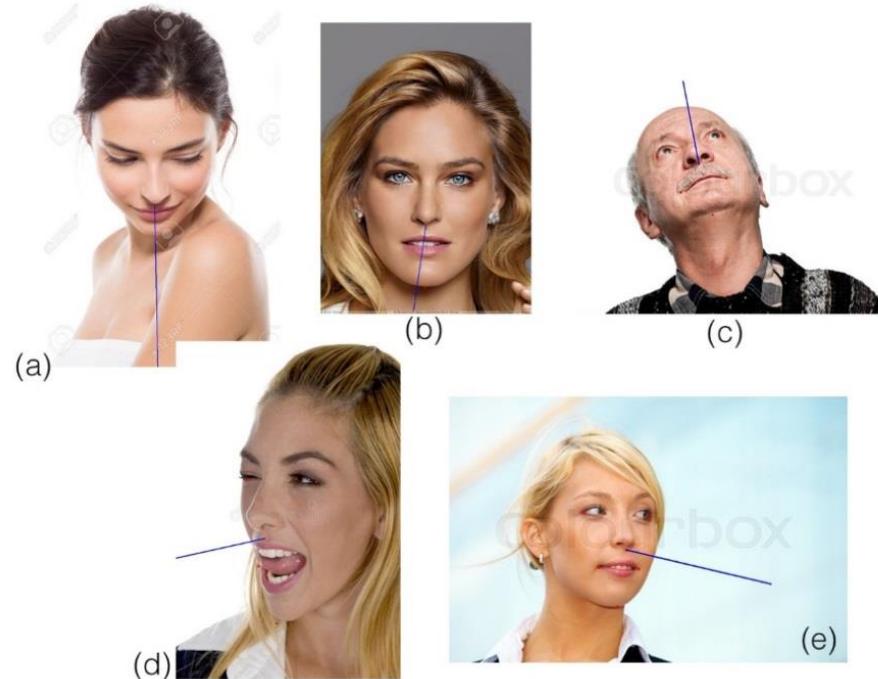
---

- *Real time Smile recognition* detektor

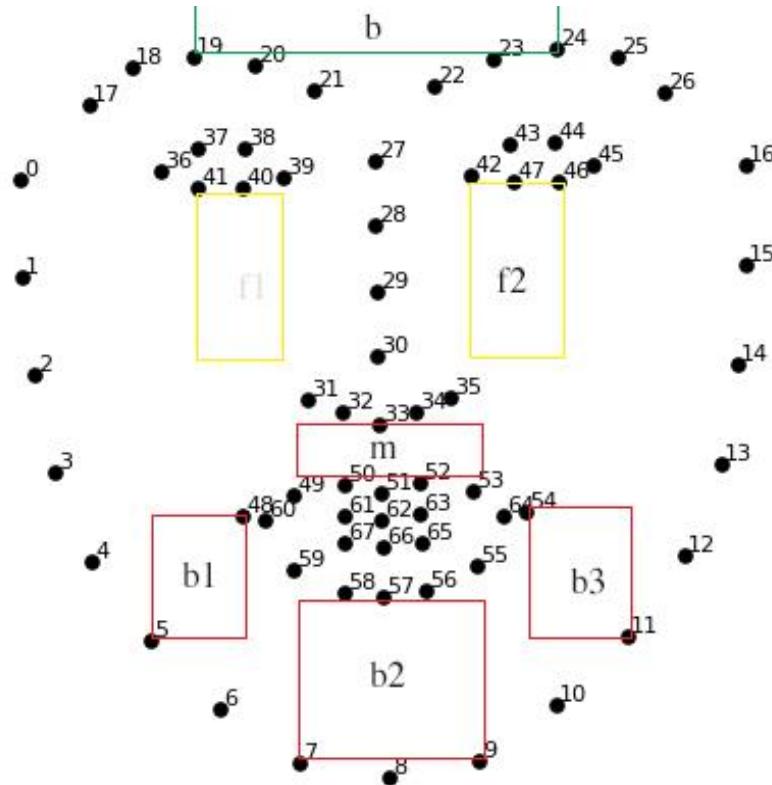
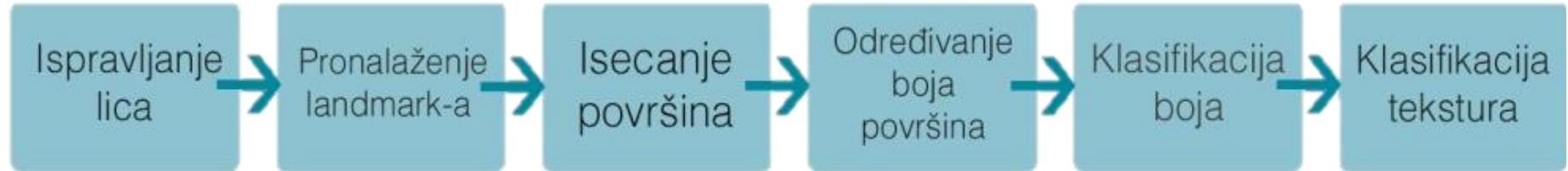
<https://github.com/aquibjaved/Real-time-Smile-Recognition>

# Prepoznavanje smera glave

- 2D koordinate lica – implementacija u *dlib*
- Određivanje smera: *projectPoints* u *OpenCV*



# Prepoznavanje brade i šiški



# Rezultati

---

- Filtriranje
  - Pre: 176 imena, 7,4 GB memorije na disku, ukupno 35543 slika
  - Nakon: 121 imena, 1,83 GB memorije, ukupno 10656 slika
- Generisanje skupa podataka
  - 1411 slika
- Mana – ista slika, drugo isecanje



# Generisanje slike za prevaru obučenih neuronskih mreža

- Tomislav Dobrički i Ognjen Vlajić
- <https://github.com/ftn-ai-lab/sc-2017-siit/issues/16>
- Materijali prezentovani ovde preuzeti su iz diplomskog rada Tomislava Dobrički

# Opis problema

---

- Algoritam prevari neuronsku mrežu tako što generiše sliku koju će mreža klasifikovati u unapred zadatu klasu
- Nije nužno da će generisana slika izgledati kao klasa u koju treba da se klasifikuje
- Poželjno je da algoritam bude lako primenljiv na veliki broj neuronskih mreža, različih arhitektura, bez preteranih izmena
- Algoritmu ne treba da budu poznate klase kojima neuronska mreža barata, već generiše sliku „slepo“

# Motivacija

---

- Generisane slike se mogu iskoristiti za augmentaciju skupa za treniranje neuronske mreže (Rubbish class). Time je moguće dostići bolju generealizaciju modela
- Bolje razumevanje procesa zaključivanja neuronske mreže na osnovu generisanih slika



Algoritmom generisana slika koja bi trebala da predstavlja cifru 8

# Direktan pristup

- Pristup baziran na izračunavanju optimalne ulazne vrednosti neuronske mreže za neku klasu

- Formula za aktivaciju neurona:

$$a_4 = x_1 w_{41} + x_2 w_{42} + x_3 w_{43}$$

- Aktivacija neurona izlaznog sloja:

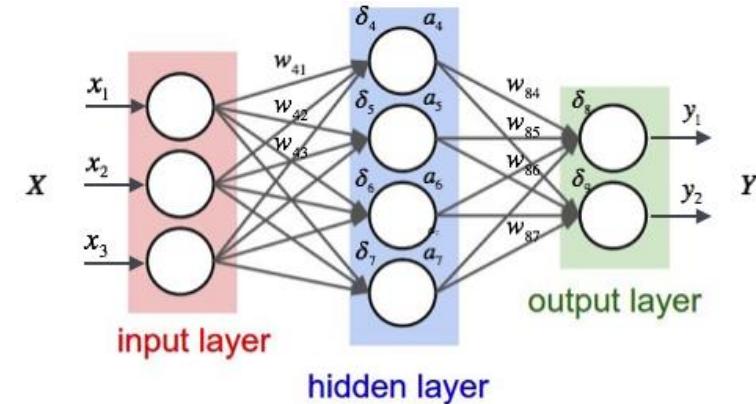
$$y_1 = a_4 w_{84} + a_5 w_{85} + a_6 w_{86} + a_7 w_{87}$$

- Ako uvrstimo vrednosti svakog "a":

$$y_1 = K_1 x_1 + K_2 x_2 + K_3 x_3 + K_4 x_4$$

- Gde je svako  $K$  zbir proizvoda težina:

$$K_1 = w_{41} w_{84} + w_{51} w_{85} + w_{61} w_{86} + w_{71} w_{87}$$



- Na osnovu vrednosti  $K$  možemo izračunati maksimalnu vrednost od  $y_1$
- Dovoljno je da znamo da li je  $K_n$  pozitivno ili negativno. Ako je pozitivno onda za  $x_n$  treba da uzmemo što veću vrednost (255), a u suprotnom što manju (0).

# Direktan pristup

---

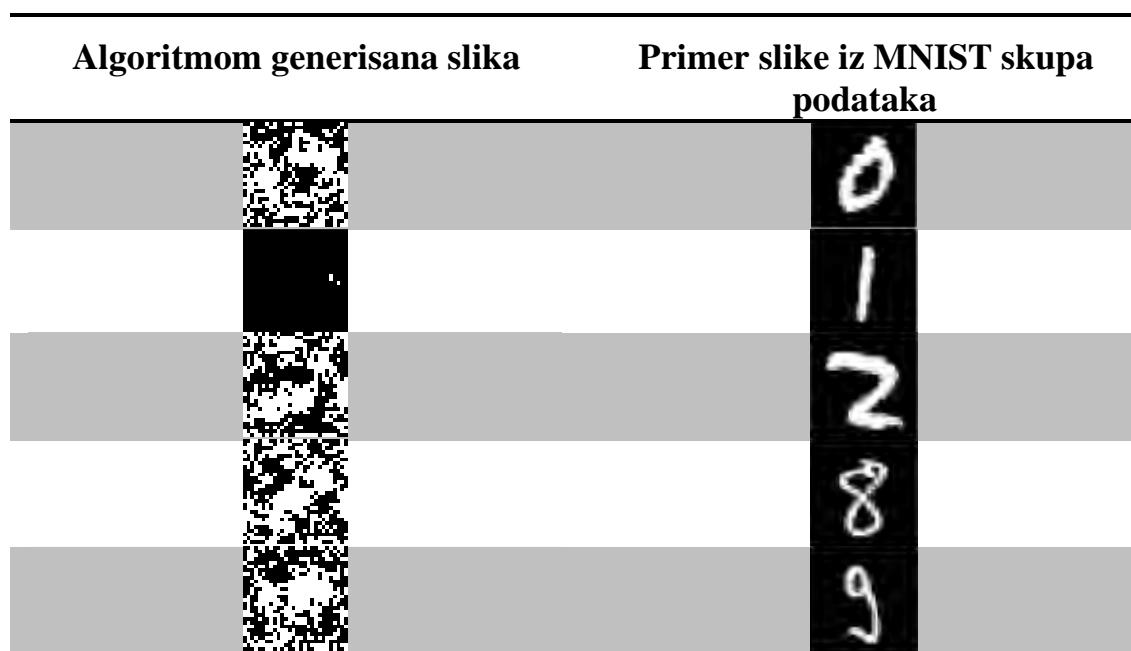
- Izvedene formule zanemariju neke česte karakteristike neuronskih mreža. Zbog toga ovaj pristup ima sledeća ograničenja:
  - Primenljiv je samo na potpuno povezane mreže;
  - Potrebno je voditi računa o svim mogućim aktivacionim funkcijama;
  - Mora se poznavati unutrašnja struktura mreže, zajedno sa vrednostima svih težina;
  - Množenje malih brojeva je računarski zahtevna operacija, a težine su obično mali brojevi.

# Pristup baziran na pretrazi

- Pretražuje se skup svih mogućih slika za date dimenzije
  - $256^n$  kombinacija (gde je  $n$  broj piksela na slici)
  - Ako iskoristimo formule iz direktnog pristupa i za vrednosti piksela razmatramo samo 255 i 0, možemo smanjiti skup na  $2^n$
- Prilikom pretrage se za funkciju heuristike koristi sama neuronska mreža za koju generišemo sliku
  - Vrednost jednog piksela slike se postavlja na jednu od vrednosti (0 ili 255) i slika se prosleđuje mreži
  - Ako je povratna vrednost veća od predhodne tada vrednost piksela ostaje, u suprotnom se vraća svoju inicijalnu vrednost
  - Ovaj proces se ponavlja za svaki piksel na slici.
  - Kod konvolutivnih neuronskih mreža se više puta prolazi kroz svaki piksel slike, sve dok se ne dobije dovoljno velika vrednost heuristike

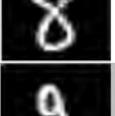
# Verifikacija – NM za klasifikaciju cifara

- Potpuno povezana neuronska mreža trenirana na MNIST skupu podataka
  - Prilikom evaulacije dostignuta je tačnost od 98.15%
  - Za svaku cifru algoritam uspešno pronalazi sliku koju mreža klasificiše sa 100 % pouzdanosti



# Verifikacija – CNN za klasifikaciju cifara

- CNN takođe trenirana na MNIST skupu podataka
  - Prilikom evaulacije dostignuta je tačnost od 98.64%
  - U proseku je potrebno 10 prolaza kroz svaki piksel slike da se dostigne tačnost od 99%

Generisana slika	Primer cifre iz MNIST skupa	Generisana slika	Primer cifre iz MNIST skupa
			
			
			
			
			

# Verifikacija – NM za detekciju lica

- Neuronska mreža klasificuje slike u klase „lice“ i „nije lice“
  - Za obučavanje su korišćeni skupovi podataka *Labeled faces in the Wild* i CIFAR-10
  - Prilikom evaulacije je zabeležena tačnost od 95%
  - Algoritam uspešno izgeneriše sliku koju mreža sa 100% puzdanosti klasificuje kao lice



Primeri slika iz *Labeled faces in the Wild* skupa podataka



Slika dobijena kao rezultat algoritma

# Prepoznavanje demencije sa MRI snimaka

- Miloš Pavlić i Ognjen Francuski
- <https://github.com/ftn-ai-lab/sc-2017-siit/issues/6>
- Materijali prezentovani ovde preuzeti su iz diplomskog radova autora

# Motivacija

---

- Alchajmerova bolest je progresivni nepovratni poremećaj mozga
  - Manifestuje se kao gubitak kognitivnih sposobnosti i pamćenja
  - u najgorim slučajevima dolazi i do nemogućnosti izvršavanja čak i najjednostavnijih zadataka
- Samo u 2010., 135 miliona pacijenata je dijagnostikovano
- Ekonomski uticaj u svetu je procenjen na \$600 biliona
- Procenjuje se da će do 2050. biti afektovano 1.5 biliona ljudi

# Motivacija

---

- Uzrok je redukcija sekrecije neurotransmitera koja dolazi sa starenjem
- Iako je bolest neizlečiva, njen progres se može značajno usporiti blagovremenim tretmanima – rana dijagnoza je od velike važnosti
- Bolest se dijagnostikuje pomoću kliničkih testova i MRI snimaka

# Motivacija

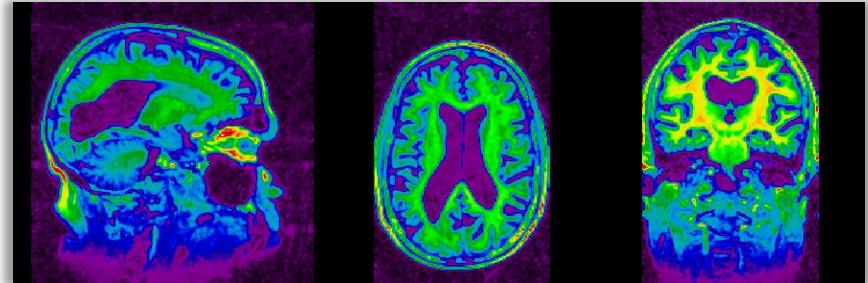
---

- Broj starijih ljudi u svetu se uvećava mnogo brže od broja eksperata koji mogu da dijagnostikuju bolest
- Cilj ovog istraživanja je olakšavanje i ubrzanje rad stručnog medicinskog osoblja pri dijagnostici

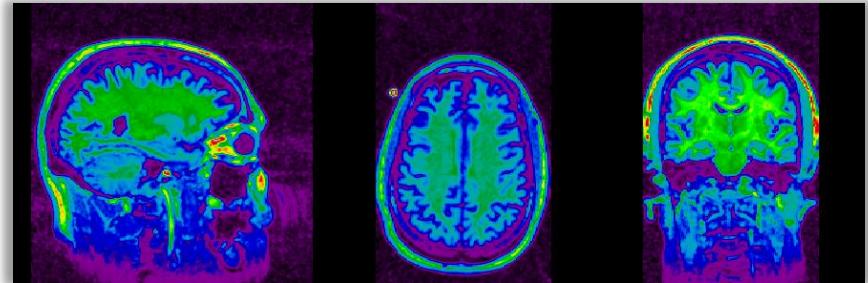
# Problem

- Prepoznavanje demencije na osnovu
  - MRI snimaka
  - Demografskih podataka
  - Rezultata kliničkih testova
- Korišćena obeležja se mogu prikupiti
  - Neinvazivnim metodama
  - Od stane osoblja koje nisu domenski eksperti (neurolozi ili psihijatri)

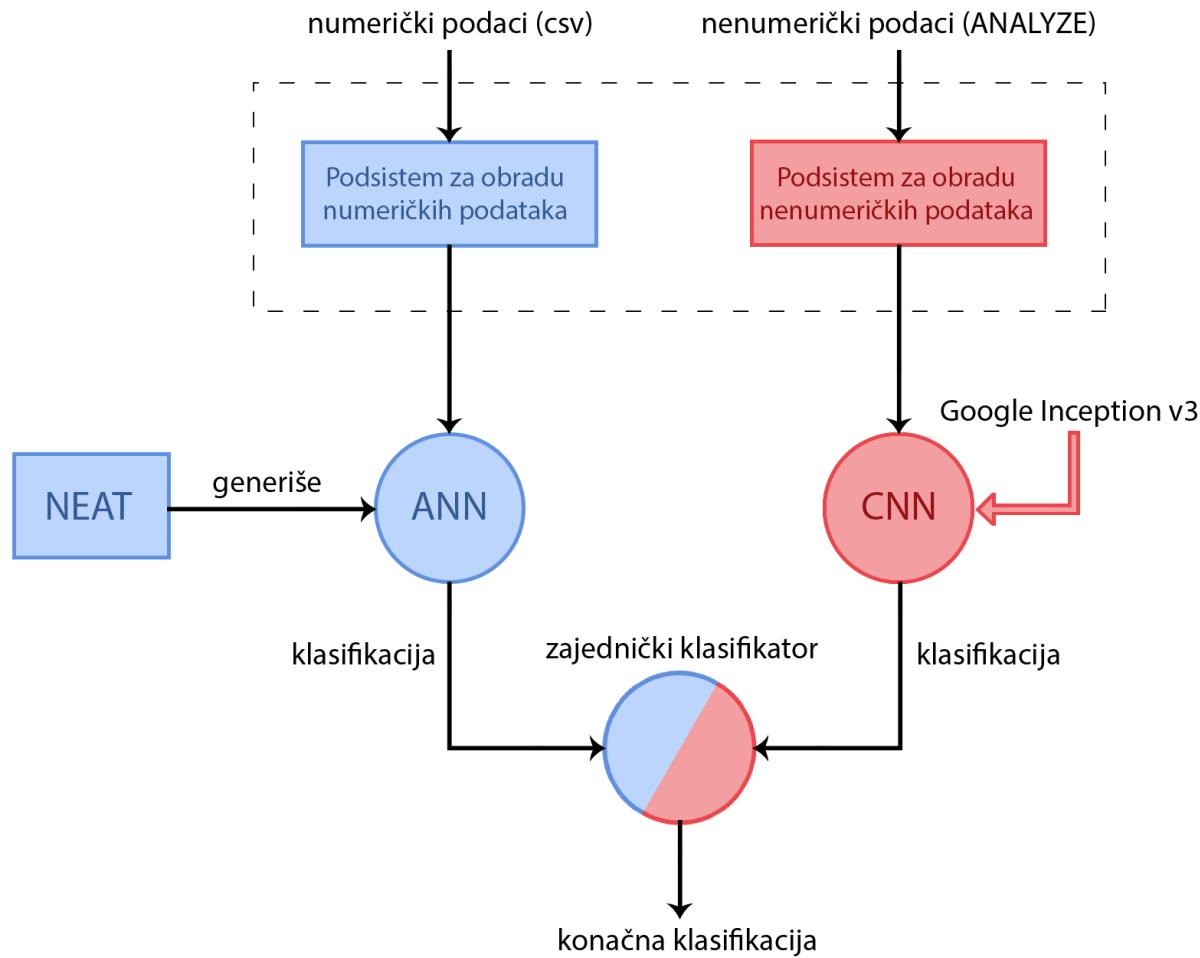
Sa demencijom



Bez demencije



# Metod



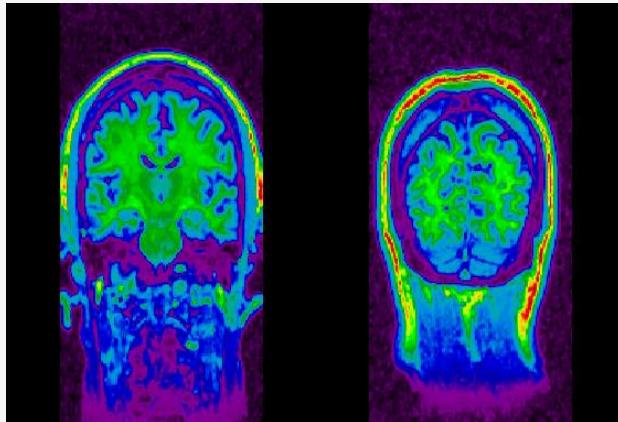
# Analiza rešenja

- ▶ Uzeti preseci snimaka

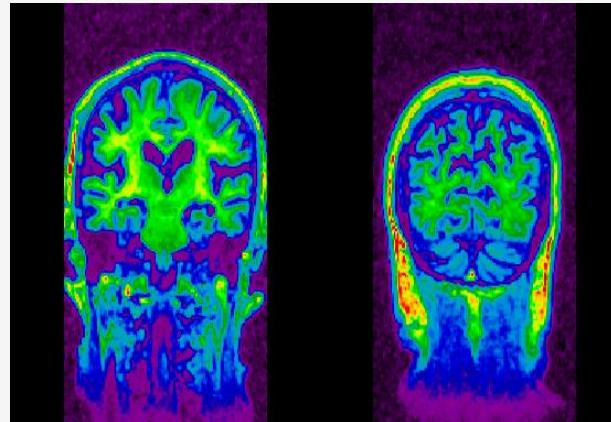
- ▶ Problemi:

1. Razlike između klasa su jasnije na središnjim presecima, manje jasne na kraјnim presecima
2. Preseci pored moždanog tkiva pokazuju i ostala, dovodi do učenja *feature-a* nebitnih za konkretni problem

Bez demencije



Sa demencijom



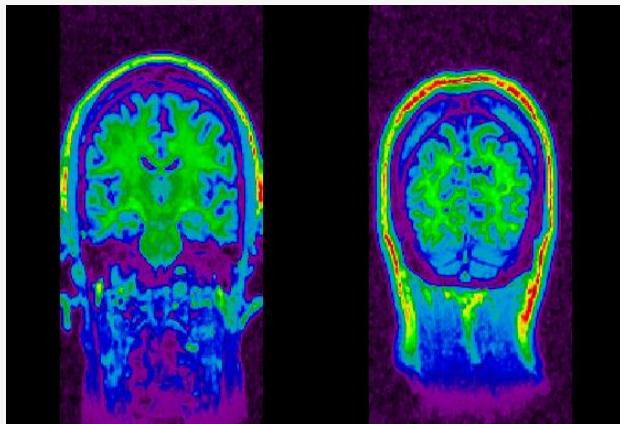
# Analiza rešenja

- ▶ Uzeti preseci snimaka

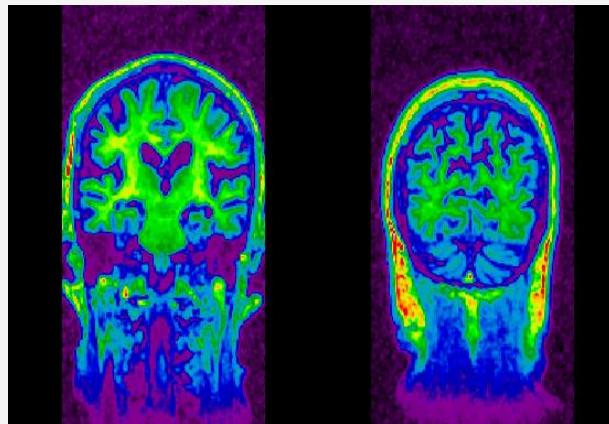
- ▶ Rešenja:

1. Izbaciti slike preseka na krajevima
2. Izbaciti sva tkiva koja ne predstavljaju sivu masu mozga sa MRI snimaka, pa zatim raditi dalje preprocesiranje i konverzije

Bez demencije



Sa demencijom



# Prepoznavanje broja i imena autobuske linije sa LED natpisa na osnovu slika autobra

- Olivera Hrnjaković i Vlada Đurđević
- <https://github.com/ftn-ai-lab/sc-2017-e2/issues/53>

# Postupak

## 1. Pronalaženje autobusa na slici

- a) *Grayscale*
- b) Histogram
- c) Klasifikacija slika na osnovu nivoa osvetljenosti
- d) Primena filtera za plavu boju na osnovu klase slike



# Postupak

## 2.

### Izdvajanje prednjeg dela autobusa

- a) Erozija i dilacija (otklanjanje šuma)
- b) Pronalaženje kontura (*Canny*)
- c) Izdvajanje prednjeg dela autobusa na osnovu osobina konture
- d) Isecanje prednjeg dela autobusa sa slike



# Postupak

## 3. Pronalaženje slova koja čine natpis

- a) Primena filtera za boju slova sa natpisa
- b) Pronalaženje kontura (*Canny*)
- c) Filtriranje: ostavljaju se one konture koje po dimenzijama odgovaraju obliku slova



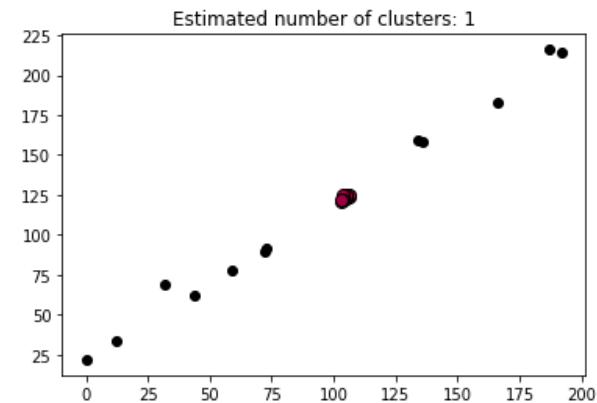
# Postupak

## 4. Izdvajanje natpisa autobusa

- Predstavljanje kontura u 2D prostoru na osnovu vrednosti gornje i donje granice konture
- Uklanjane viška kontura uz pomoć DBScan algoritma
- Isecanje samog natpisa slike



- Klasifikacija slika na osnovu osvetljenja
  - Otsu izračunavanje praga
  - Klasifikacija slika na osnovu broja najosvetljenijih piksela
  - Određivanje globalnog praga tako što se od praga dobijenog Otsu metodom oduzme (ili doda) vrednost u odnosu na klasu prema osvetljenju



Original Noisy Image



Otsu's Thresholding (88.0)



# Postupak

---

## 5. Prepoznavanje slova neuronskom mrežom

- a) Pronalaženje kontura na natpisu i izdvajanje slova na osnovu visine i širine
- b) Slanje slova na veštačku neuronsku mrežu
- c) Predikcija slova i određivanje autobuske linije na osnovu najmanje Levenštajnove duljine u odnosu na unapred određene vrednosti mogućih autobuskih linija

# Potencijalni problemi

---

- Osvetljenje
  - Delimično rešen, kategorisanjem slika po osvetljenju
- Mali skup podataka
- Šum iste boje kao autobus ili natpis
  - Uvesti dodatne filtere koji se oslanjaju i na boje i konture
  - CNN segmentacija

# Chat robot

- Dejan Kuzmanović
- <https://github.com/kuzmanovicd/chat-robot>

# Problem

---

- Potrebno je kreirati *chat* robota koji treba da odgovara na pitanja korisnika
  - Obučiti robota da automatski odgovara na jednostavnija pitanja u formi Q&A
  - Jedna od poželjnih osobina je oponašanje odgovora neke specifične ličnosti
- Motivacija
  - *Chat* robot se može iskoristiti kao *chat* agent za razne stvari (sales support, IT support, Q&A websites) a da korisnik i ne primeti da razgovara sa robotom

# Skup podataka

---

- Može se koristiti bilo koji tekstualni skup podataka koji sledi format:
  - Svaka neparna linija je input (Question)
  - Svaka parna linija je output (Answer)
- U projektu je iskorišćen *facebook chat history* sa privatnog profila
  - Ideja je da robot oponaša jednog *facebook* korisnika

# Obrada podataka

---

- Parsiranje html fajla sa chat istorijom  
<https://github.com/ownaginatious/fbchat-archive-parser>
- Konvertovanje svih slova u lower case
- Uklanjanje akcentovanih slova č,ć,š,ž,đ
- Izbacivanje specijalnih karaktera
- Odvajanje interpunkcijskih znakova razmakom (da bi se stvorili odvojeni tokeni)
- Spajanje više uzastopnih poruka jednog korisnika u jedno pitanje, odnosno odgovor
- Svaka kombinacija "hahaha" je pretvorena u "haha"

# Stvaranje tokena za svaku reč

- Predefinisani tokeni
  - `_UNK` - za reči koje se ne pojavljuju često
  - `_EOS` - End of string za označavanje kraja rečenice
  - `_GO` - Indikator za početak generisanja odgovora
  - `_PAD` - Prazno mesto zbog fiksne veličine ulaza
- Primer tokenizovane rečenice gde je broj ulaza fiksiran na 8:  
`>>> Dodji na pivo.`  
`['dodji', 'na', 'pivo', '.', '_EOS', '_PAD', '_PAD', '_PAD']`
- Odgovor:  
`>>> Eto me`  
`['_GO', 'eto', 'me', '.', '_EOS', '_PAD', '_PAD', '_PAD']`
- Veličina rečnika: 1150 (postotak nepoznatih reči je ispod 2%)
  - Svaka reč dobija jedinstven indeks korišćen za obučavanje NN
  - NN uči vezu između različitih indeksa (reči)

# Metodologija

---

- Rekurentna neuronska mreža (RNN)
  - Empirijski dokazana kao najbolje rešenje za modelovanje prirodnog jezika
- Mreža na kojoj se obučavao chat robot se sastoji iz dve rekurentne mreže:
  - Encoder – obrađuje ulaz
  - Decoder – generiše izlaz na osnovu izlaza iz encoder-a
  - Ovaj model je poznat kao *Sequence-to-Sequence* model

# Rezultati (odabrani zanimljivi odgovori)



# Zaključci

---

- Ograničenja
  - Kratko obučavanje
  - Mali (i nepravilan) skup podataka
- Mreža je uspela da nauči korelaciju između reči i da odgovara u sličnom ili istom kontekstu u kom je i pitanje
- Chat robot ne vodi evidenciju o prethodnim porukama i ne može da vodi konverzaciju – nastavak ovog projekta

# Drift AI Q-Learning

- Ljubiša Moćić
- <https://github.com/ftn-ai-lab/sc-2017-e2/issues/69>

# Problem

- Kreiranje autonomnog agenta koji je sposoban da savlada veštinu *drift-a*
- *Okruženje: Need for Speed Carbon*, staza Kime Temple.
- **Važni faktori:**
  - brzina ulaska u krivinu
  - vreme aktiviranja ručne kočnice
  - kontrolisanje vozila pri klizanju
  - izbegavanje udarca u zid
- **Nagrada:**
  - Na osnovu vremena klizanja, broja krivina u kojima se zaredom održi drift i brzine, se dodeljuju bodovi
  - Bodovi predstavljaju nagradu na osnovu koje algoritam shvata u kom stanju je bitno primeniti određene akcije



# Metodologija

---

- Q-Learning. U obzir se uzimaju:
  - Stanja u kojima se vozilo nalazi
  - Trenuci u kojima će algoritam reagovati radi izvršavanja akcija koje će dovesti do veće nagrade (tj. boljeg stanja)
  - Nagrada su bodovi
- Detekcija nagrade: *Google-ovo Tesseract* okruženje. Koraci detekcija bodova:
  - pravljenje *screenshot-a* kompletног ekrana
  - osnovna obrada frejma
  - detekcija regiona od interesa i isecanje ogovarajućeg dela frejma
  - validacija detekcije pomoću filtera

# Detekcija plućnih bolesti na rendgenskim snimcima upotrebom konvolutivnih neuronskih mreža

- Igor Majić
- Prikazani rezultati su preuzeti iz diplomskog rada autora
- <https://github.com/ftn-ai-lab/sc-2018-siit/issues/3>

# Motivacija

---

- Rendgenska dijagnostika grudnog koša je široko dostupna, jeftina i brza procedura koja može ukazati na razna oboljenja pluća
- Globalni nedostatak radiologa primorava lekare drugih specijalnosti da tumače rendgenske snimke
- Dobro obučen model bi mogao delimično zameniti eksperte u toj oblasti
  - Ogroman broj rendgenskih snimaka zajedno sa radiološkim izveštajima se čuva u arhivama velikih bolница
  - Ove arhivirane snimke je moguće pretvoriti u korisne skupove podataka

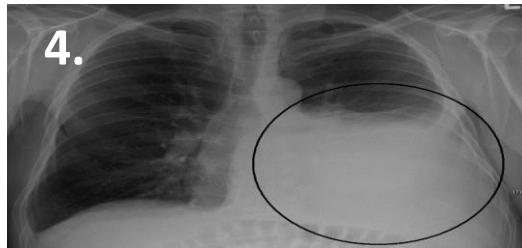
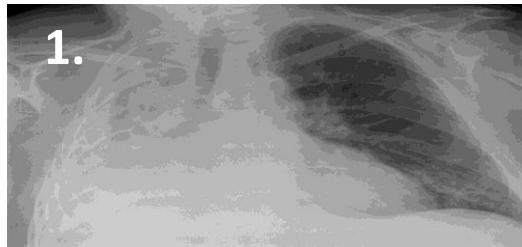
# Specifikacija problema

---

- Implementacija sistema za detekciju plućnih bolesti na osnovu rendgenskih snimaka grudnog koša
- Problem detekcije je *multi-label*
  - na jednom snimku se može pojaviti više bolesti

# Klase plućnih bolesti

- Obučeni model klasificuje četiri klase bolesti
  - (1) Atelectasis (2) Pneumothorax (3) Emphysema (4) Effusion



- Normalan nalaz je peta klasa

- (5) No finding



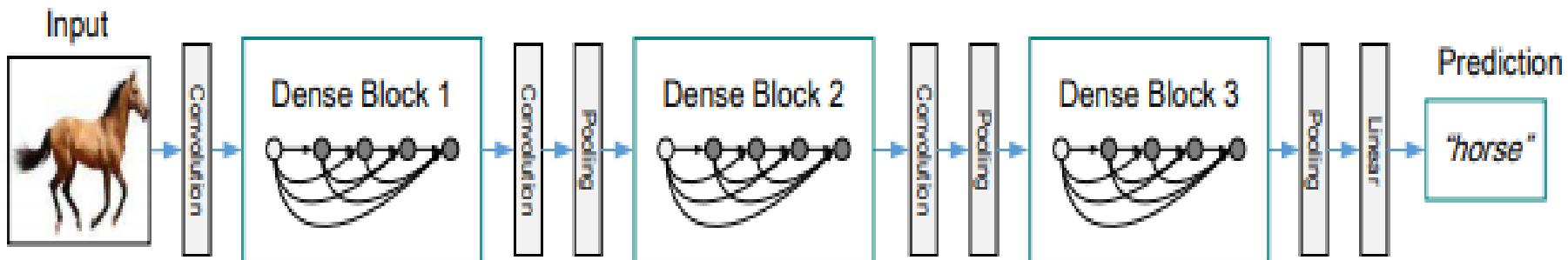
# Metodologija

---

- Sistem je sastavljen iz tri podsistema (modula):
  1. Modul u kome se vrši učitavanje, preprocesiranje i augmentacija
    - Skaliranje, normalizacija, vertikalno i horizontalno rotiranje
  2. Modul koji se bavi kreiranjem i podešavanjem modela
  3. Modul za obučavanje i testiranje

# Metodologija – kreiranje modela

- Model je konvolutivna neuronska mreža bazirana na DenseNet121 arhitekturi
- Pretrenirana na ImageNet skupu podataka, a potom dodatno obučavana
- Klasifikacioni sloj (softmax) od 1000 neurona zamenjen sigmoid slojem sa četiri izlaza (broj klasa bolesti)



# Skup podataka

---

- Javno dostupan skup podataka *ChestX-ray14*
- Ceo skup podataka ima (svih 14 bolesti) se sastoji
  - 112,120 snimaka / 30,805 jedinstvenih pacijenata
- Ovde je korišćen podskup od 4 bolesti i *No finding* klase
  - Ove klase su izdvojene korišćenjem ekspertskog znanja, zbog njihovih uzročno-posledičnih veza i dovoljnosti rendgenskog snimka za postavljanje dijagnoze
  - U pitanju je multi-label podskup
- Train/test podela je izvršena tako da se pacijent iz train skupa ne nađe u test skupu

# Rezultati

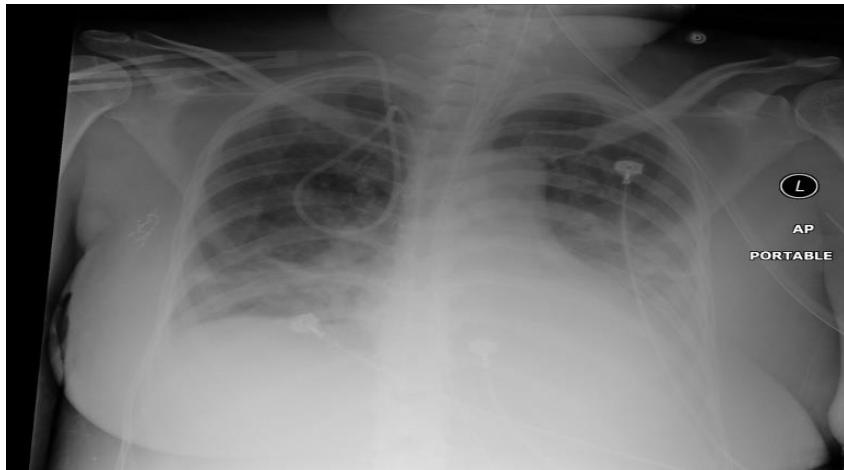
Klasa	F-mera
Atelectasis	0.356
Pneumothorax	0.395
Emphysema	0.452
Effusion	0.605
No Finding	0.700

- Posle *No findig*, model najbolje klasificuje efuziju, a najgore atelektazu
- Rezultati su blizu *state-of-the-art*

# Analiza grešaka modela



- Slika je označena kao emfizem i pneumotoraks
- Model nije uočio pneumotoraks, ali se u desnom plućnom krilu vidi dren kojim je evakuisan vazduh i time eliminisao pneumotoraks
- Ovo ilustruje nedostatak text mining algoritma pomoću kog je skup formiran



- Prikazan je snimak pleuralnog izliva (efuzija)
- Model je pored efuzije, detektovao i ateletazu
- Snimak je rađen u ležećem položaju
- Srčana senka je šira, što je najverovatnije protumačeno kao ateletaza (ateletaza je senka slične gustine)

# Analiza grešaka modela



- Pacijent je u ležećem položaju i rotiran
- Snimak je lošijeg kvaliteta
- Na snimku ipak nema bolesti, a model prepoznae pneumotoraks
- Najverovatnije donji deo desnog plućnog krila daje lažnu sliku parcijalnog pneumotoraksa

# Potencijalna proširenja

---

- Sistem bi se mogao poboljšati dodatnim obučavanjem i podešavanjem hiper-parametara
  - Inicijalni tempo učenja
  - Smanjivanje tempa učenja kad validaciona greška prestane da opada
- Inteligentnije mere augmentacije podataka, koje u obzir uzimaju regije karakteristične za bolesti
- Proširenje skupa uvođenjem više primera, kod manje frekventnih klasa

# Rekonstrukcija slika u uslovima ekstremno slabog osvetljenja

- Nikola Tomić
- Prikazani rezultati su preuzeti iz diplomskog rada autora
- <https://github.com/ftn-ai-lab/sc-2018-siit/issues/7>

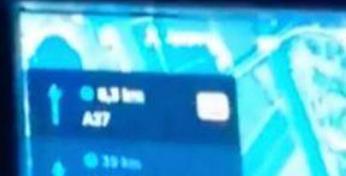
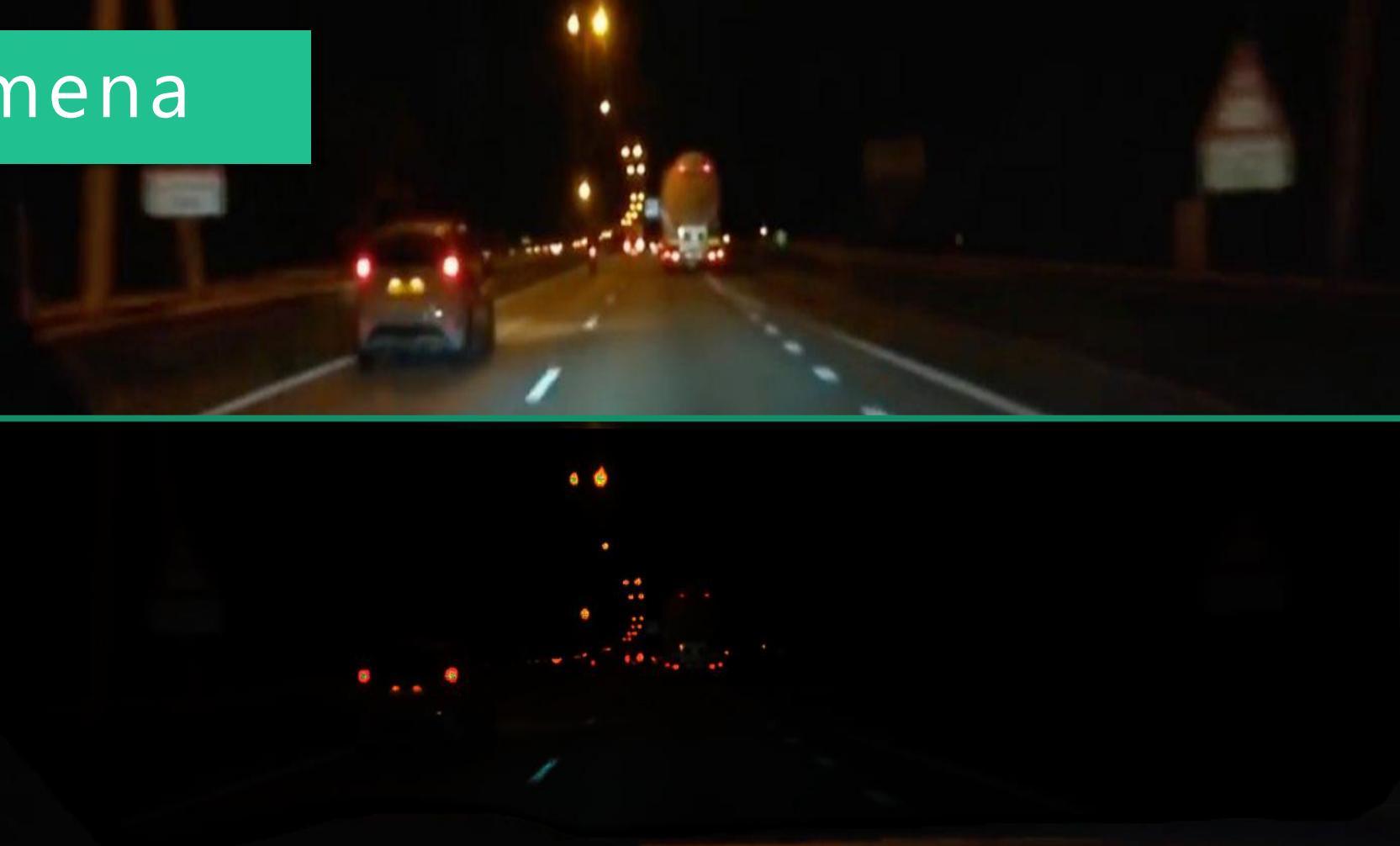
Motivacija

Phone X

Image shot on iPhone Xs. iPhone is a TM of Apple Inc.

Pixel 3a

# Primena



# Skup podataka

- ▶ 5049 slika kratke ekspozicije
- ▶ 424 referentnih slika duge ekspozicije
- ▶ **FUJITSU** 6000x4000
- ▶ **SONY** 4256x2848



10  
S

# Preprocesiranje



- ▶ Nekompresovan format
- ▶ Najveći broj informacija
- ▶ Oko 16384 nijanse po boji
- ▶ Kompresovan format
- ▶ Mali broj informacija
- ▶ 255 nijansi po boji



lsečak (ulaz)



okrenuta

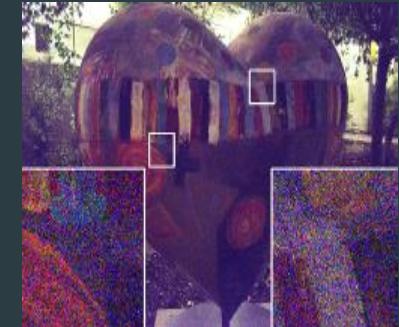


transponovana

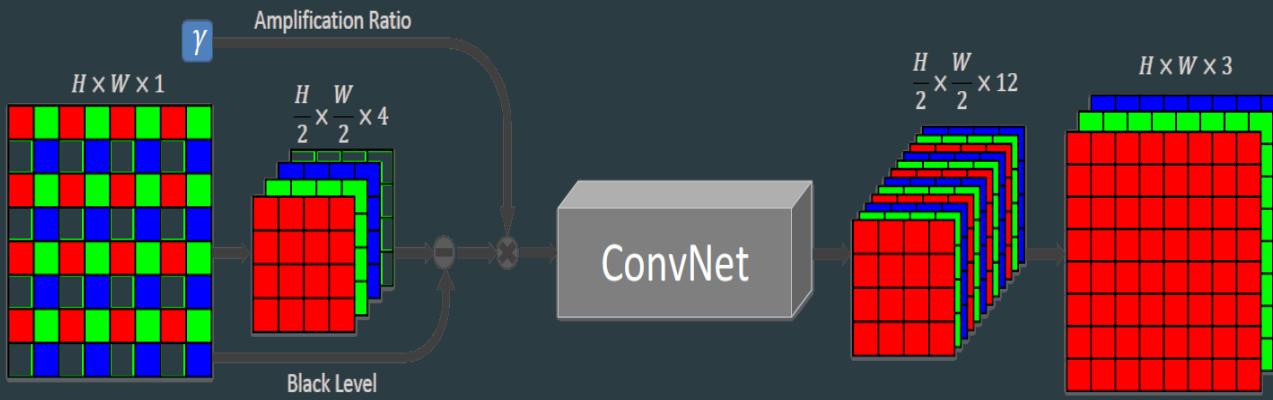
# Rešenje



Tradicionalni pristup



Burst denoising



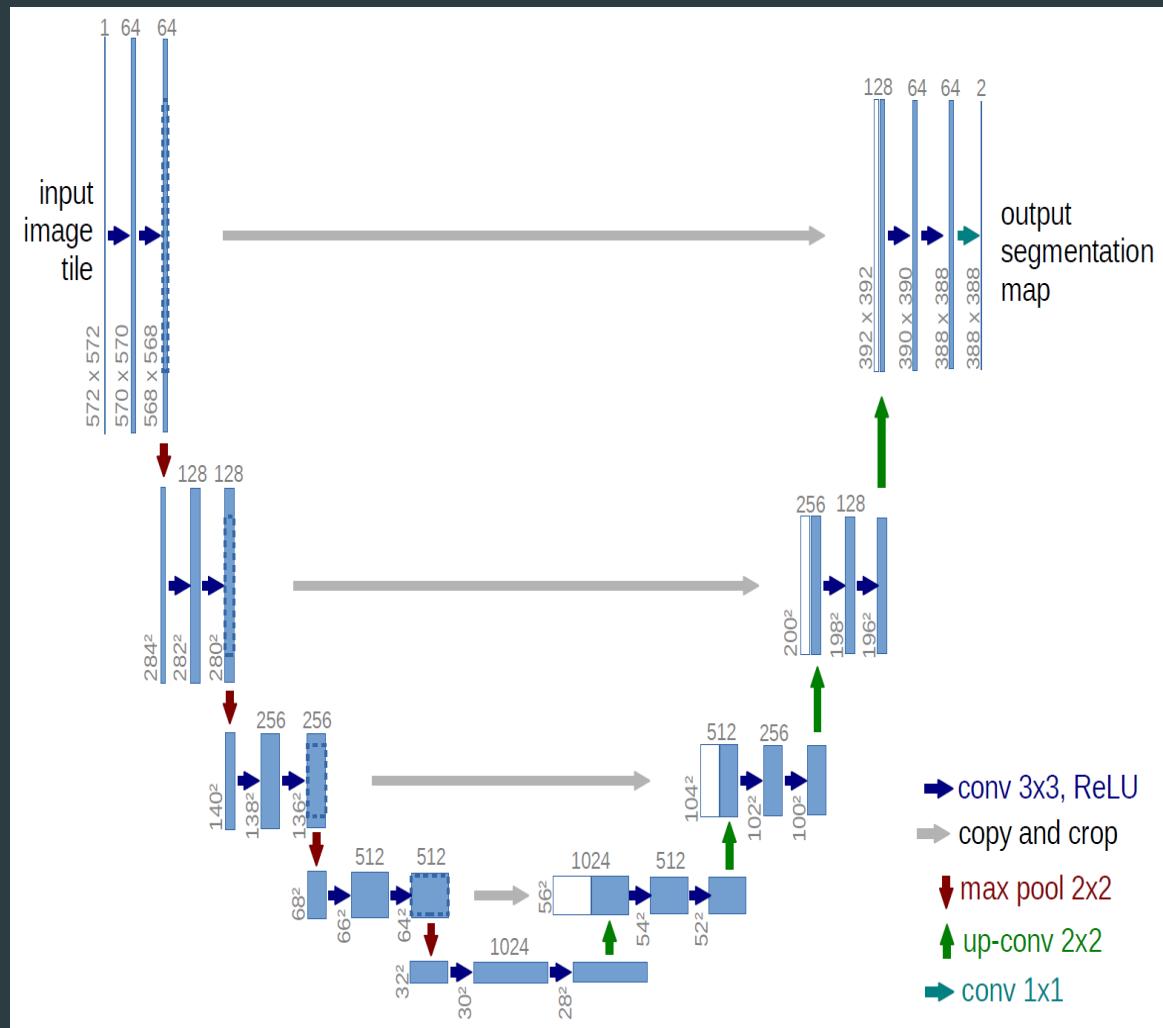
CNN pristup



CNN

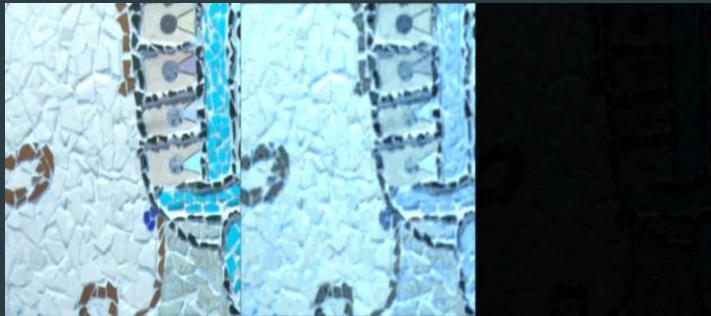
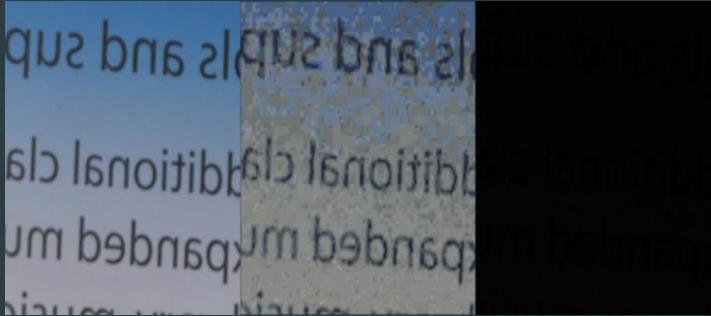
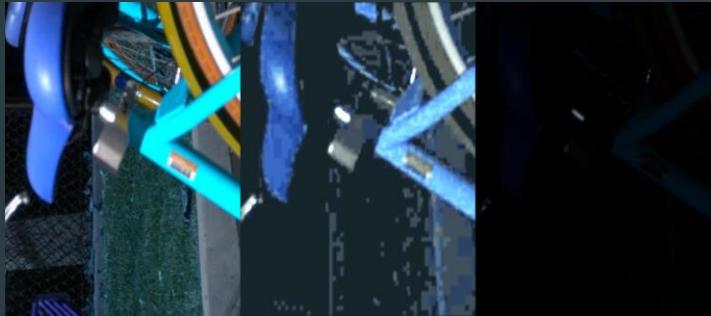
# U-Net

- ▶ Encoding - Decoding deo
- ▶ 10 konvolucionih slojeva (3x3)
- ▶ 5 max pooling slojeva (2x2)
- ▶ 4 dekonvolucionala sloja (2x2)

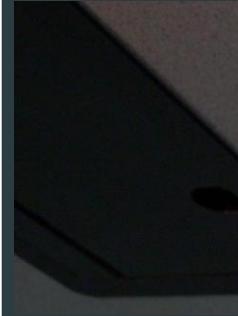


# Rezultati

Dobri primeri



Loši primeri



# DETEKCIJA ZAVISNIKA NA OSNOVU CRTA LICA

- Boris Bibić
- Prikazani rezultati su preuzeti iz diplomskog rada autora
- <https://github.com/ftn-ai-lab/sc-2018-e2/issues/31>

# Motivacija

- Povećan broj projekata koji koriste lice za identifikaciju
- Povećan broj zavisnika u svetu
- Propadanje lica kod višegodišnjih zavisnika
- Slabije performanse komercijalnih sistema za identifikaciju kod prepoznavanja zavisnika



# Metod – K-NN i SVM

## HOG verzija

- ulaz – slike lica
- skaliranje slike
- HOG deskriptor
- treniranje i evaluacija klasifikacionih algoritama
- izlaz – ispis rezultata

## Parametarska verzija

- ulaz – slike lica
- izvlačenje karakterističnih tačaka lica
- izračunavanje parametara
- treniranje i evaluacija klasifikacionih algoritama
- izlaz – ispis rezultata

# Metod – parametarska verzija

---

- Obeležja relevantna za dati problem
  - Simetrija lica
  - Zlatni presek
  - Naboranost kože
  - Crvenilo lica

# Skup podataka

- slike ručno prikupljene sa interneta
- 2 klase – „positive“ i „negative“
- 134 slike po klasi
- predprocesiranje – promena osvetljenosti, zasićenosti i nijansi boja



# Evaluacija

---

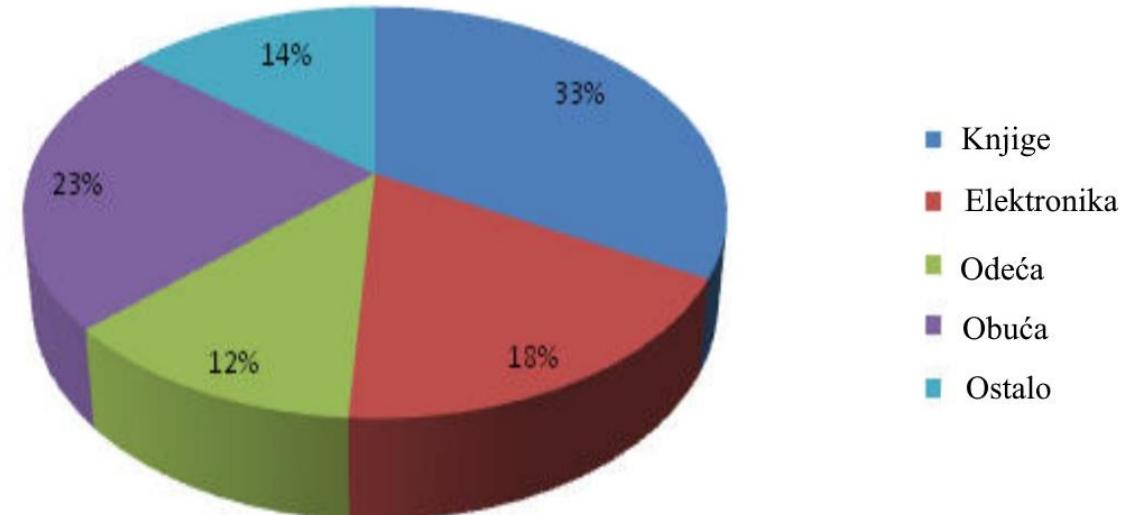
- Poređenje
  - tačnosti
  - brzine treniranja modela
  - brzine klasifikacije
  - analiziran je uticaj različitih obeležja
- Parametarsko učenje postiže bolje rezultate nego HOG deskriptor u zavisnosti od izbora parametara

# PREPORUČIVANJE ODEVNIH PREDMETA SA VIDEO ZAPISA

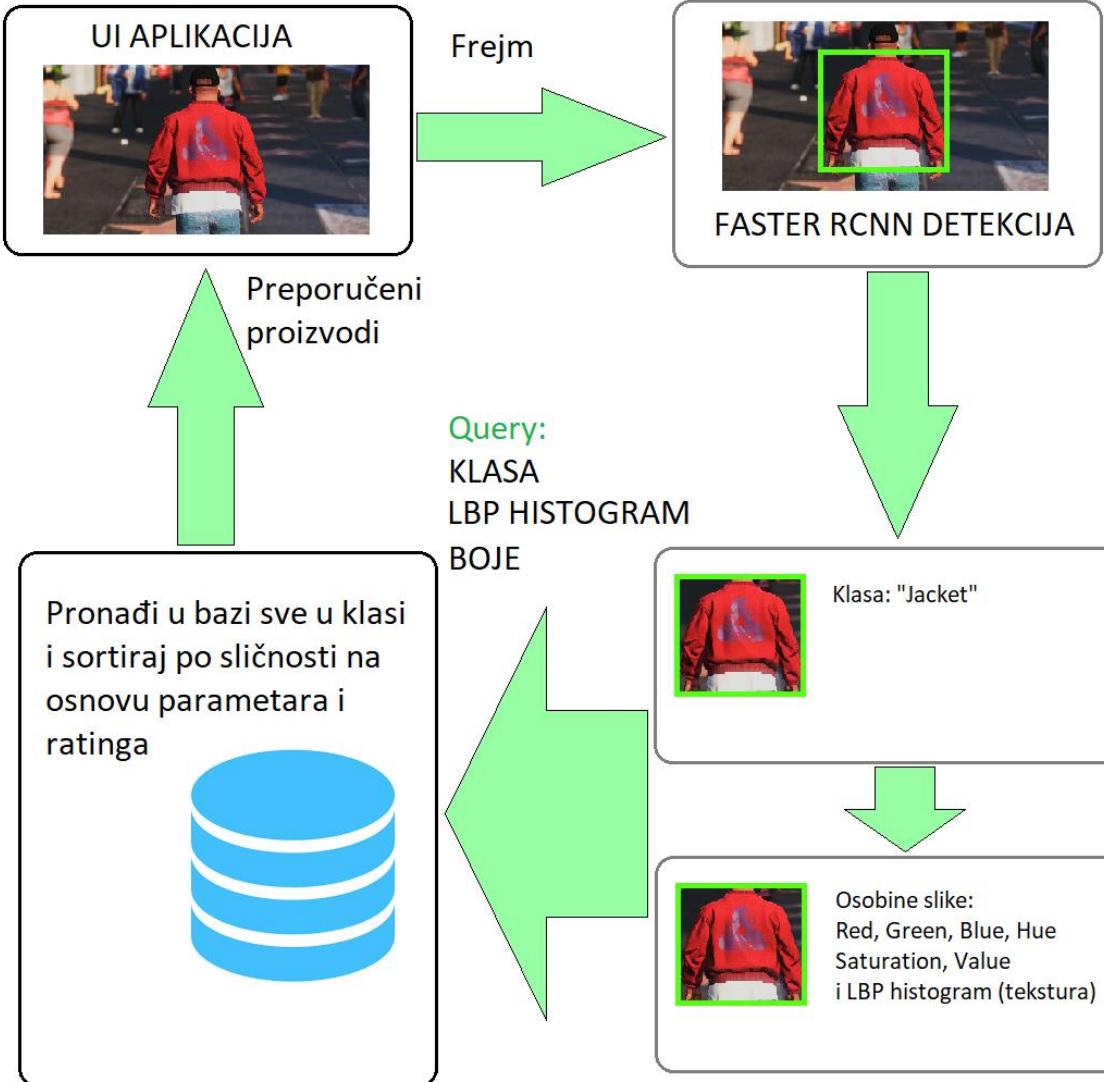
- Nebojša Basarić
- Prikazani rezultati su preuzeti iz master rada autora

# Motivacija

- *Online marketing* je u porastu
- Najzastupljenija je prodaja obuće i odeće
- Cilj: pobojšati interakciju kupaca i prodavaca u pogledu prodaje odeće
  - Kupac koji zapazi odeću koja mu se dopada na spotu ili filmu može jedim klikom dobiti preporuke vizuelno sličnih predmeta iz online prodavnica



# Metod



- Klasifikacija i detekcija: CNN (*Faster R-CNN inception coco V2*). Inicijalno obučenja na COCO skupu podataka, prilagođena primenom *transfer learning*

- *DeepFashion dataset*. Slike su označene sa
  - 50 opisnih kategorija
  - 1000 opisnih atributa

# Metod – modul za preporučivanje



Izvlačenje najveće konture



Red: 194  
Green: 13  
Blue: 4

Hue: 2  
Saturation: 230  
Value: 93

Texture: A blue histogram showing the distribution of pixel intensities across the t-shirt's surface.

Izvlačenje relevantnih obeležja

# Analiza ponašanja modela



Prepoznavanje šorca  
na slikama gde se ne  
vidi ceo donji deo tela

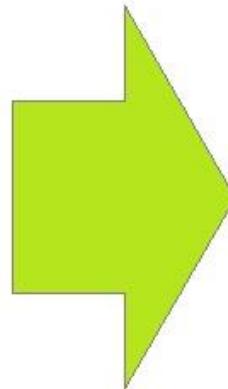


Suknja se često  
pogrešno prepoznaje  
kao haljina ili  
venčanica

- Najveća preciznost postignuta je kod farmerki
- Razlog za to je najverovatnije što su farmerke jedini odevni predmet koji predstavlja dugački donji deo odeće. Kada bismo uneli klasu pantalone pored farmerki, verovatno bismo značajno narušili preciznost

# Analiza ponašanja modela

- Najveći problem kod preporuke vizuelno sličnih predmeta je pronalaženje najveće konture na slici
- Problem se najčešće javlja kada je pozadina šarena ili je kontrast odeće i pozadine slab



# Klasifikacija mladeža sa dermoskopskih snimaka

- Svetislav Simić
- Prikazani rezultati su preuzeti iz diplomskog rada autora

# Motivacija

---

- Melanom je najopasniji tip tumora kože
- Rana detekcija je ključna
- Mobilni telefoni imaju veliki potencijal kao uređaji koji bi mogli da doprinesu rešavanju ovog problema

# Problem

- Binarna klasifikacija:



Benigni nevus



Melanom

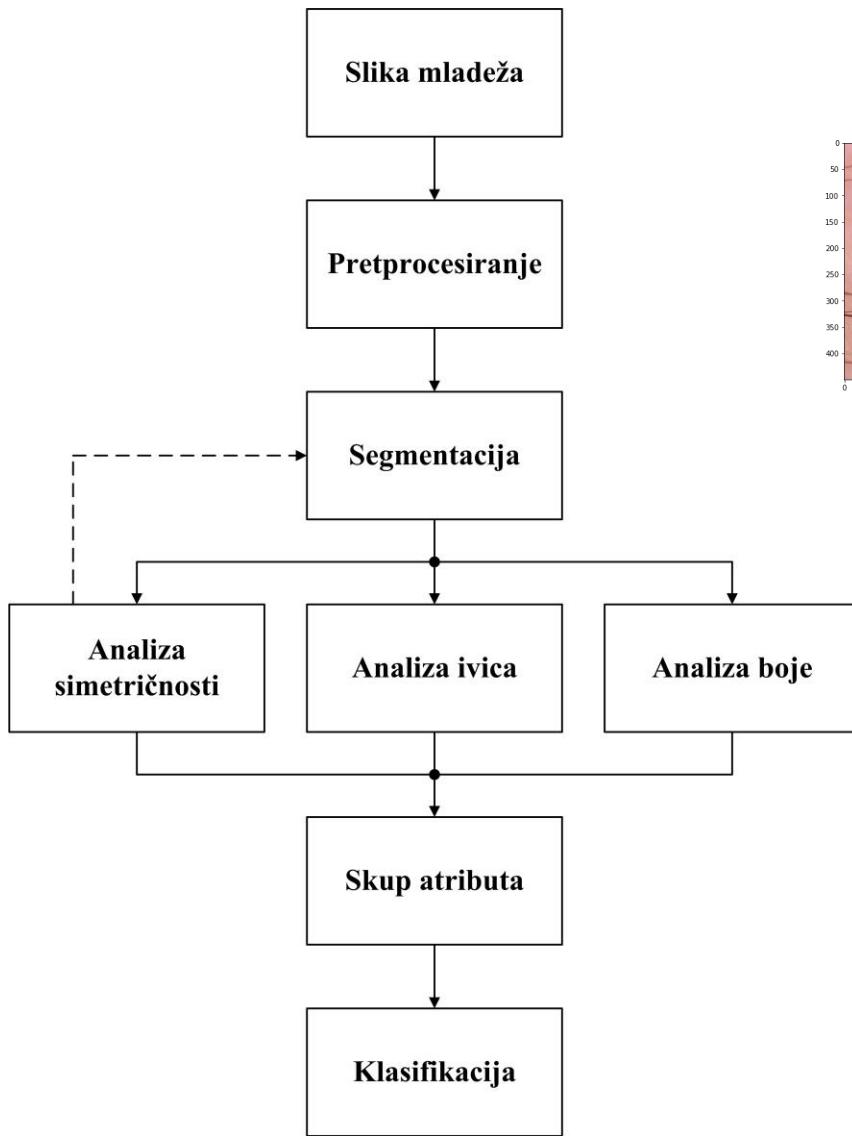
- Problem je ovako postavljen zbog opisanog značaja u ranoj detekciji melanoma
- Primjenjena su dva pristupa koji se zasnivaju na:
  - Tradicionalnim metodama mašinskog učenja
  - Dubokom učenju
- skup podataka *HAM10000* – slike kožnih promena

# ABCDE pravilo

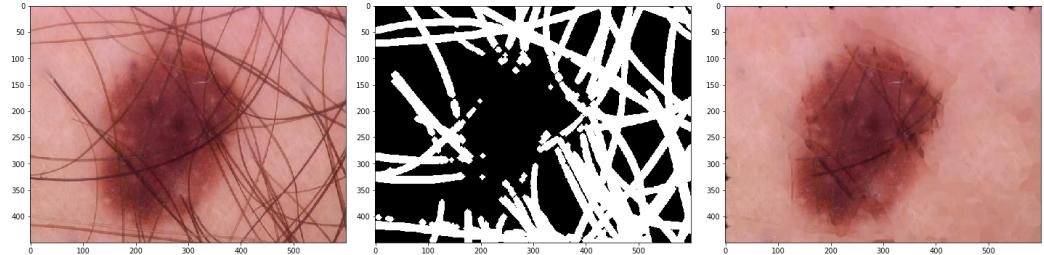
---

- Široko usvojena metoda koja se koristi u prepoznavanju melanoma:
  - A(symmetry) – sa smanjenjem simetrije raste rizik da je mladež melanom
  - B(order) – ivice melanoma su obično nepravilnije i/ili zamućene, često sa usecima i testeraste
  - C(olor) – kod melanoma se često javlja više nijansi jedne ili čak više boja
  - D(imension) – melanomi su obično veći od benignih nevusa
  - E(volution) – označava promenu mladeža tokom vremena
- U tradicionalnom pristupu analiziraju se prve tri karakteristike
- Dimenzije i promenu tokom vremena nije bilo moguće analizirati
- Problem prepoznavanja melanoma i dalje nije trivijalan

# Tradicionalan pristup



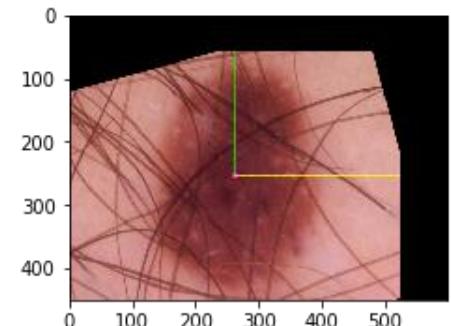
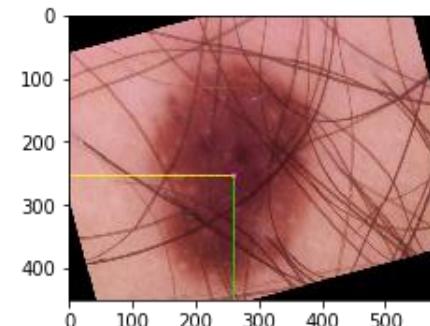
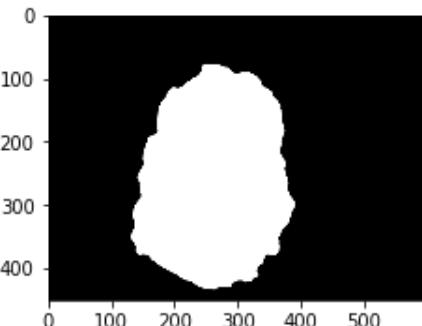
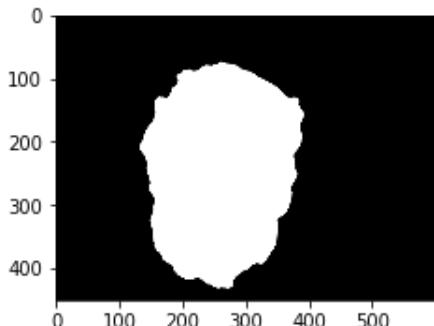
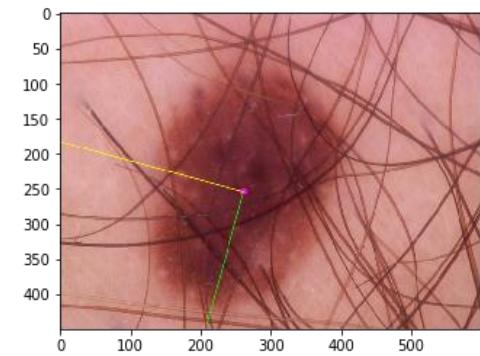
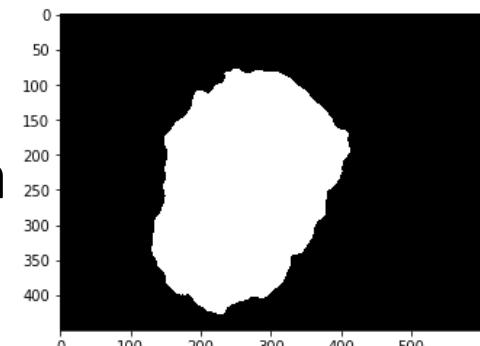
- Preprocesiranje: *Dull Razor*



- Segmentacija: najveća kontura
- Ekstrancija obeležja: A, B, C
- SVM i K-NN

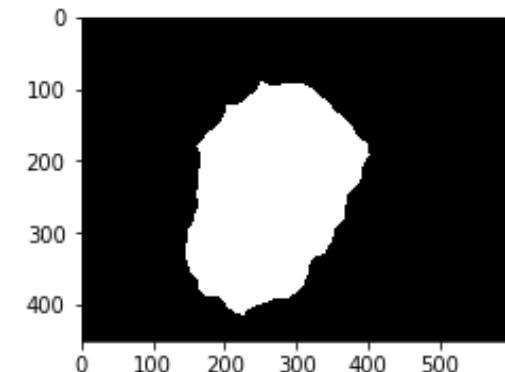
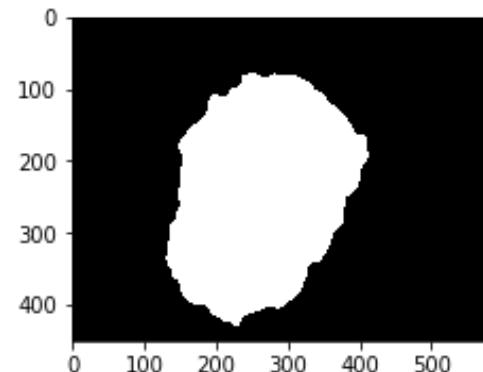
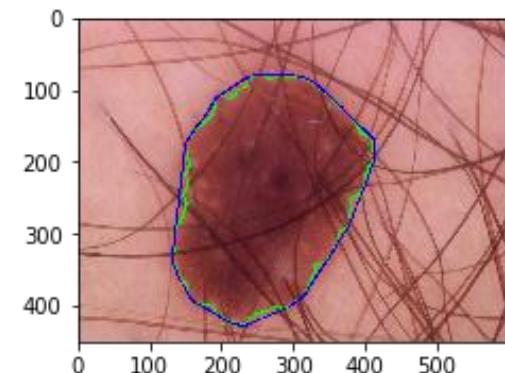
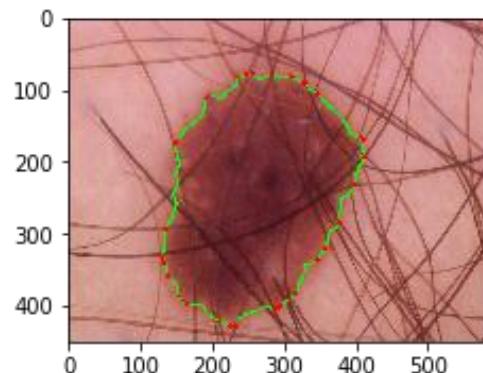
# Ekstrakcija obeležja – simetrija

- Koristi se binarna slika mladeža na kojoj je ucrtana kontura
- PCA metoda se koristi kako bi se odredila glavna osa konture mladeža
- Slika se rotira tako da glavna osa zaklapa prav ugao sa nekom od ivica
- Potom se slika rotira za 180 stepeni
- Upoređuje se procenat poklapanja belih piksela na slikama pre i posle rotacije za 180 stepeni



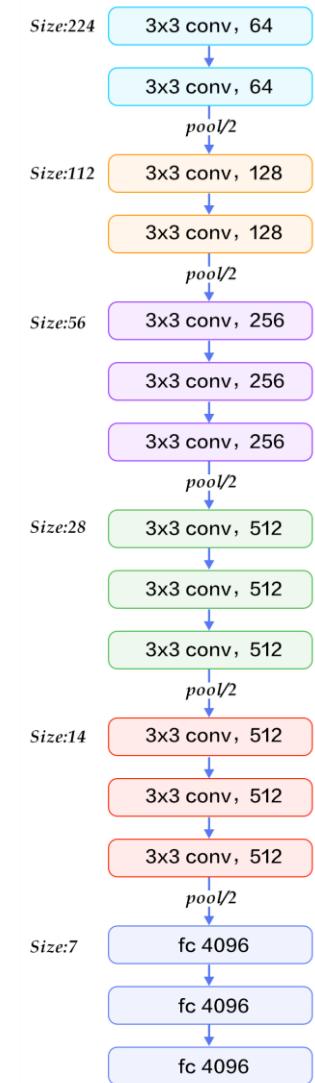
# Ekstrakcija obeležja – ivice i boja

- Ivice se analiziraju opisivanjem konveksnog poligona oko konture mladeža
- Numerička vrednost se dobija kao odnos broja piksela u konturi mladeža i novonastaloj konturi
- Boja se analizira određivanjem devijacije po sva tri kanala
- U obzir se uzimaju pikseli koje obuhvata erodirana kontura



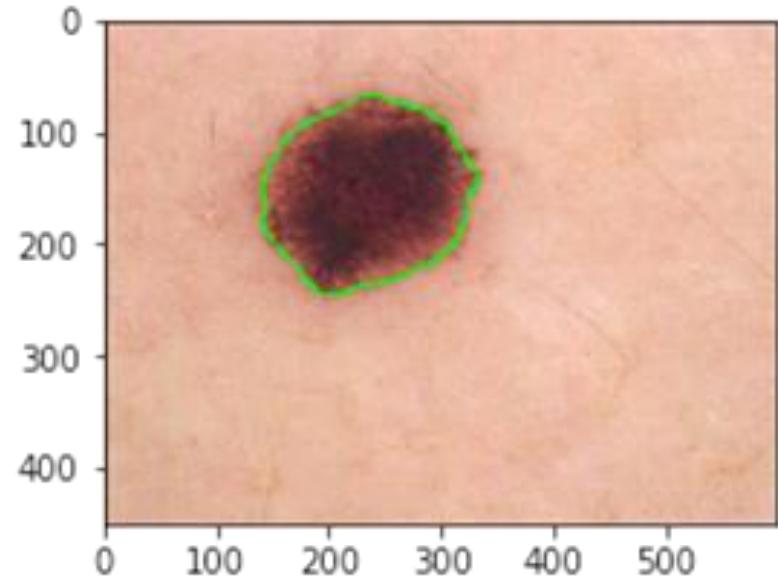
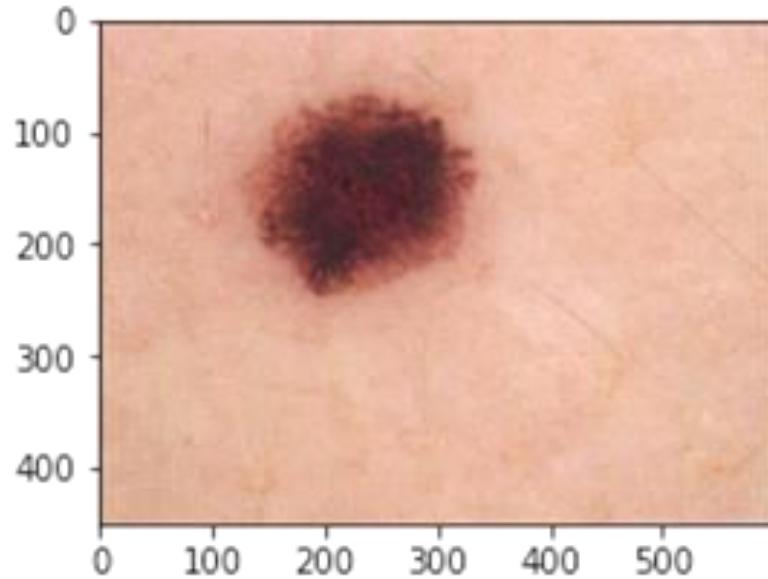
# Rešenje zasnovano na dubokom učenju

- VGG-16 arhitektura trenirana na *ImageNet*
- Transfer learning
  - modifikacija potpuno povezanih slojeva



# Analiza grešaka modela

- Greške su najčešće posledica nedovoljno dobre segmentacije i analize ivica mlađeža

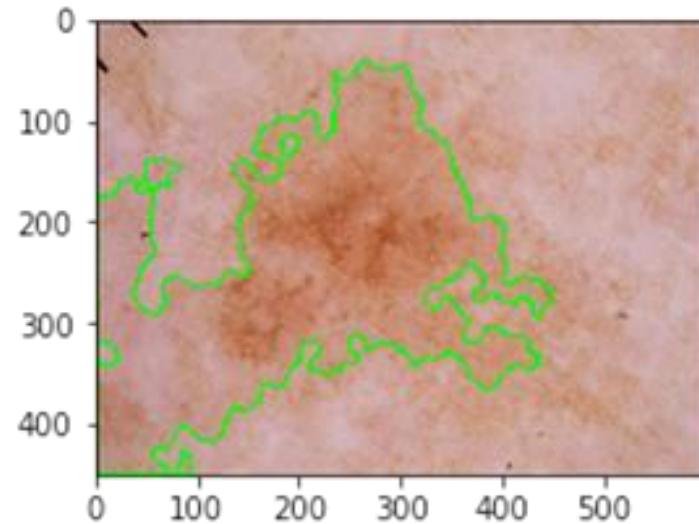
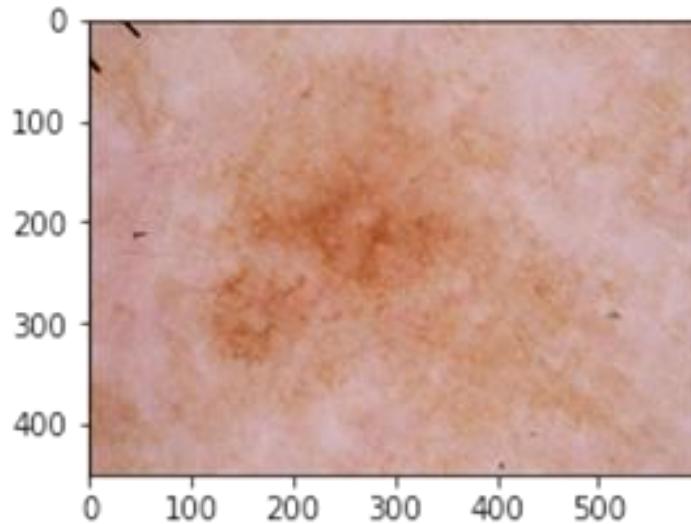


Slika **Error! No text of specified style in document..6**  
Melanom koji je pogrešno klasifikovan.

Ivica je zamućena (što nije analizirano)

# Analiza grešaka modela

- Problem je i segmentacija u slučaju kada su mladeži veoma slični koži
  - Problem i za netrenirane ljudske eksperte



Slika **Error! No text of specified style in document..7**  
Tloše segmentiran melanom.

# Detekcija pospanosti vozača

- Ivana Tomić
- Prikazani rezultati su preuzeti iz diplomskog rada autora
- <https://github.com/ftn-ai-lab/sc-2018-e2/issues/73>

# Motivacija

---

- Umor i pospanost su poznati kao jedan od najvećih neprijatelja bezbedne vožnje
- Pospane osobe imaju smanjenu sposobnost reagovanja na događaje
- Sistem koji detektuje da je vozač pospan bi mogao da pruži zvučno upozorenje



# Metod

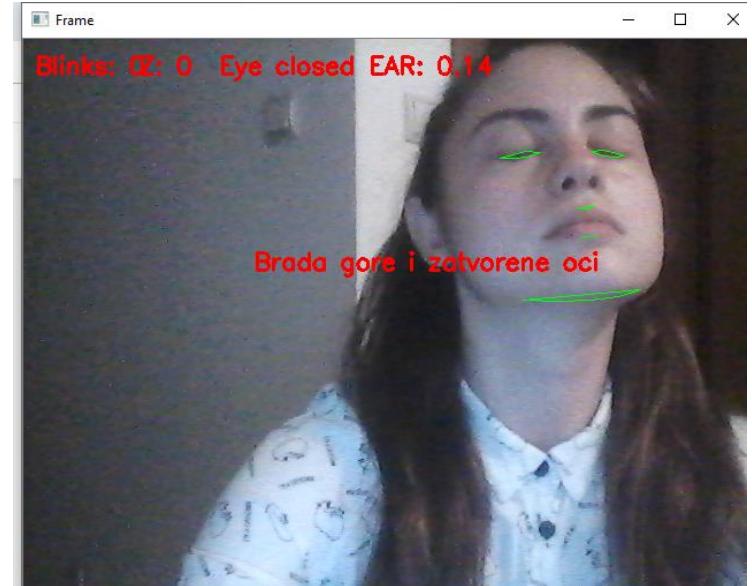
---

- Detekcija relevantnih obeležja u video zapisu
  - Broj zevanja
  - Broj treptanja
  - Položaj glave i očiju
- Izdvajanje frejmova
- Detekcija lica na frejmu
- Određivanje karakterističnih tačaka lica
- Klasifikacija pospanosti (SVM)

# Metod

## Ugrađena upozorenja:

- Vozač drži zatvorene oči više od dve sekunde
- Položaj glave: pomeranje brade ne gore uz sklopljene oči
- Zevanje više od 5 puta u poslednja 3 minuta



- Skup podataka
  - Kreiran od strane autora
    - Položaj kamere, ponašanje ljudi, osvetljenost, dužina snimka
  - Dodati snimci preuzeti sa interneta