

Obrada slike

Jednostavne operacije nad slikama

Jednostavne operacije nad slikama

- Slike smo reprezentovali kao matrice brojeva
- Ovo nam omogućava jednostavne operacije nad slikama kao što su:
 - Sabiranje
 - Oduzimanje
 - Logičke operacije
 - Afine transformacije
 - ...
- Ove operacije su veoma jednostavne i brze, a imaju širok spektar primena

Jednostavne operacije nad slikama

- Za dve ulazne slike f_1 i f_2 i datu funkciju Op :

$$g(x, y) = Op(f_1(x, y), f_2(x, y))$$

operator se primenjuje na odgovarajuće parove piksela ulaznih slika (*pairwise*). Ulazne slike moraju biti iste veličine

- Pseudokod:

```
for all pixel positions x, y:  
    out[x, y] = func( image1[x, y] , image2[x, y]  
)
```

Sabiranje

$$g(x, y) = f_1(x, y) + f_2(x, y)$$



+



=

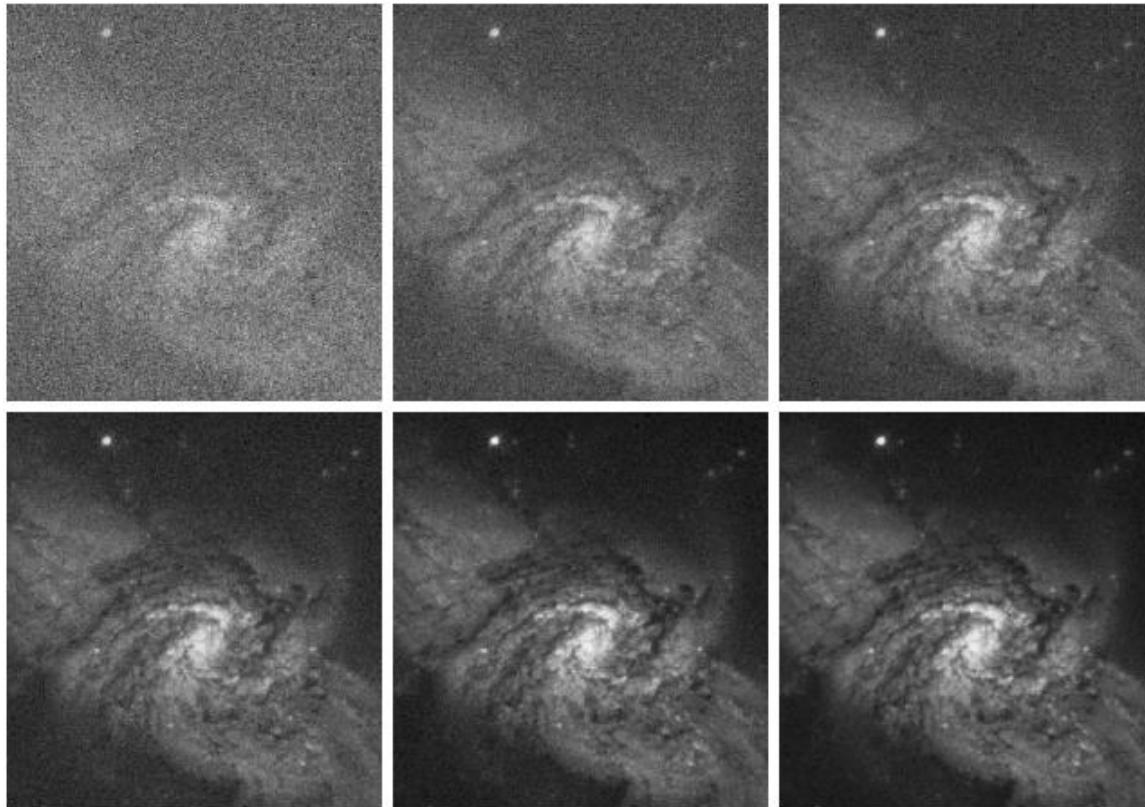


- Može se raditi i *weighted blend*:

$$g(x, y) = \alpha_1 f_1(x, y) + \alpha_2 f_2(x, y)$$

Sabiranje

- NASA je snimila sliku iste galaksije više puta
 - Svaka slika sadrži šum
 - Slike su poravnate (nije menjan ugao kamere)



- Sabiranjem slika postaje sve jasnija
 - Scena ostaje ista, a šum se menja
 - Kada sabiramo dodajemo uvek isti broj + slučajni šum
 - Ovo smanjuje šum

Oduzimanje

- Zgodno za pronalaženje razlike između dve slike:

$$g(x, y) = f_1(x, y) - f_2(x, y)$$

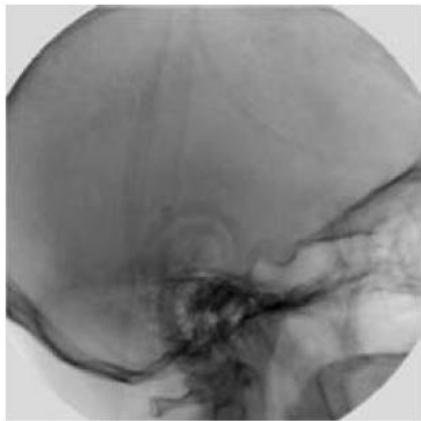


- Pošto bi vrednosti piksela rezultujuće slike trebali biti nenegativni, možemo:
 - Tražiti apsolutnu razliku: $g(x, y) = |f_1(x, y) - f_2(x, y)|$
 - Shift piksela tako da se najniže vrednosti mapiraju na 0

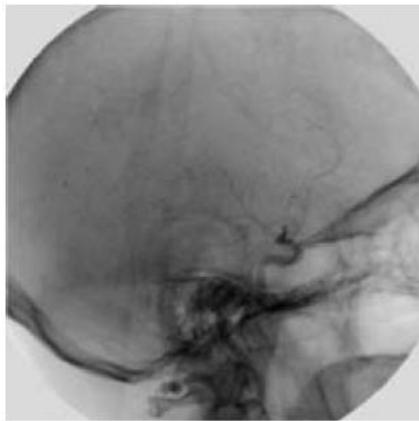
Oduzimanje (maska)

- Imamo dve varijante iste scene, ali snimljene pod različitim uslovima

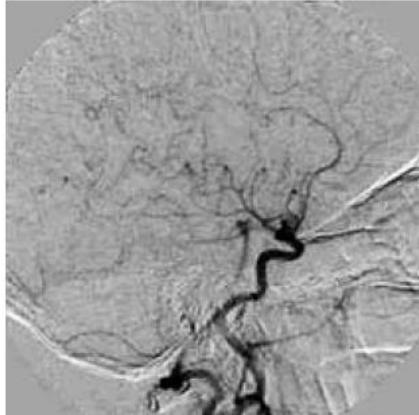
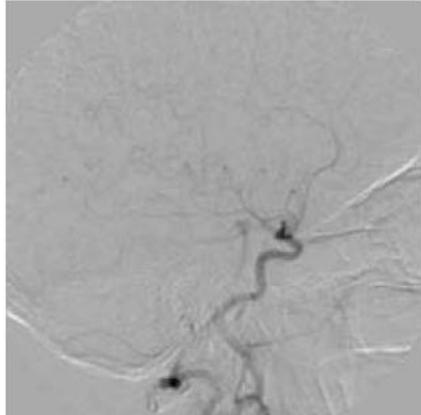
I_1



I_2



$I_1 - I_2$



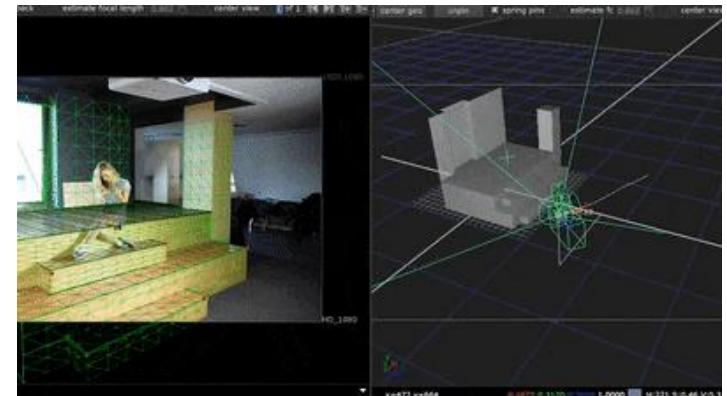
- Pre (I_1) i posle (I_2) ubrizgavanja jodnog kontrastnog sredstva u krvotok
- Poslednja slika: nakon pobojšanja kontrasta slike $I_1 - I_2$
- Dobijamo jasnu mapu kako se sredstvo propagira kroz krvne sudove mozga

Oduzimanje

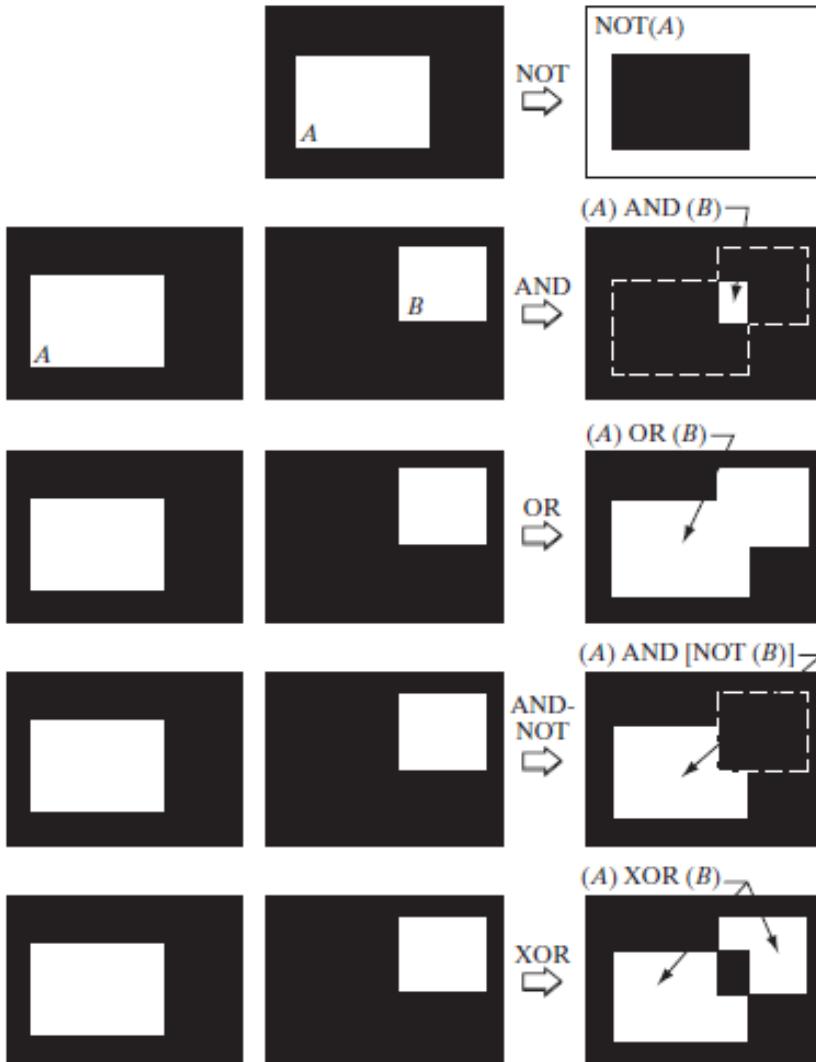
- Detekcija pokreta u inače statičnoj sceni (stacionarna kamera)
 - Osnova *motion tracking* tehnika
 - Nadgledanje, praćenje vozila na putu,...
- Praćenje pokreta kamere
 - Deo većeg procesa u snimanju filmova koji se zove *match move*
 - Ubacivanje kompjuterski generisane grafike u stvarne scene sa pokretnom kamerom
 - Odgovarajuća pozicija, skaliranje, orientacija...
- Video kompresija
 - Enkodirati samo razlike između frejmova
 - Detekcija/predikcija pokreta se koristi u video kompresijama poput MPEG



[https://www.pyimagesearch.com/2015/05/25/
basic-motion-detection-and-tracking-with-
python-and-opencv](https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv)



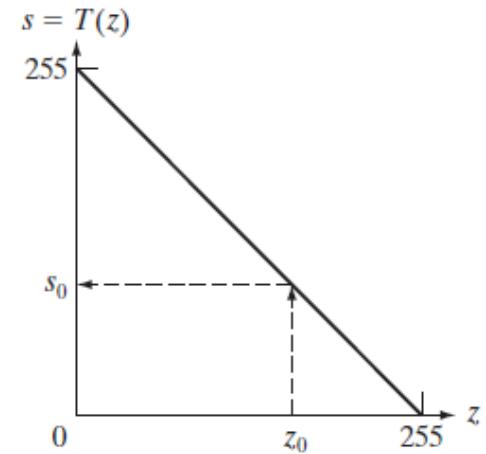
Logičke operacije



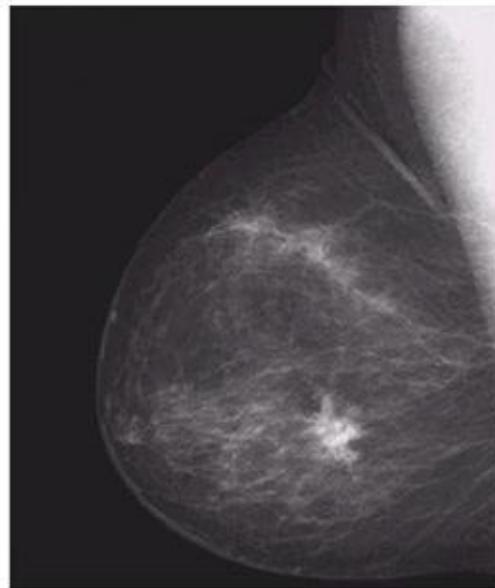
- Prikazane su osnovne logičke operacije nad binarnim slikama
- 0 – crni pikseli
- 1 – beli pikseli
- Primena: kombinovanje slika, segmentacija regija od interesa (primena maske), *green screening*

Negativ

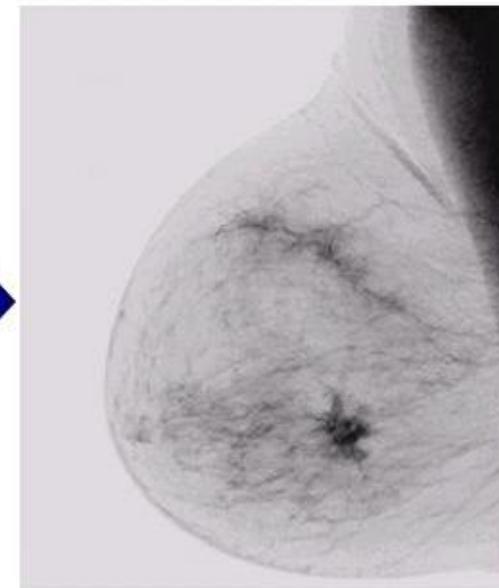
- Možemo invertovati sliku: $0 \rightarrow 255, 255 \rightarrow 0$
- Negativi su korisni da se pojačaju beli ili sivi detalji na crnoj pozadini



Original
Image



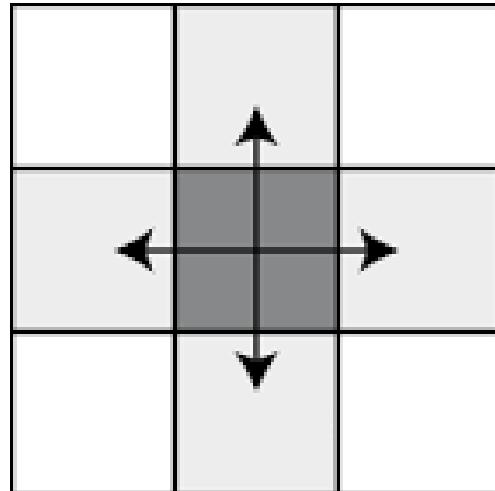
$$s = 1.0 - r$$



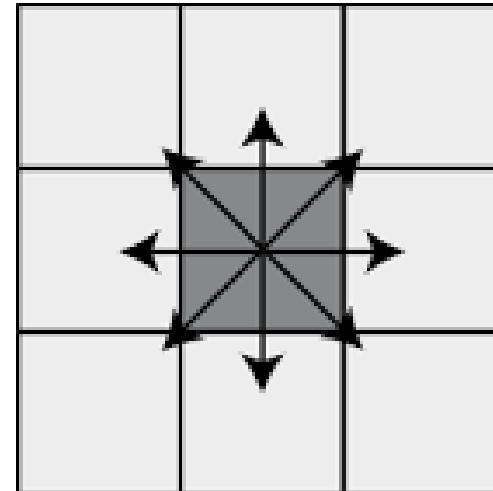
Negative
Image

Pojam susedstva

- Sliku smo reprezentovali putem matrice
- Možemo definisati susedstvo piksela
- Tipično, sa aspekta prostorne povezanosti, koristimo 4-susedstvo ili 8-susedstvo



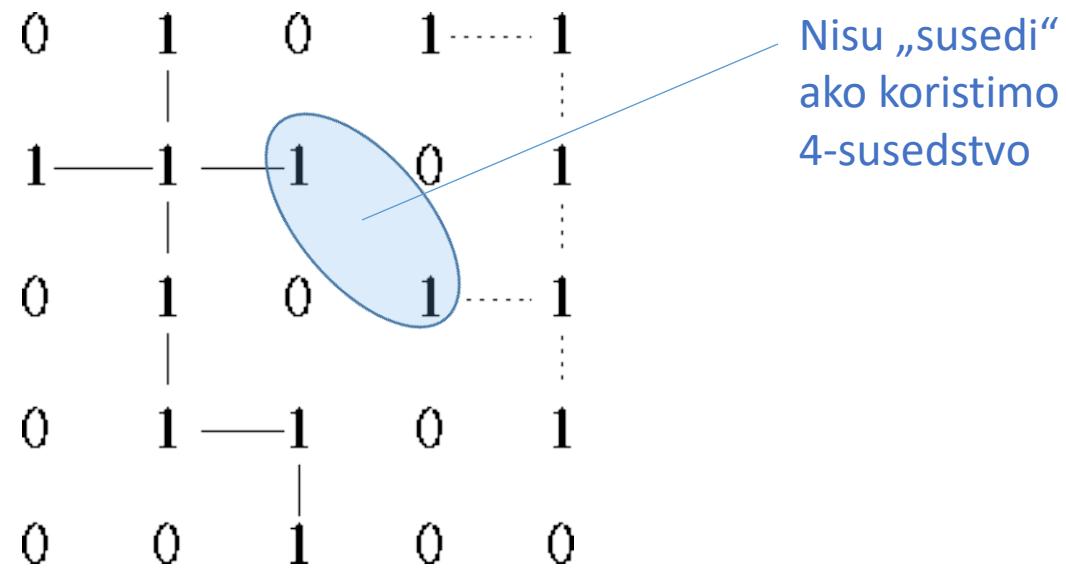
4-Connectivity



8-Connectivity

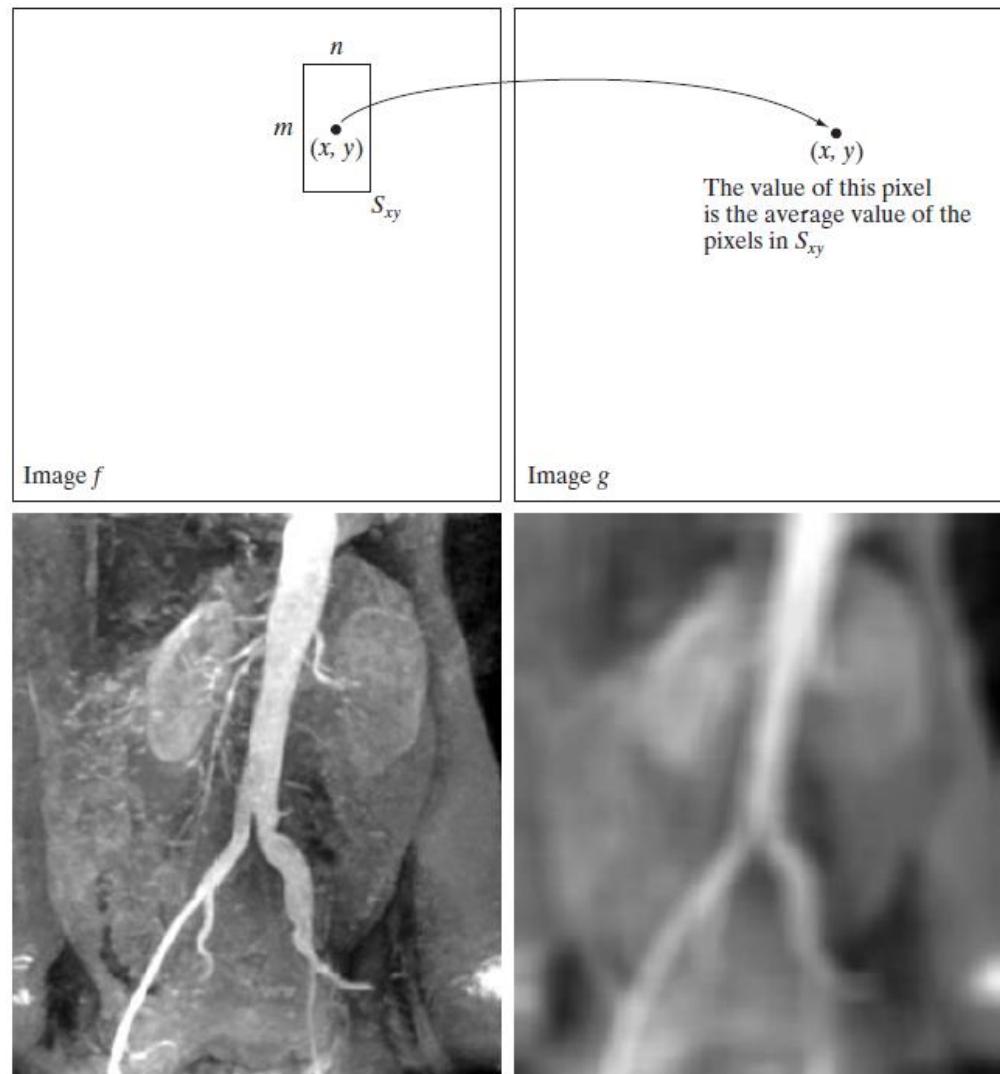
Povezanost piksela

- Pojam susedstva možemo definisati i za grupe piksela
 - Recimo, na slici su prikazane dve grupe piksela koje smatramo „susedima“ ako koristimo 8-susedstvo
- Rezultati će se razlikovati u zavisnosti od korišćenog pojma susedstva
 - Na primer, ako koristimo 4-susedstvo, grupe na slici ne smatramo „susedima“

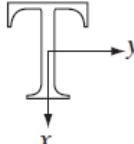
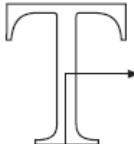
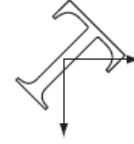
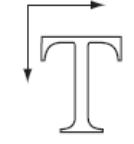
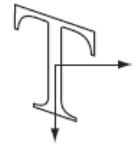
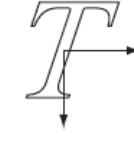


Lokalno uprosečavanje

- Vrednost piksela zamenjujemo prosečnom vrednošću njegovog susedstva
 - Zavisi od definicije susedstva
- Uklanjamo šum, ali dobijamo zamućeniju sliku
 - Ima sofisticiranijih metoda koje prepoznaju gde su veće promene i ne koriste piksele sa druge strane oblasti koja predstavlja promenu

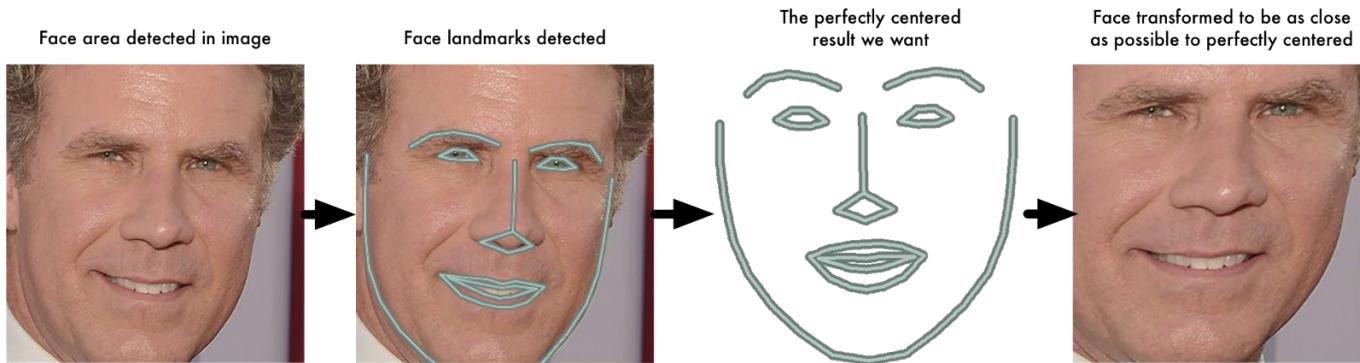


Affine transformacije

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \cos \theta + w \sin \theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

$$[x \ y \ 1] = [v \ w \ 1] \mathbf{T} = [v \ w \ 1] \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

Affine transformacije

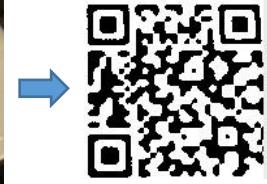


Rotacija lica: radi pojednostavljenja prepoznavanja

<https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>

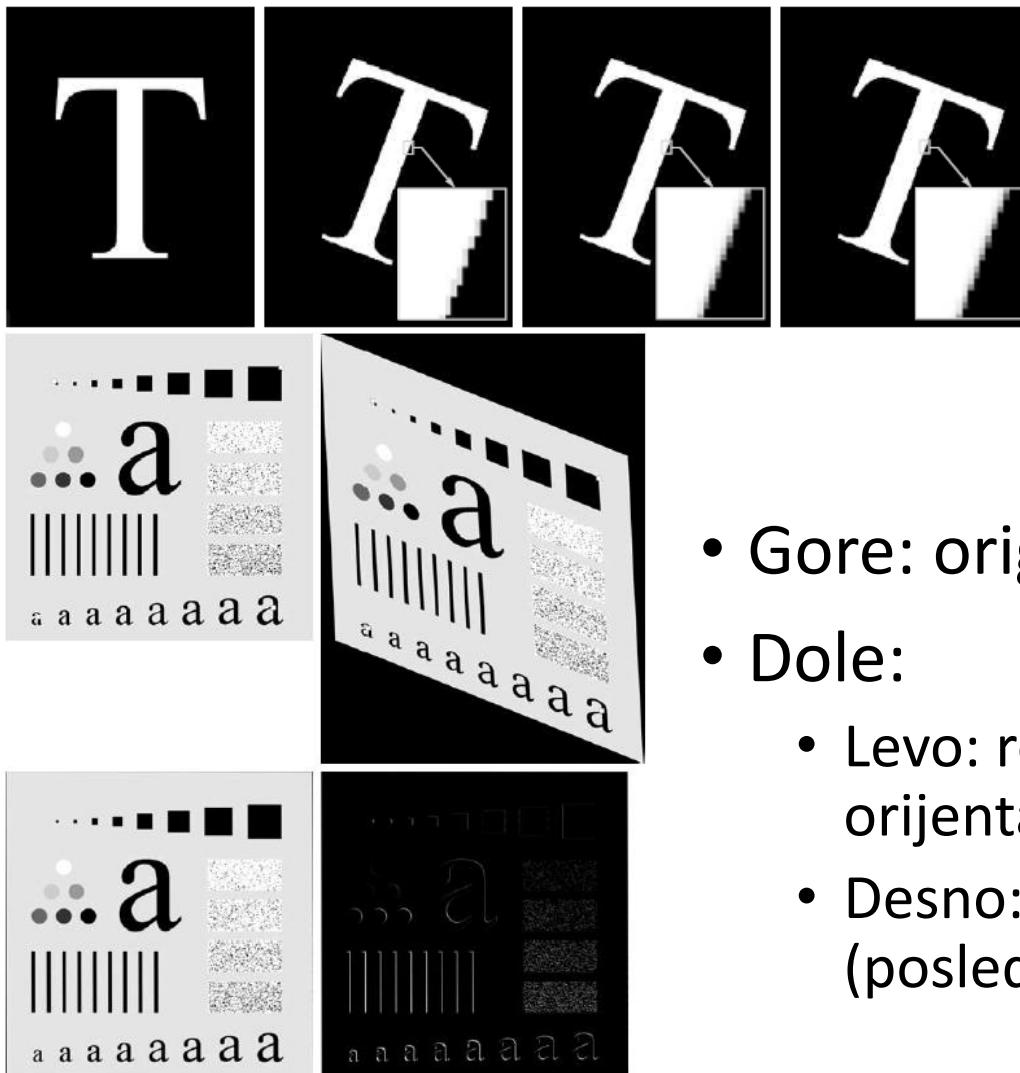


Isti otisak, ali različita pozicija i veličina – teško za poređenje



<http://eric-yuan.me/simple-qr-code-scanner-1/>

Rotacija – problem diskretizacije



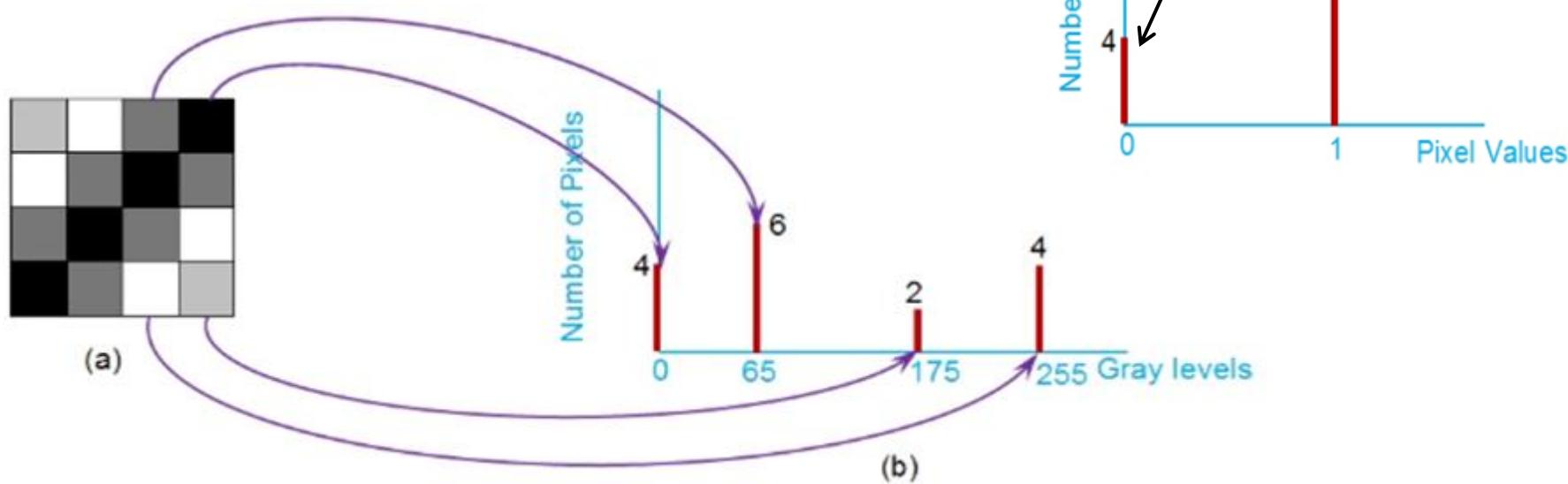
- Gore: originalna slika i rotirana slika
- Dole:
 - Levo: rotirana slika je rotirana na istu orijentaciju kao ranije
 - Desno: razlika originalne slike i leve (posledice diskretizacije)

Statističke osobine slike

- Predstavljaju sažetak sadržaja slike koji pomaže njenoj interpretaciji

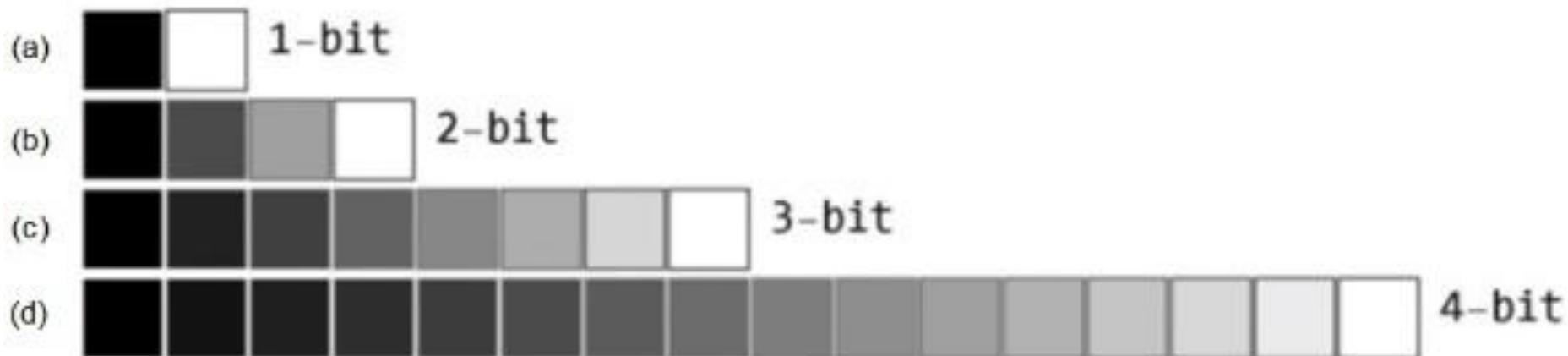
Histogram

- Histogram predstavlja grafički prikaz brojnosti piksela datog intenziteta
 - Na x osi se predstavlja intenzitet piksela
 - Za svaki intenzitet na x osi prebrojimo koliko ima piksela tog intenziteta – visina stupca prikazanog na histogramu će biti proporcionalna tom broju



Histogram monohromatske (grayscale) slike

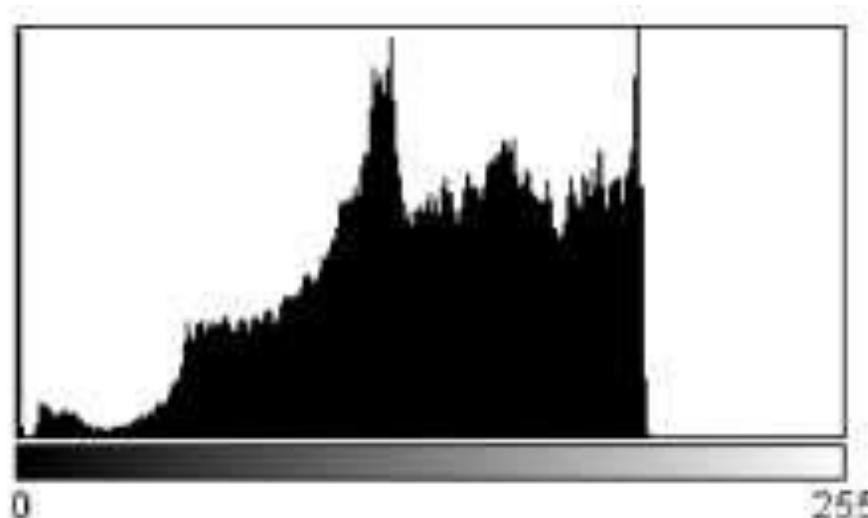
- X-osa: vrednosti intenziteta piksela
 - Broj bita neophodan da predstavi sliku → broj nijansi koje je moguće predstaviti
 - N bitova → 2^N mogućih vrednosti piksela u opsegu $[0, 2^N - 1]$



Histogram monohromatske (grayscale) slike



Histogram je grafička reprezentacija količine svakog nivoa sive u slici

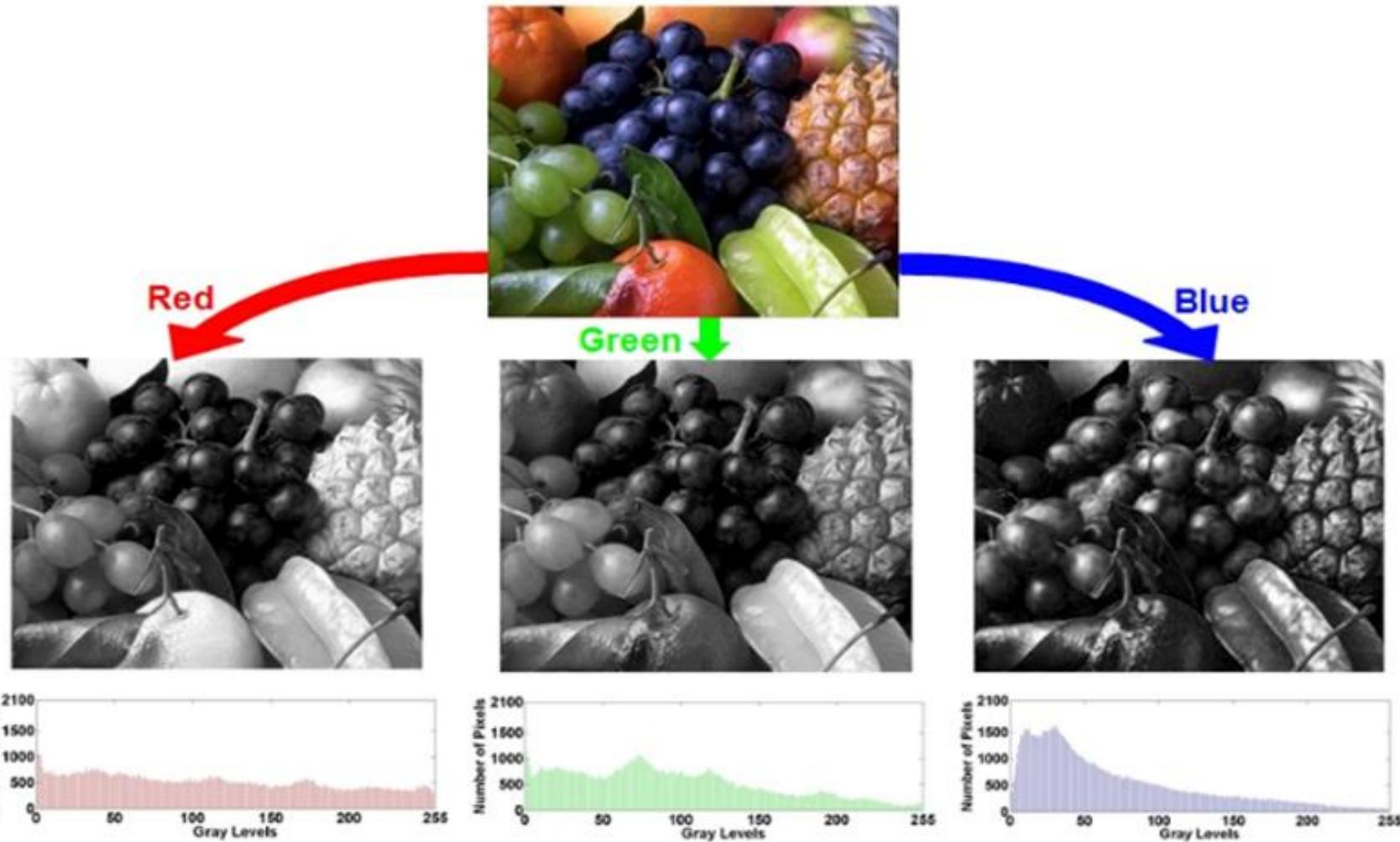


Count: 179463
Mean: 126.565
StdDev: 43.800

Min: 0
Max: 213
Mode: 192 (2241)

Primer histograma slike u boji

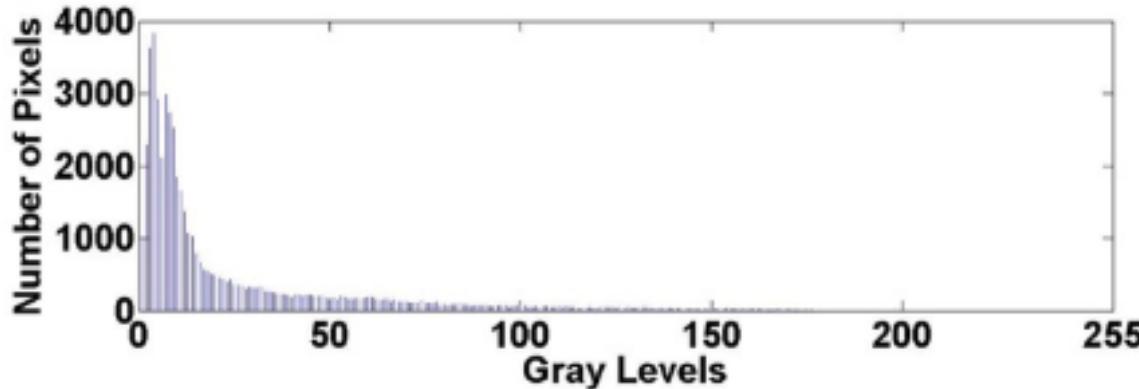
- Za slike u boji, moguće je analizirati svaki kanal zasebno



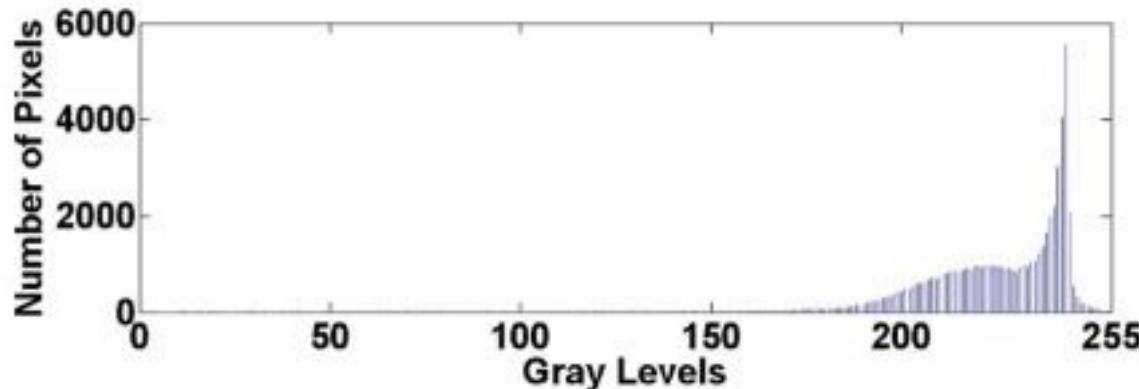
Šta možemo saznati sa histograma?

- Raspon vrednosti na histogramu: osvetljenje
 - Prosečna vrednost nam govori o osvetljenosti slike
 - *min* i *max* nam govore o rasponu intenziteta koji je snimljen
- Standardna devijacija: kontrast
 - Visoka: slika nešto predstavlja (barem u nekom svom delu)
 - Niska: monotona sliku

Osvetljenje slike



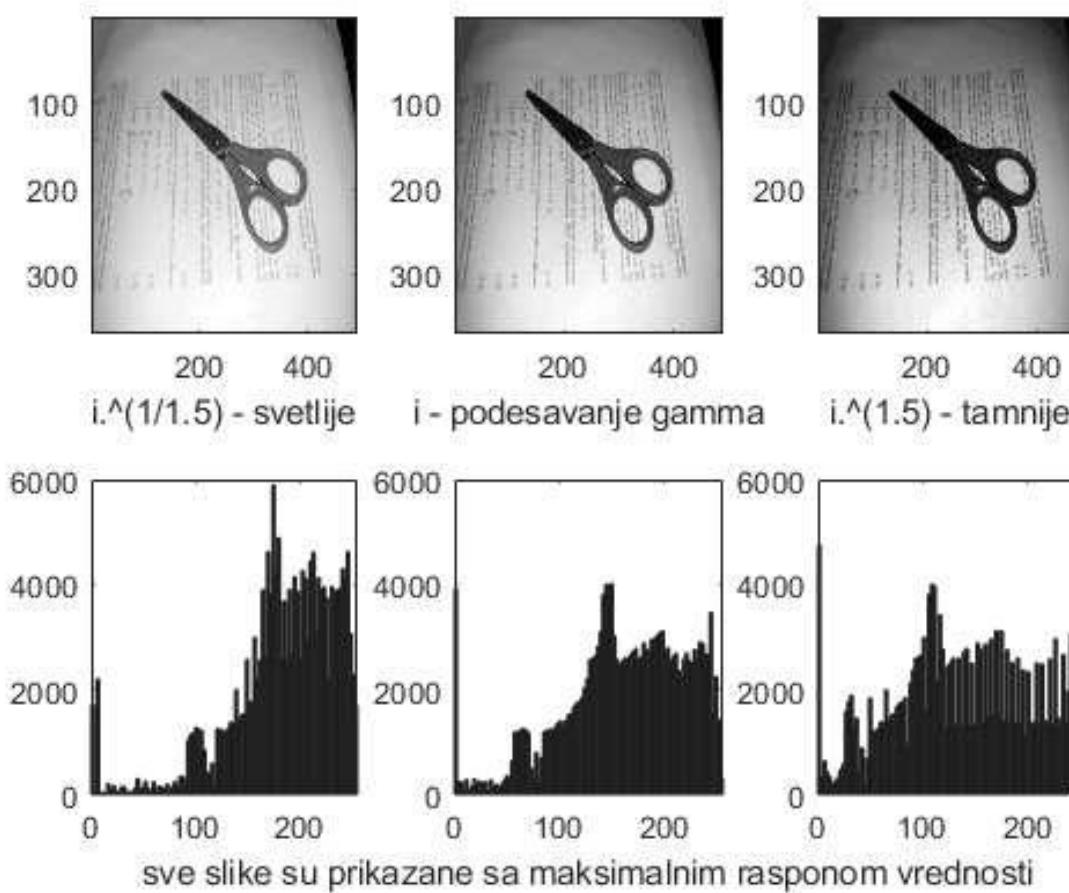
Nisko osvetljenje: vrednosti histograma su koncentrisane uglavnom na levoj strani



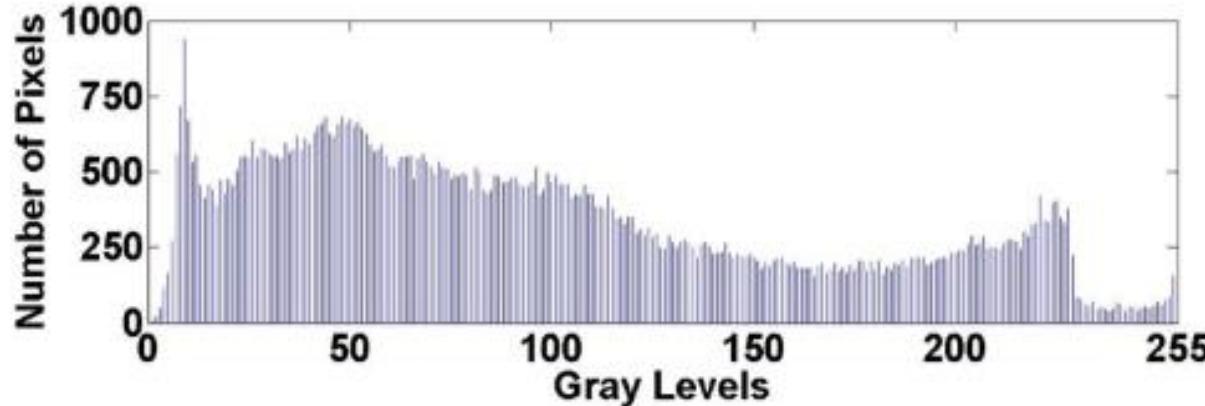
Visoko osvetljenje: vrednosti histograma su koncentrisane uglavnom na desnoj strani

Osvetljenje slike

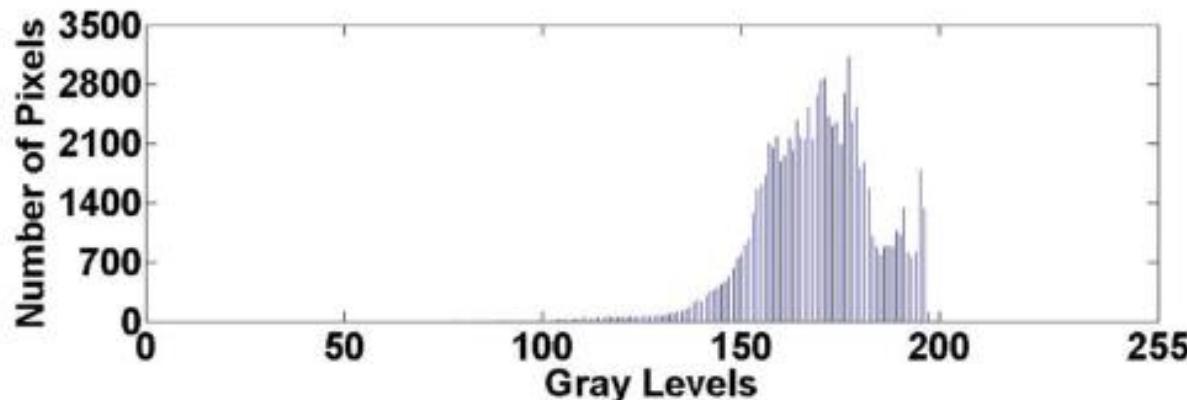
- Ako ima previše piksela sa visokim ili niskim intenzitetom, tada je obično kontrast slike nizak i slika je previše svetla ili previše tamna



Kontrast



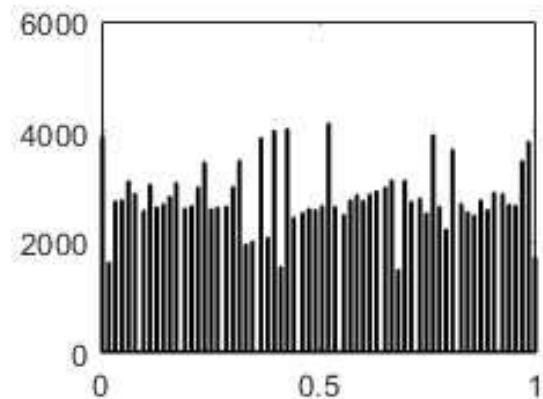
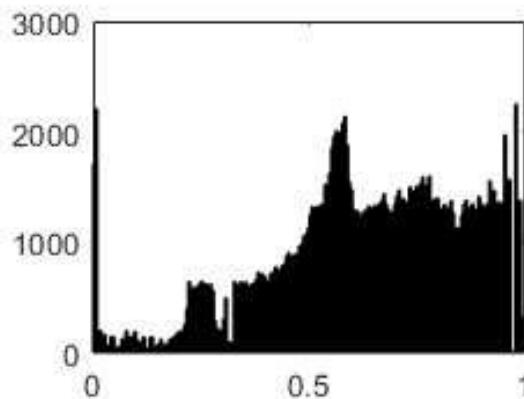
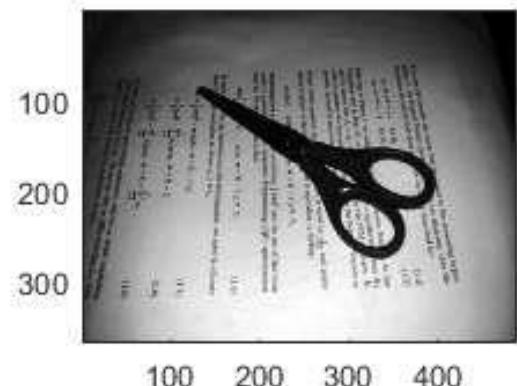
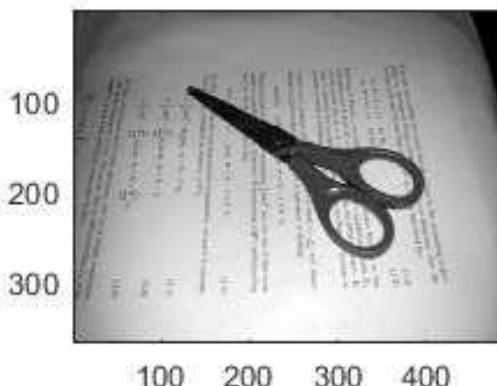
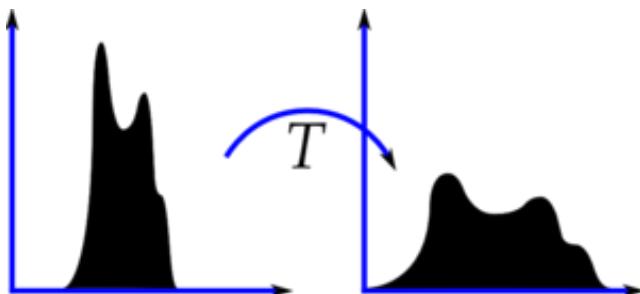
Vrednosti ravnomerno raspodeljene po intenzitetima \rightarrow dobar kontrast u osvetljenju



Koncentracija vrednosti oko jednog određenog intenziteta \rightarrow loš kontrast u osvetljenju

Poravnavanje histograma (*Histogram Equalization*)

- Modifikujemo distribuciju intenziteta piksela tako da postoji otprilike jednak broj piksela za svaki od intenziteta

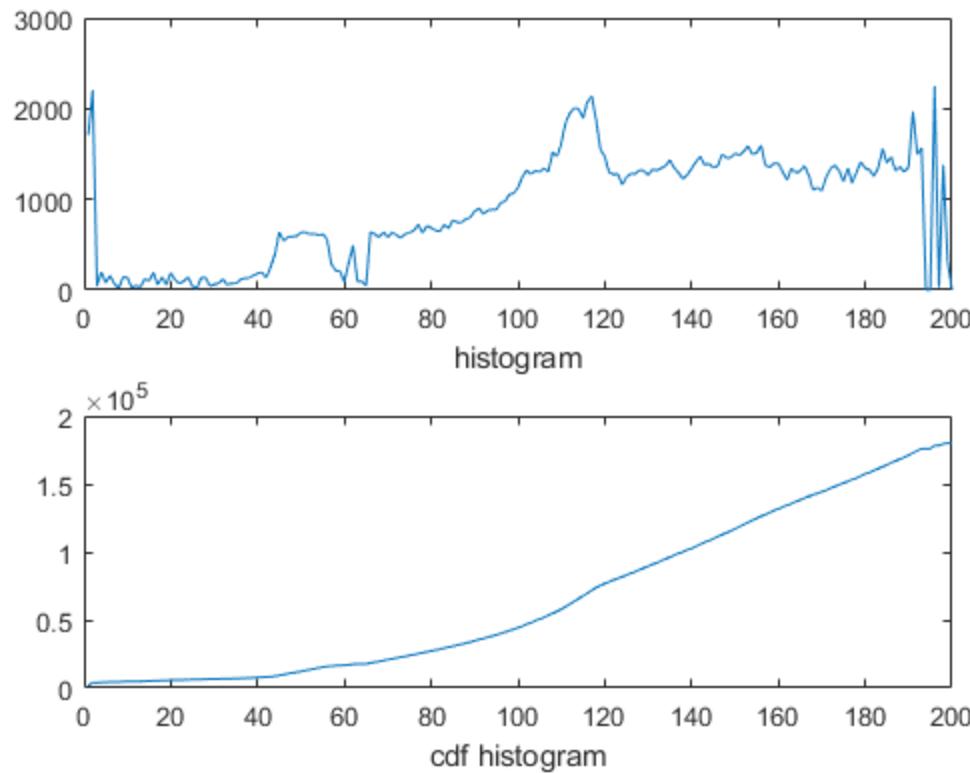


Poravnavanje histograma

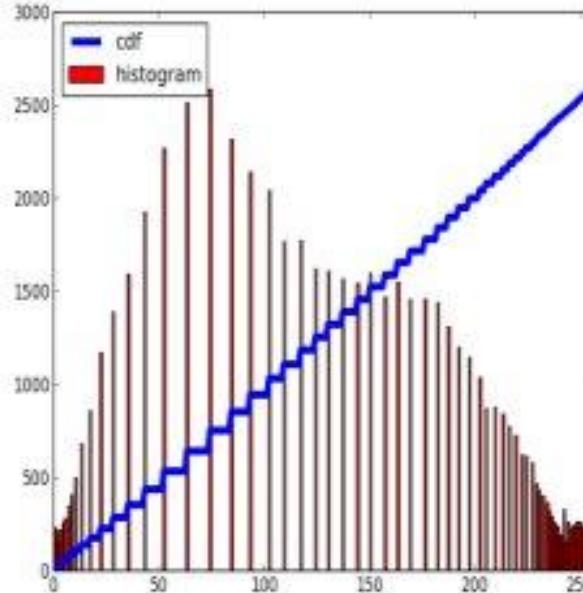
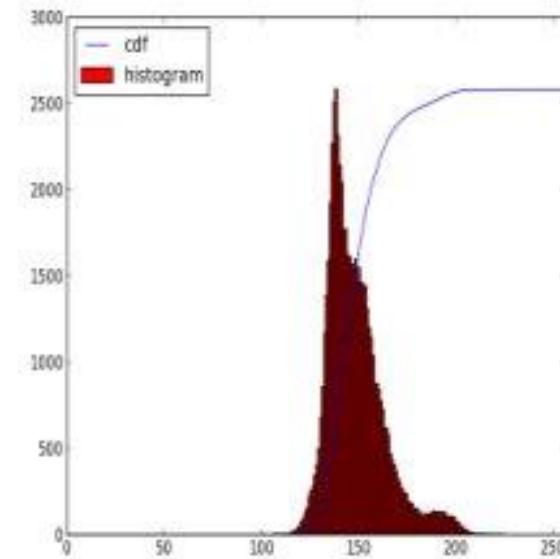
- Kumulativna funkciju gustine (CDF) histograma

$$cdf(x) = \sum_{k=-\infty}^x P(k),$$

k – intenzitet piksela, $P(k)$ – verovatnoća pojave piksela intenziteta k



Poravnavanje histograma



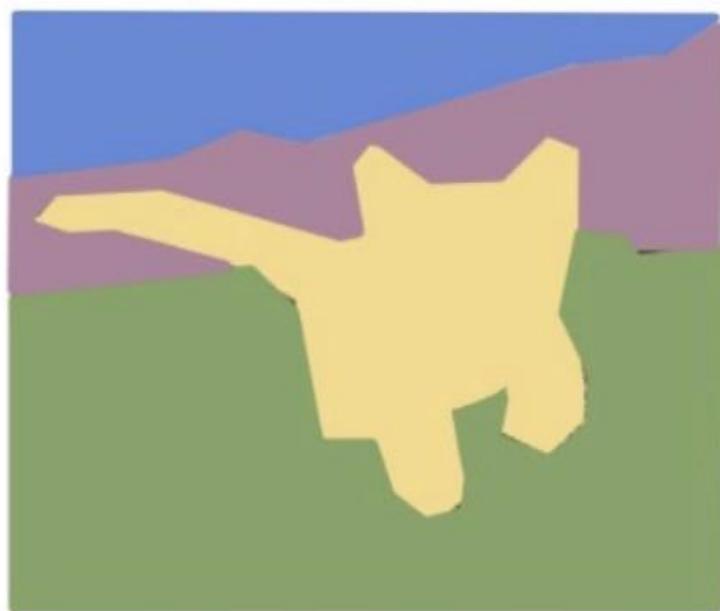
Poravnavanje histograma

- Veoma korisno za slike gde su i pozadina i objekti od interesa previše svetli ili previše tamni
- Recimo, ovo pomaže kod rendgena da se bolje vidi struktura kostiju i da se pobojšaju slike koje su *overexposed* ili *underexposed*
- Reverzibilna operacija
- Mana je što može da poveća kontraste na pozadini, a smanji koristan signal

Segmentacija slike

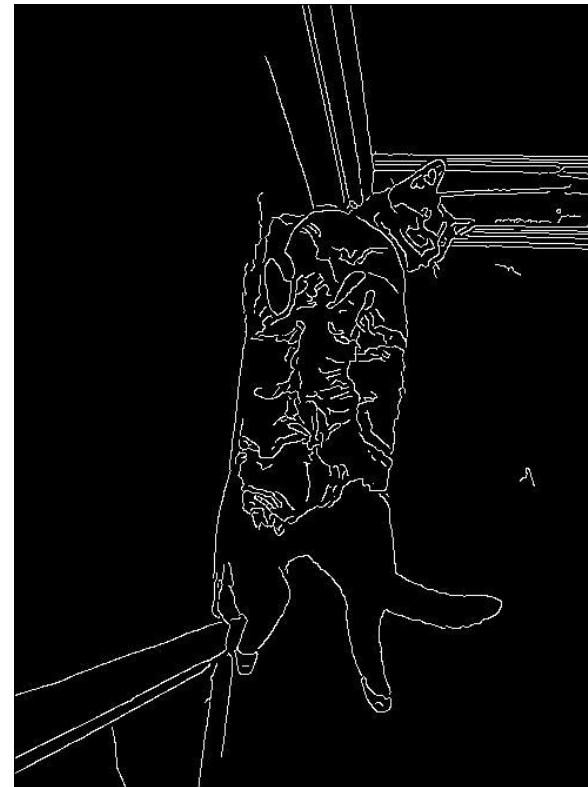
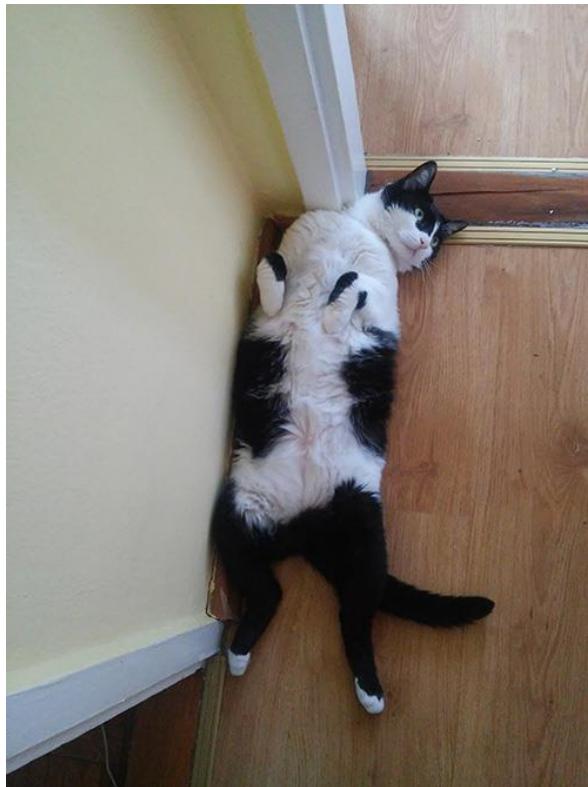
Segmentacija slike

- Predstavlja postupak podele slike na regije sa sličnim atributima, npr.
 - Izdvajanje objekta sa pozadine (*foreground/background separation, background subtraction*)
 - Semantička anotacija delova slike (klasifikacija piksela slike u odgovarajuće kategorije)

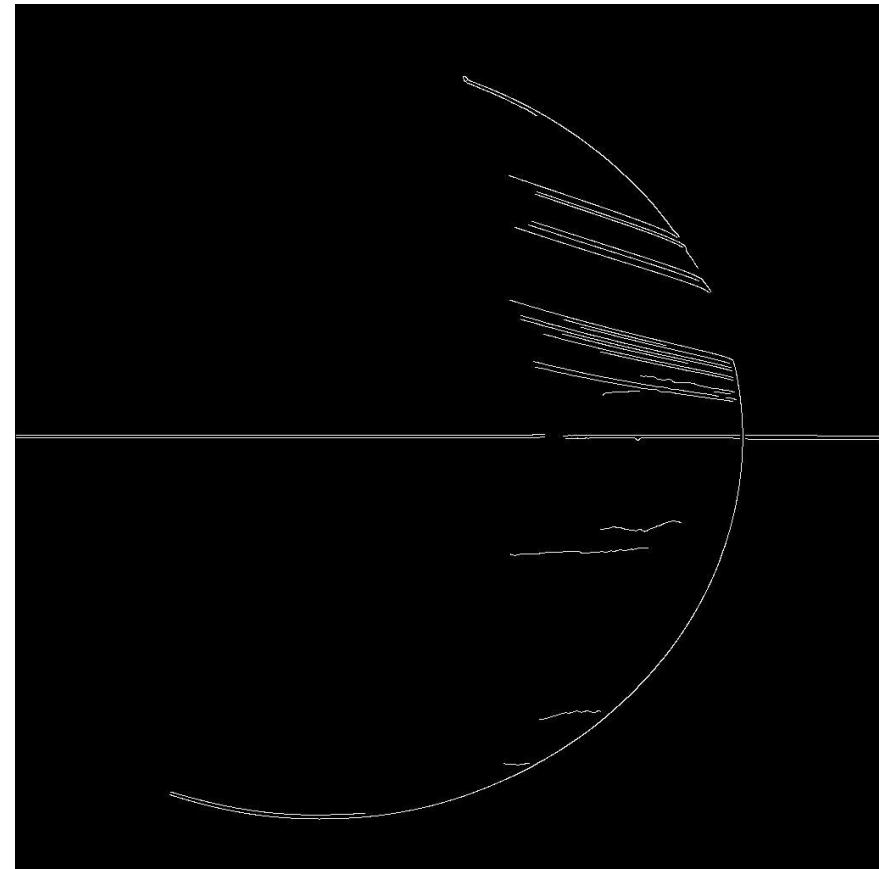
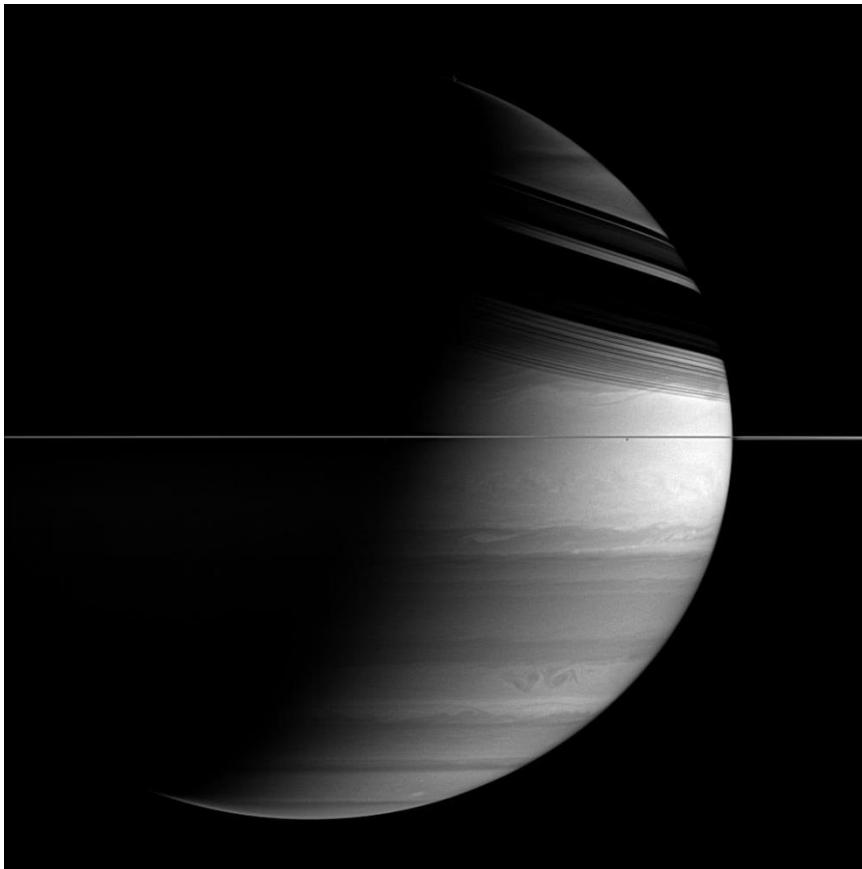


Na osnovu čega segmentišemo?

- Koji tip informacija (atribute) možemo koristiti za segmentaciju?
 - Osvetljenost, boja, ivice, mere teksture,...



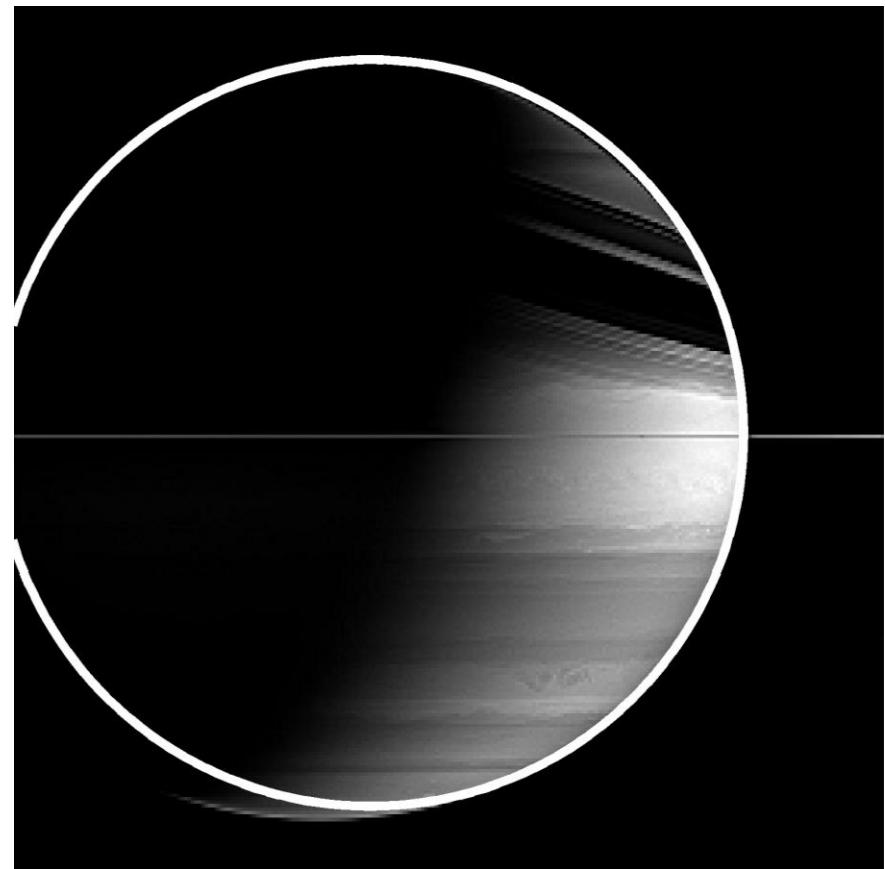
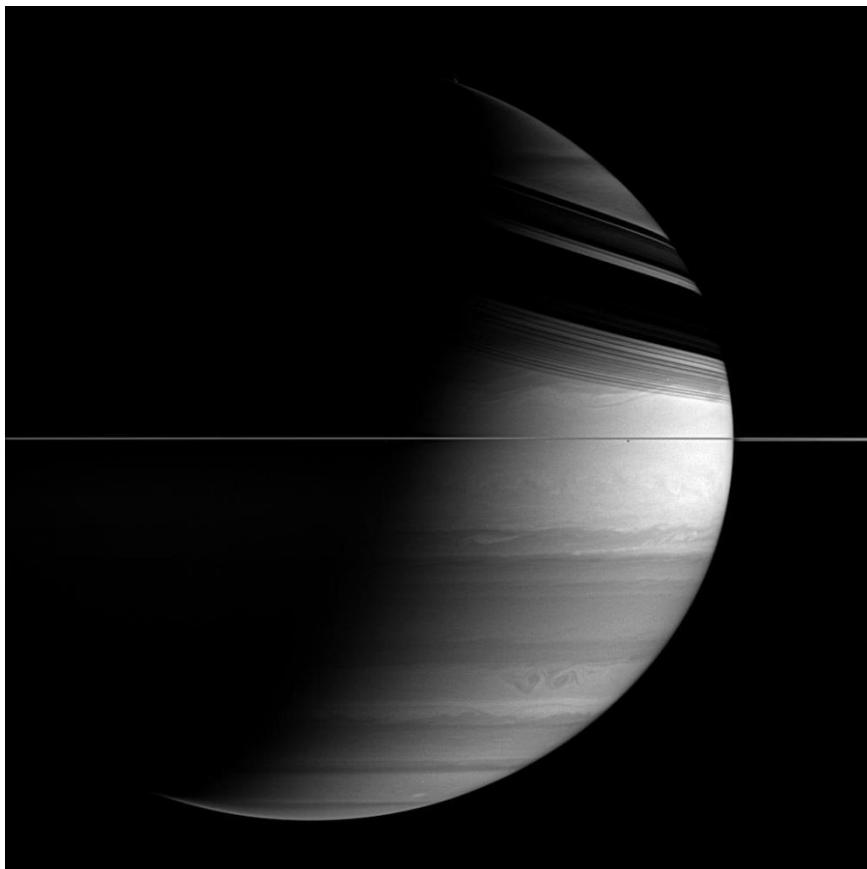
Na osnovu čega segmentišemo?



- Ivice su važan atribut, ali nisu uvek dovoljne
- Dobro je obuhvatiti više izvora informacija

Na osnovu čega segmentišemo?

- Hough pronalaženje krugova:



Segmentacija slike

- Kako znamo da je segmentacija dobra?
 - Pikseli iste kategorije (sličnih vrednosti) formiraju povezani region
 - Susedni pikseli u različitim kategorijama i imaju različite vrednosti
- Segmentacija je najčešće kritičan korak u analizi slike jer je to tačka u kojoj se prelazi sa piksela na objekat kao jedinice obrade
- Ako je segmentacija uspešna, ona će olakšati ostale korake analize slike

Segmentacija slike

- Gruba podela pristupa:
 - Histogram (određivanje praga)
 - Klasterovanje (poseban termin predavanja)
 - Praćenje ivica (morphološki pristupi, aktivne konture,...)
 - ...

Segmentacija slike pomoću praga

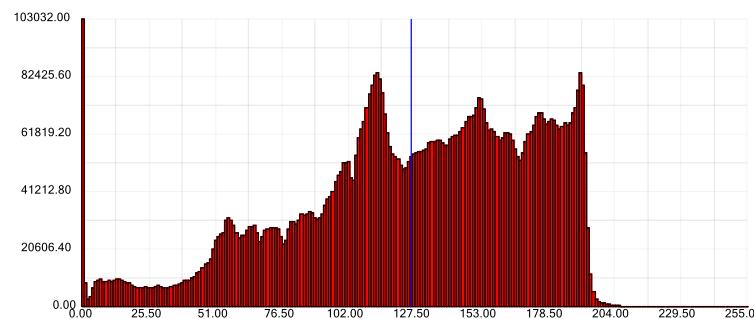
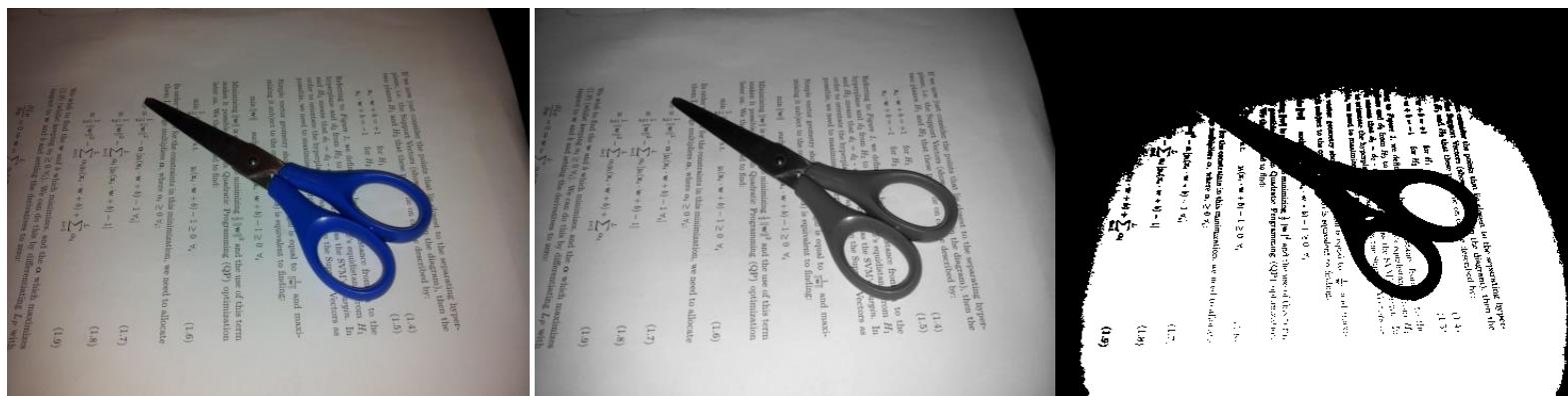
- Najjednostavniji način za segmentaciju slike je segmentacija pomoću praga (*thresholding*)
- Poredimo osvetljenost piksela sa pragom (ili više pragova) i svrstavamo piksele u dve (ili više) kategorija

Segmentacija pomoću globalnog praga

- Najjednostavniji vid segmentacije pomoću praga je da poredimo vrednosti piksela sa samo jednim pragom

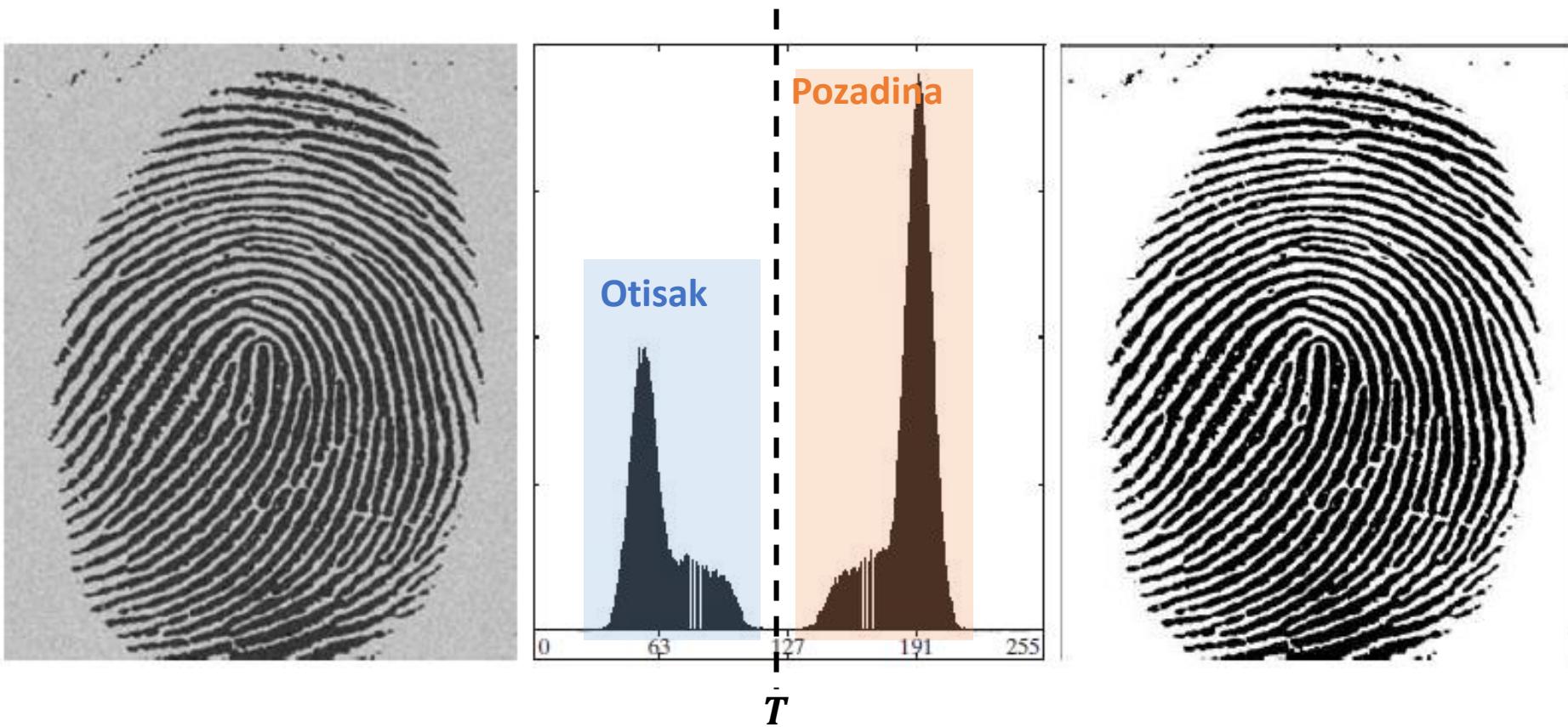
1. Sliku čemo pretvoriti u *grayscale*
 2. Odabraćemo vrednost praga T
 3. Klasifikujemo slike prema intenzitetu

$$B[x, y] = \begin{cases} 0, & f(x, y) \leq T \\ 1, & f(x, y) > T \end{cases}$$

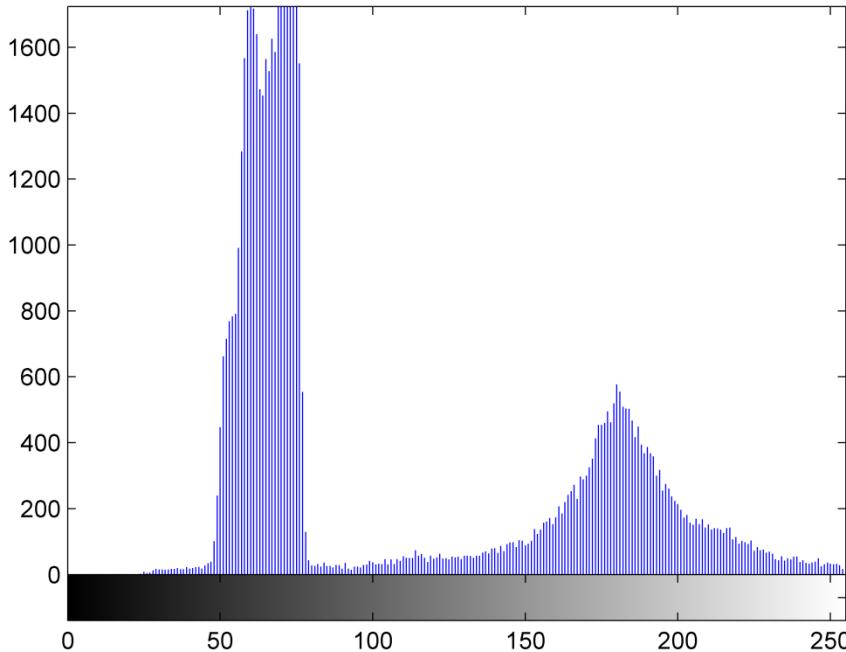
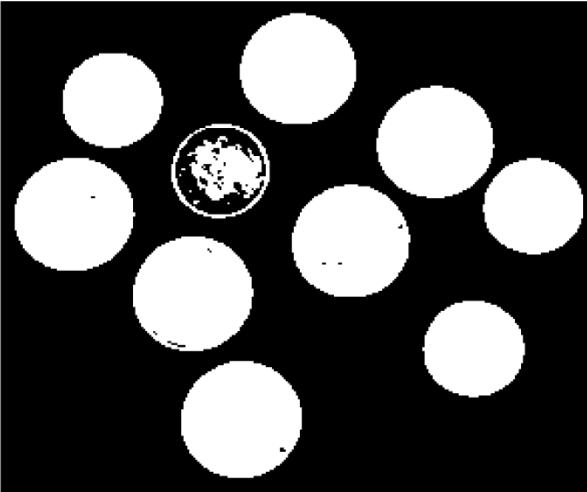


Kako odrediti vrednost praga?

- Otsu: auto pronalazi prag na osnovu vrednosti histograma
- Ideja je prikazana na slici:
 - Multimodalni histogram: jasno razdvojeni intenziteti linija otiska i pozadine
 - Optimalan prag je obično u „dolinama“ histograma



Otsu – prag je u “dolini” histograma



- Veoma jednostavno
- Potpuno automatski

Otsu – prag je u “dolini” histograma

- Imamo dve klase: 0-pozadina, 1-objekat od interesa
- Otsu minimizuje *weighted within-class variance*:

$$\sigma_w^2(T) = q_1(T)\sigma_1^2(T) + q_2(T)\sigma_2^2(T)$$

- T – prag , $\sigma_w^2(T)$ - veličina koju pokušavamo da minimizujemo
- $q_1(T)$ i $q_2(T)$ su verovatnoća svake od klase:

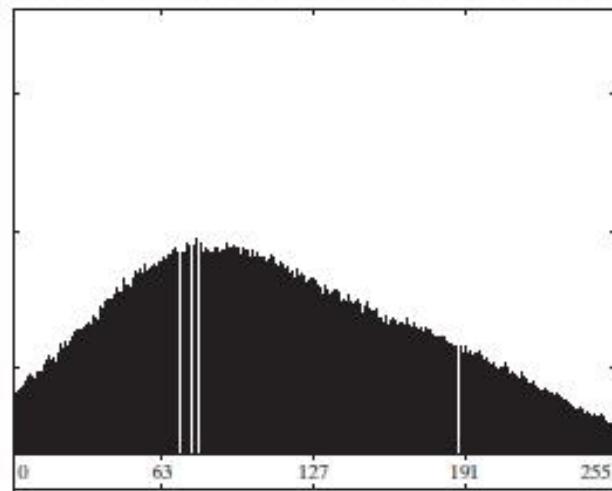
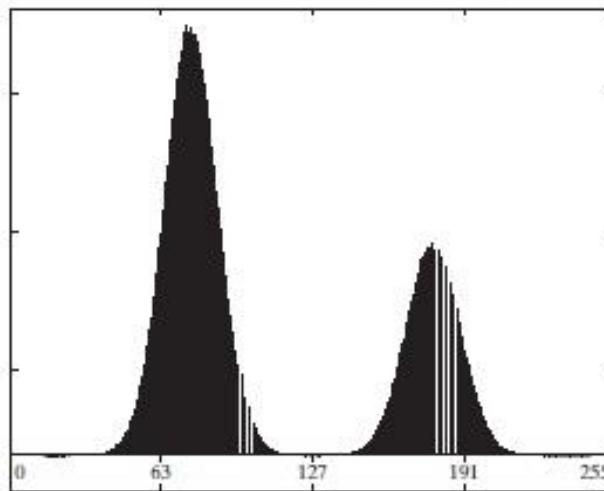
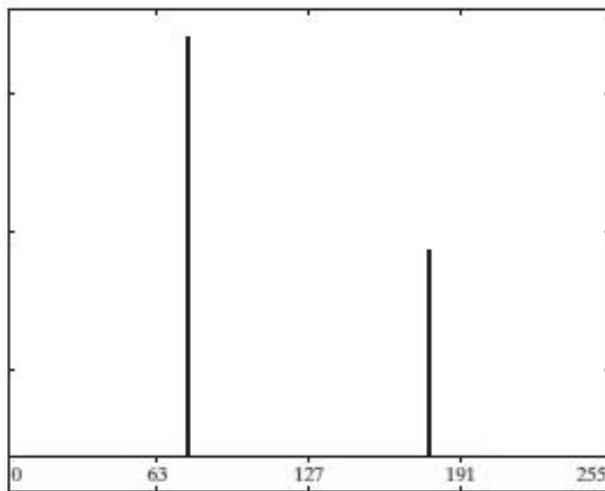
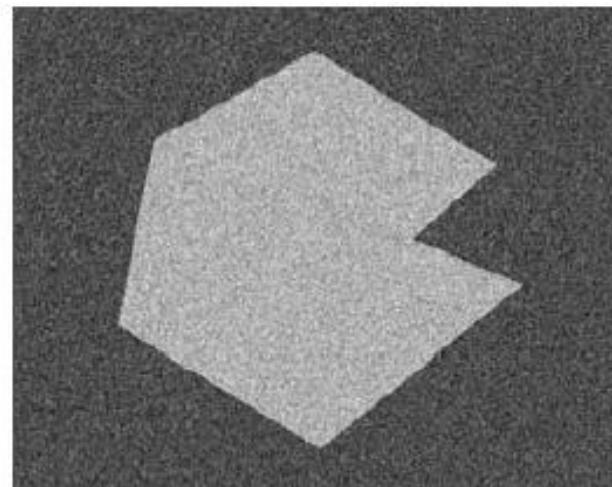
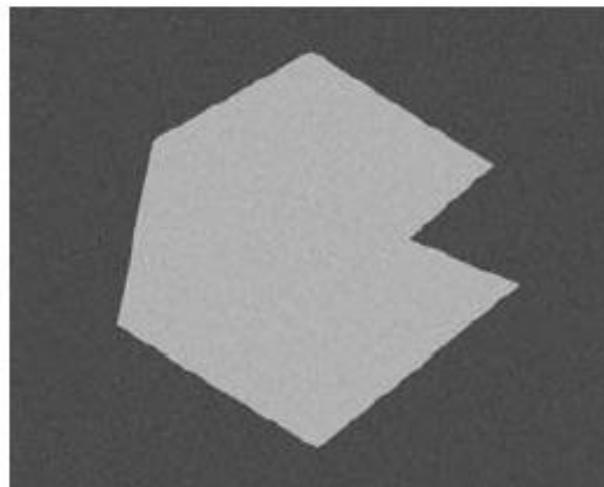
$$q_1(T) = \sum_{i=1}^T P(i) \quad q_2(T) = \sum_{i=T+1}^I P(i)$$

- P – verovatnoća piksela (koju znamo na osnovu histograma)
- σ_1^2 i σ_2^2 su varijanse svake od klase:

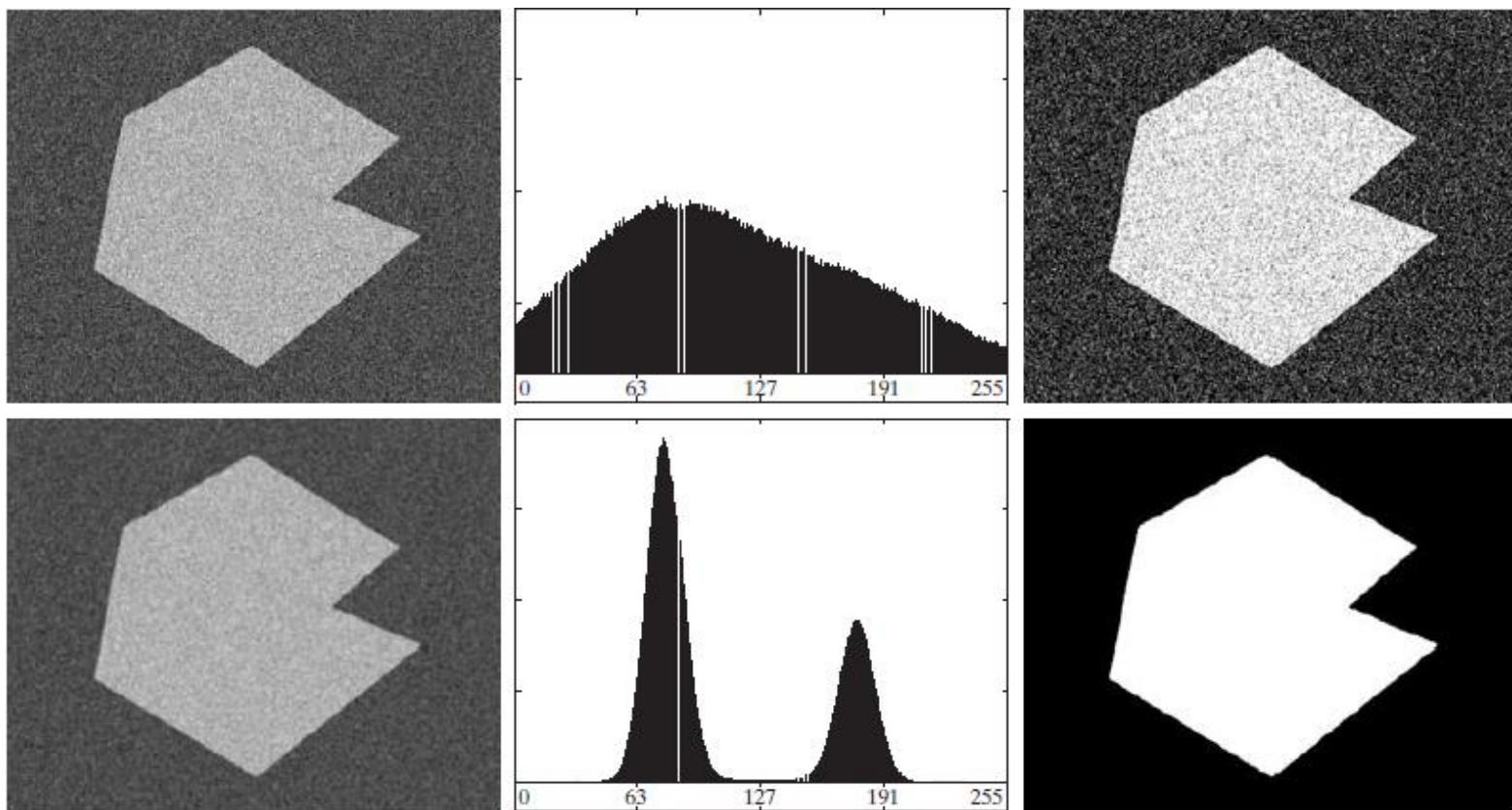
$$\mu_1(T) = \sum_{i=1}^T \frac{iP(i)}{q_1(T)}, \quad \mu_2(T) = \sum_{i=T+1}^I \frac{iP(i)}{q_2(T)}$$

$$\sigma_1^2(T) = \sum_{i=1}^T [i - \mu_1(T)]^2 \frac{P(i)}{q_1(T)}, \quad \sigma_2^2(T) = \sum_{i=T+1}^I [i - \mu_2(T)]^2 \frac{P(i)}{q_2(T)}$$

Problem šuma



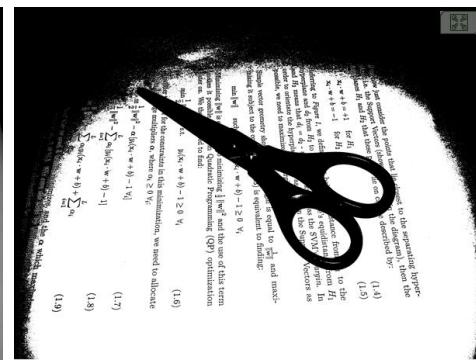
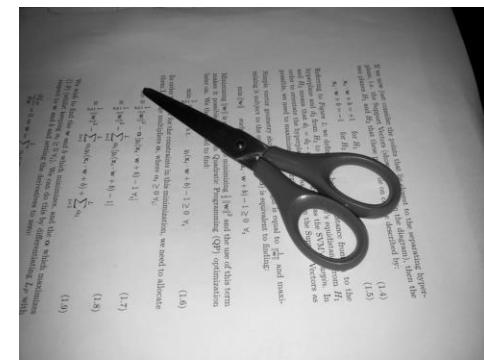
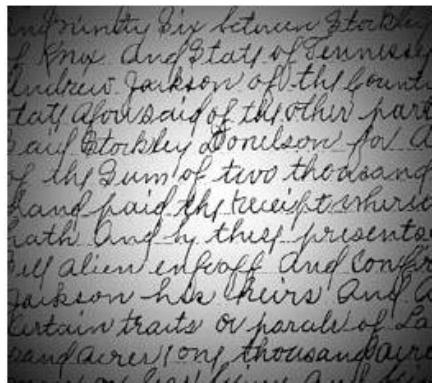
Problem šuma – rezultat primene Otsu praga



- Gore: na zašumljenim slikama, segmentacija pomoću Otsu praga nije dobra
- Dole: problem možemo ispraviti tako što pre primene Otsu algoritma uklonimo šum

Otsu nedostaci

- Otsu zanemaruje prostorne relacije
 - Koristi se samo histogram
 - Regije sa sličnim vrednostima intenziteta piksela se na histogramu nalaze blizu, bez obzira što mogu biti na sasvim različitim pozicijama slike
 - Otsu će ih tretirati kao da se radi o istoj regiji
- Nejednako osvetljenje (generalni problem kod korišćenja globalnog praga):



and Country Six between Stockley & Know And States of Tennessee Andrew Jackson off the County of Alton said of the other part was Stockley Donelson for a sum of two thousand and paid the receipt whereto hath and by these presents by alien enoff and confirm Jackson her heirs and certain traits or parcels of land acre or one thousand dollars.

$$\text{Min}_{w_1, w_2} \frac{\sum_{i=1}^m w_i}{\sum_{i=1}^m w_i + \sum_{i=m+1}^{m+n} w_i}$$

where $w_i = 0 \text{ or } 1$ (1.6)

In order to minimize this minimization, we need to introduce quadratic programming [1] and the use of thin arms

$$w_1^2 - 2w_1w_2 + w_2^2 - 1 \geq 0 \quad (1.7)$$
$$w_1^2 - 2w_1w_2 + w_2^2 - 1 \geq 0 \quad (1.8)$$

where $w_1, w_2 \in \{0, 1\}$ (1.9)

which means that the next back propagation, the w which maximizes L_w will be

$$\frac{\partial L_w}{\partial w} = w - \frac{1}{m+n}$$

is equal to $\frac{1}{m+n}$ and max w is equal to $\frac{1}{m+n}$.
In order to solve this problem, we need to use quadratic programming [1] and the use of this arm
we can use it to find w which maximizes L_w . In
the SVD we can use it to find w which maximizes
 $w^T w - b^T w - 1 \geq 0 \quad (1.6)$

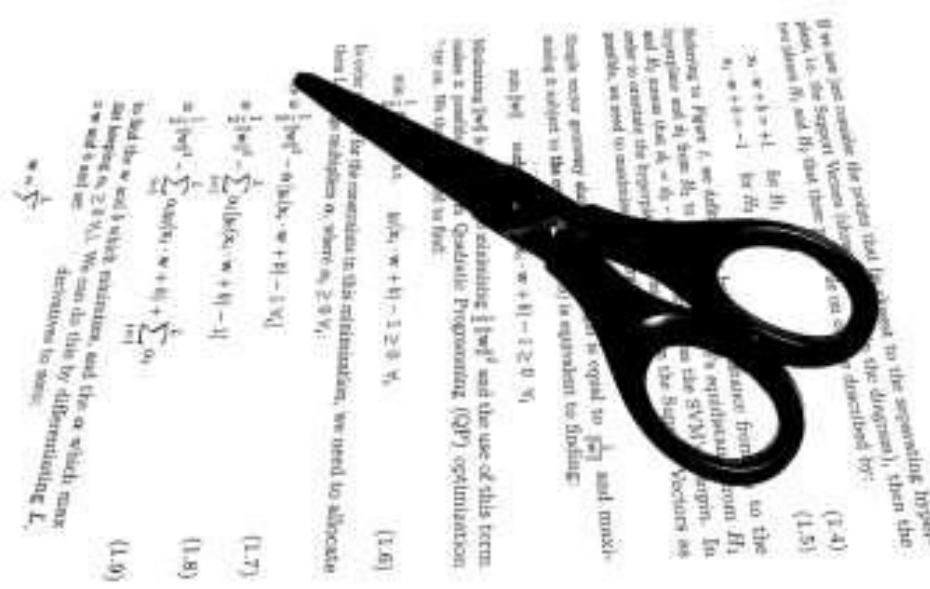
$$w^T w - b^T w - 1 \geq 0 \quad (1.7)$$
$$w^T w - b^T w - 1 \geq 0 \quad (1.8)$$
$$\sum_{i=1}^m w_i = 1 \quad (1.9)$$

Segmentacija pomoću lokalnog praga

- Rešenje nejednakog osvetljenja: tražimo lokalni prag
 - Izaberemo prag pojedinačno za svaki piksel na osnovu lokalnih vrednosti
 - Za svaki komšiluk od 50×50 piksela, izabraćemo prag koji pokriva najtamnije piksele tog komšiluka

- Ovo je proizvelo kvalitetnu segmentaciju ne samo makaza nego i teksta

- Problem: nismo uspeli da uklonimo tekst sa slike



Segmentacija pomoću lokalnog praga

In County Six between Stockley
of Knyg And State of Tennessee
Andrew Jackson off the County
tally Aforesaid of the other part
and Stockley Donelson for a
sum of two thousand
and paid the receipt wheret
neath being by these presents
all alien enforff And Confir
Jackson his heirs And a
certain traits or parale of La
sand aeever long thousand payre
and all li... and his



In County Six between Stockley
of Knyg And State of Tennessee
Andrew Jackson off the County
tally Aforesaid of the other part
and Stockley Donelson for a
sum of two thousand
and paid the receipt wheret
neath being by these presents
all alien enforff And Confir
Jackson his heirs And a
certain traits or parale of La
sand aeever long thousand payre
and all li... and his

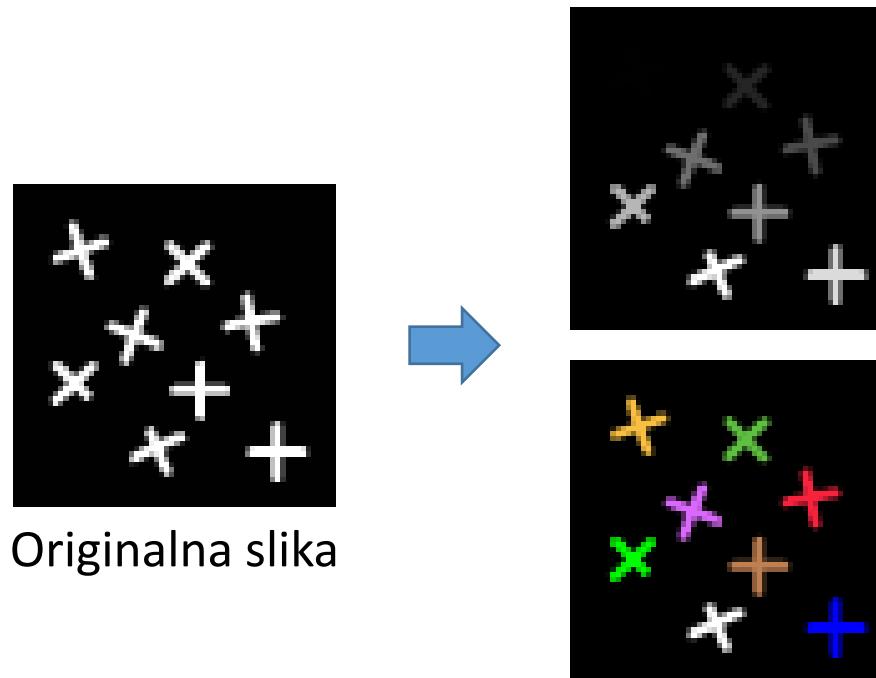
Globalni prag

In County Six between Stockley
of Knyg And State of Tennessee
Andrew Jackson off the County
tally Aforesaid of the other part
and Stockley Donelson for a
sum of two thousand
and paid the receipt wheret
neath being by these presents
all alien enforff And Confir
Jackson his heirs And a
certain traits or parale of La
sand aeever long thousand payre
and all li... and his

Lokalni prag

Označavanje povezanih regiona

- Želimo da grupišemo piksele na slici u povezane regije (komponente)
- Kada smo otkrili sve grupe piksela koje postoje, svaki piksel ćemo anotirati *grayscale* nivoom (ili bojom) zavisno od komponente kojoj pripada

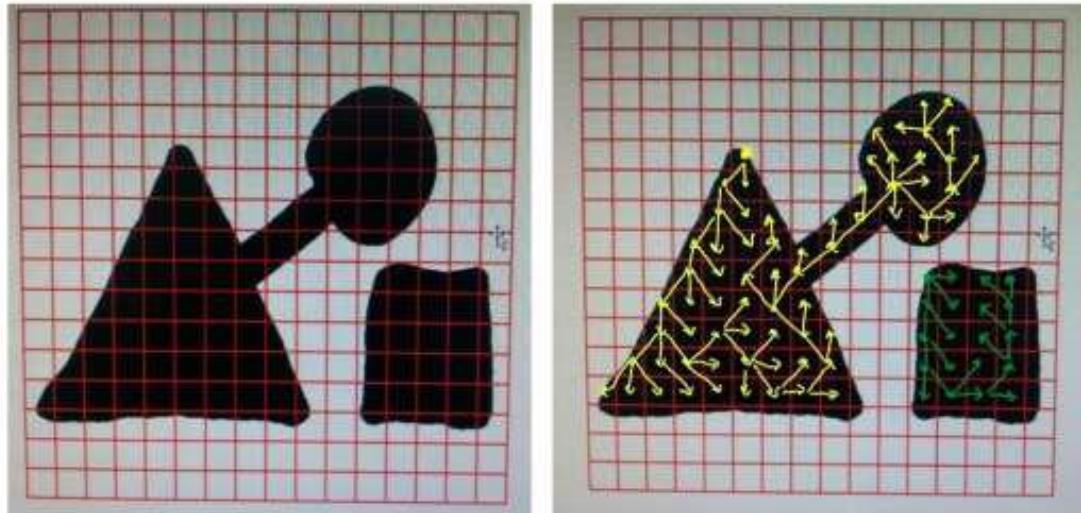


Algoritam

- U ovde izloženom algoritmu podrazumevamo da je slika binarna i podrazumevamo *8-connectivity*
- Skreniramo sliku piksel-po-piksel sve dok ne dođemo do piksela sa vrednošću 1
- Kada pronađemo piksel, razmotrimo 4 suseda ovog piksela koje smo već razmotrili (levo, iznad i dva gornja diagonalna)
 - Ako sva 4 piksela imaju vrednost 0, dodeli novu kategoriju pikselu p
 - Ako jedan od suseda ima vrednost 1, dodeli njegovu kategoriju pikselu p
 - Ako više od jednog suseda p ima vrednost 1, dodeli jednu od tih kategorija pikselu p i zabeleži ekvivalenciju kategorija
- Po završetku skeniranja piksela, dodeli jedinstvenu kategoriju svim ekvivalentnim kategorijama

Varijanta algoritma

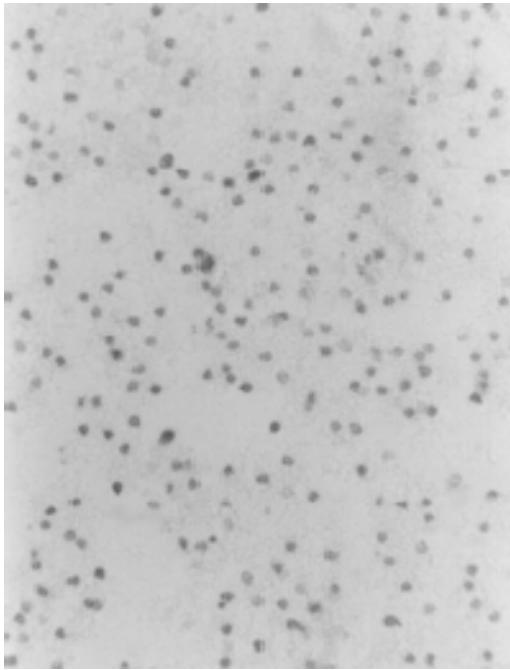
- Krenemo od prvog obojenog piksela na koga naiđemo
- Zatim pretražujemo okolne piksele koji su iste boje
 - Svaki piksel koji posetimo dodamo u **listu posećenih piksela** kako ga ne bismo obilazili dva puta
 - Sve njemu susedne piksele iste boje stavljamo u **listu otvorenih ivica**
 - U svakoj iteraciji algoritma isprobaćemo po jedan piksel iz **liste otvorenih ivica**
- Možemo vršiti *breadth first search* ili *depth first search* pretragu



Primena pronalaženja povezanih regiona

- Ekstrakcija i obeležavanje različitih povezanih komponenti na slici se nalazi u centru mnogih metoda za automatsku analizu slike
- Na primer, ovim postupkom možemo:
 - Prebrojati objekte na realnim slikama
 - Odrediti gde se koji objekat nalazi
 - Odrediti koji pikseli pripadaju objektu
 - ...

Prebrojavanje objekata



Originalna slika

Ćelije nervnog tkiva miševa: zdrave ćelije su srednjeg intenziteta sive boje, dok su mrtve ćelije guste i crne. Cilj je da se prebroje mrtve ćelije



Binarna slika

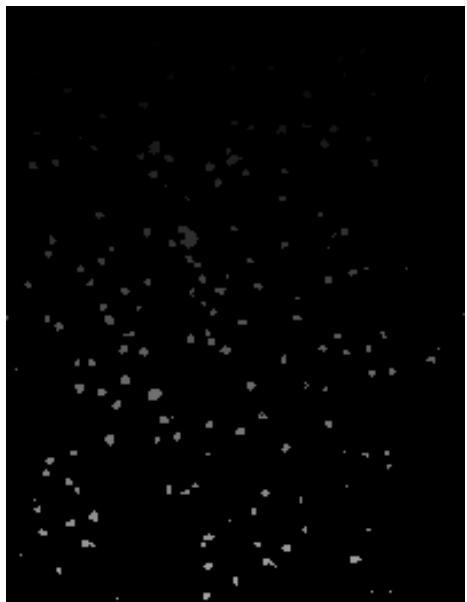
Primjenjen je prag oko 150. Bele tačke odgovaraju mrtvim ćelijama sa originalne slike



Rezultati (obojeno nijansama sive)

Algoritam je pronašao 163 komponente
Ovo je blisko originalnoj slici, ali ponegde je više različitih ćelija spojeno u istu komponentu

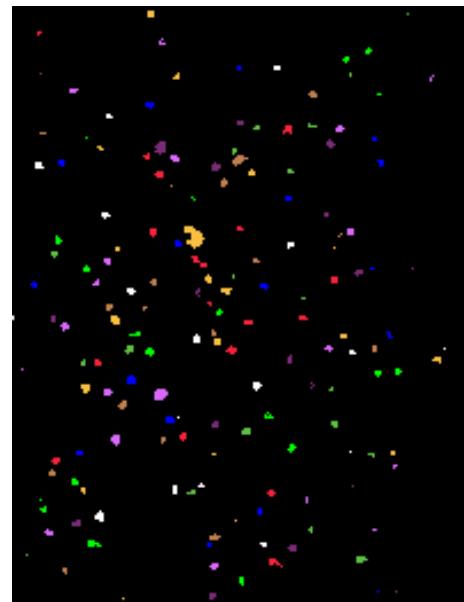
Prezentacija rezultata



Rezultati (nijanse sive)

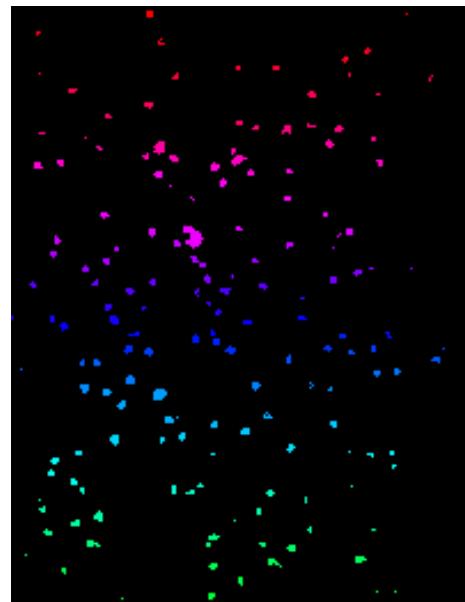
Loša vidljivost...

Možemo predstaviti bojama,
ali teško je pronaći 163 lako
raspoznatljive boje



8 boja koje je lako
razlikovati

Ali gubimo informacije jer
smo 163 kategorije mapirali
na 8



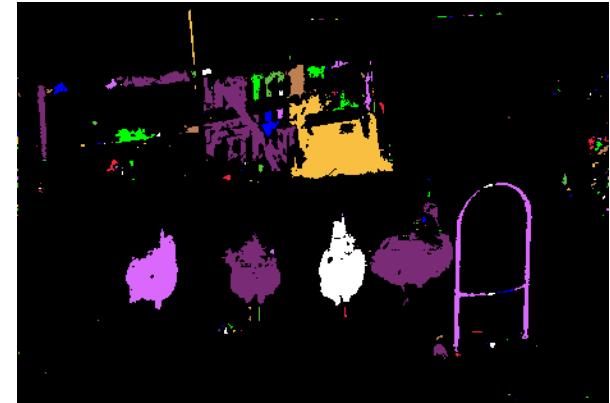
Svakom intenzitetu sive
je dodeljena druga boja

Ne gubimo informacije, ali su
mnoge nijanse slične pa je teško
videti granice bliskih komponenti

- Sofisticiranije tehnike kombinuju ove dve tehnike – osiguraće se da bliske komponente uvek imaju boje koje se lako razlikuju

Prebrojavanje objekata

- Veći problem imamo kod prebrojavanja čurki na sledećoj slici:

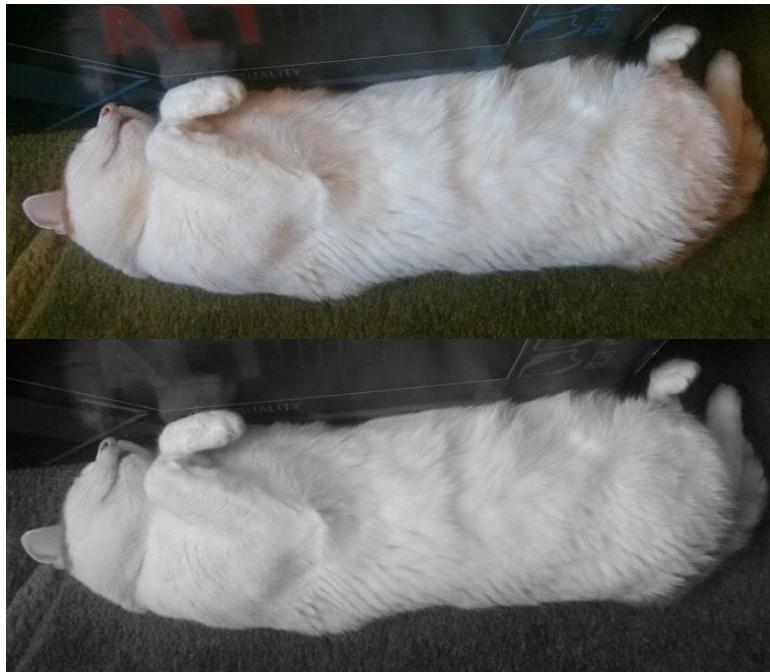


- Iako je svaka čurka predstavljena posebnom komponentom, pronašli smo ukupno 196 komponenti od kojih nisu čurke
- Ova dva primera su pokazala da je označavanje povezanih komponenti lak deo procesa automatske analize: glavni zadatak je da se dobije dobra binarna slika koja razdvaja objekte od interesa od pozadine

Morfološka obrada slike

Morfološka obrada slike – motivacija

- Segmentacija pomoću praga
 - Dobra za pronalaženje objekata koji su svetlijii ili tamniji od pozadine
 - Izlaz su binarne slike (crni pikseli – pozadina, beli – obejkat od interesa)
- Ali, gotovo uvek će rezultujuće binarne slike imati neke probleme u vidu crnog ili belog šuma

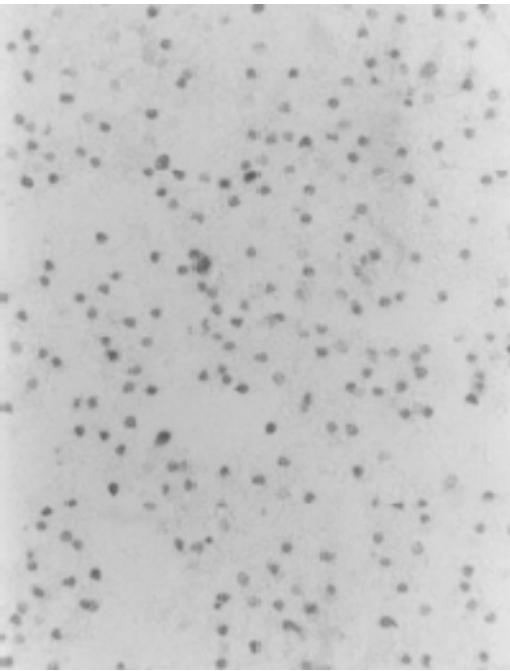


Morfološka obrada slike – motivacija



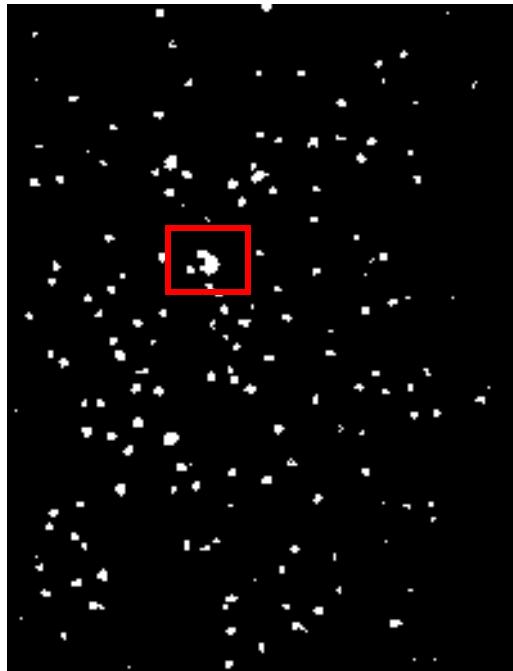
- Slova lošeg kvaliteta – njihova obrada pre prosleđivanja neuronskoj mreži bi možda pobjošala njene preformanse

Morfološka obrada slike – motivacija



Originalna slika

Ćelije nervnog tkiva miševa: zdrave ćelije su srednjeg intenziteta sive boje, dok su mrtve ćelije guste i crne. Cilj je da se prebroje mrtve ćelije



Binarna slika

Primjenjen je prag oko 150. Bele tačke odgovaraju mrtvim ćelijama sa originalne slike

- Više objekata spojenih u jedan zbog belog šuma
- Može da predstavlja problem za prebrojavanje objekata

Morfološka obrada slike

- Ulaz:
 - Binarna slika
 - Strukturni element
 - Izlaz:
 - Binarna slika iste veličine kao ulazna
 - Strukturni element
 - Predstavlja matricu sastavljenu od 0 i 1
 - Definiše okolinu piksela

Box

1	1	1
1	1	1
1	1	1

3x3

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

5x5

Disc

0	1	0
1	1	1
0	1	0

0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0

Morfološka obrada slike

- Morfološke operacije
 - Skup operacija za obradu slika koje se baziraju na oblicima
 - Vrednost svakog piksela izlaza će biti bazirana na poređenju odgovarajućeg piksela ulazne slike sa svojom okolinom
 - Odabirom veličine i oblika susedstva, možemo kreirati morforloške operacije osetljive na određene oblike na ulaznoj slici
- Osnovne binarne operacije su:
 - Dilacija – dodaje piksele na ivicu objekata na slici
 - Erozija – uklanja piksele sa ivica objekata na slici
- Njihovom kombinacijom možemo konstruisati operatore za:
 - Otvaranje
 - Zatvaranje
 - Detekciju ivica

Dilacija (Dilation)

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

- Logička operacija OR između elemenata neke okoline
 - Dovoljno je da BAR jedna tačka iz okolike bude 1 pa da tačka u sredini bude 1
- Količina piksela sa vrednošću 1 se povećava i rupe objektu (manje od strukturnog elementa) se popunjavaju (uklanja *false negatives*)

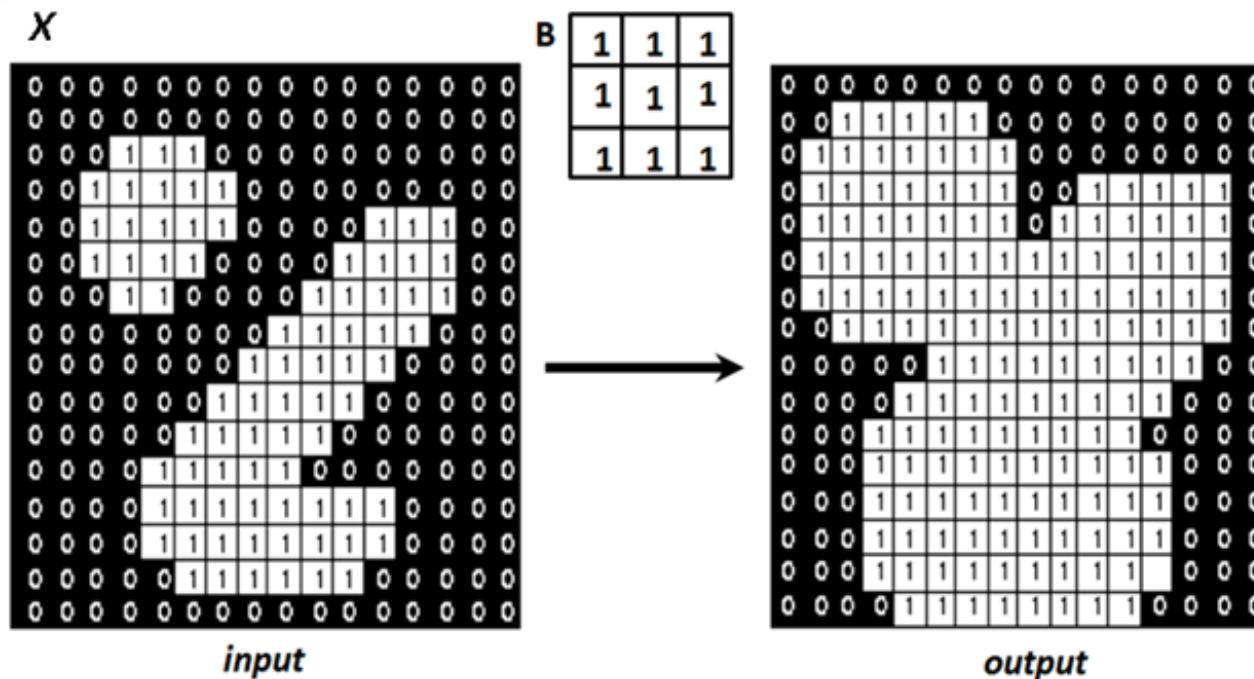


Figure 1. Effect of dilation using a 3X3 square structural element B.

Dilacija - primer

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



0	1	0
1	1	1
0	1	0

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



- Ulagna slika:

- Loš kvalitet slova
- Znamo da je maksimalna veličina prekida 1-2 piksela – ne treba nam veliki strukturni element

- Izlazna slika:

- Rupe su popunjene
- Možemo primeniti algoritam za pronalaženje povezanih komponenti za detekciju slova

Erozija (*Erosion*)

$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\}$$

- Logička operacija **AND** između elemenata neke okoline
 - Piksel izlazne slike će biti 1 ukoliko je svaki piksel u okolini tog piksela na ulaznoj slici ima vrednost 1. U suprotnom, biće 0
 - Količina piksela sa vrednošću 1 se smanjuje i veličina objekta se smanjuje (uklanja *false positives*)

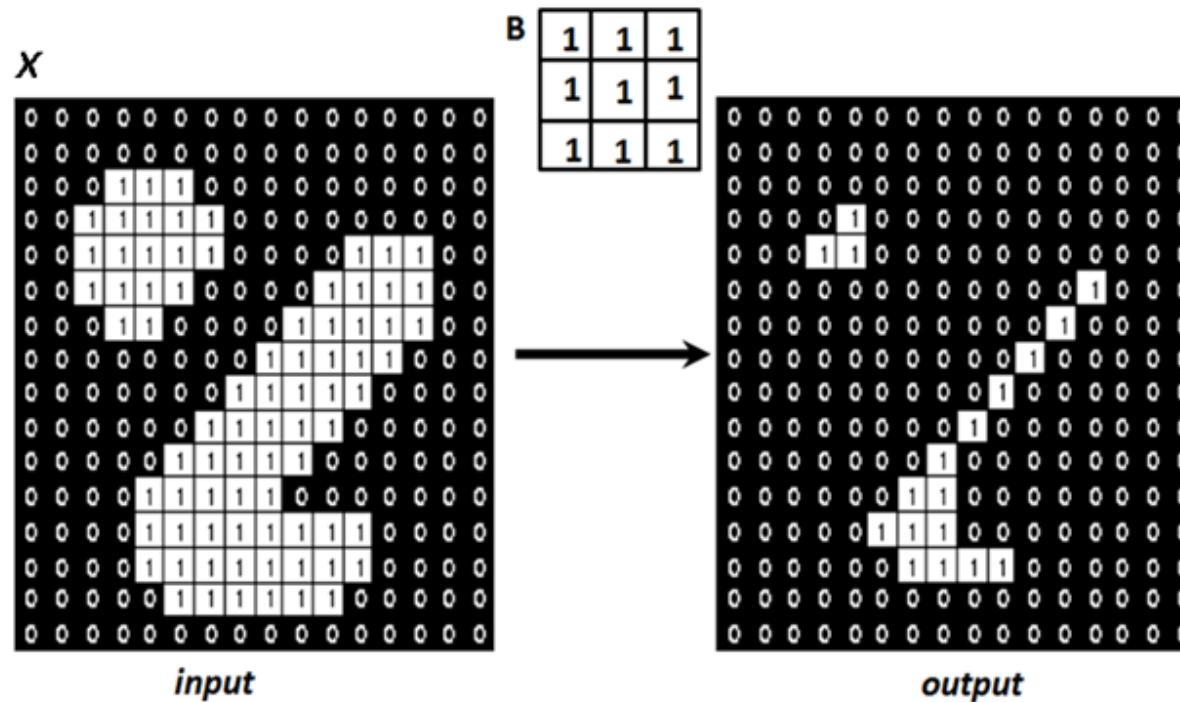
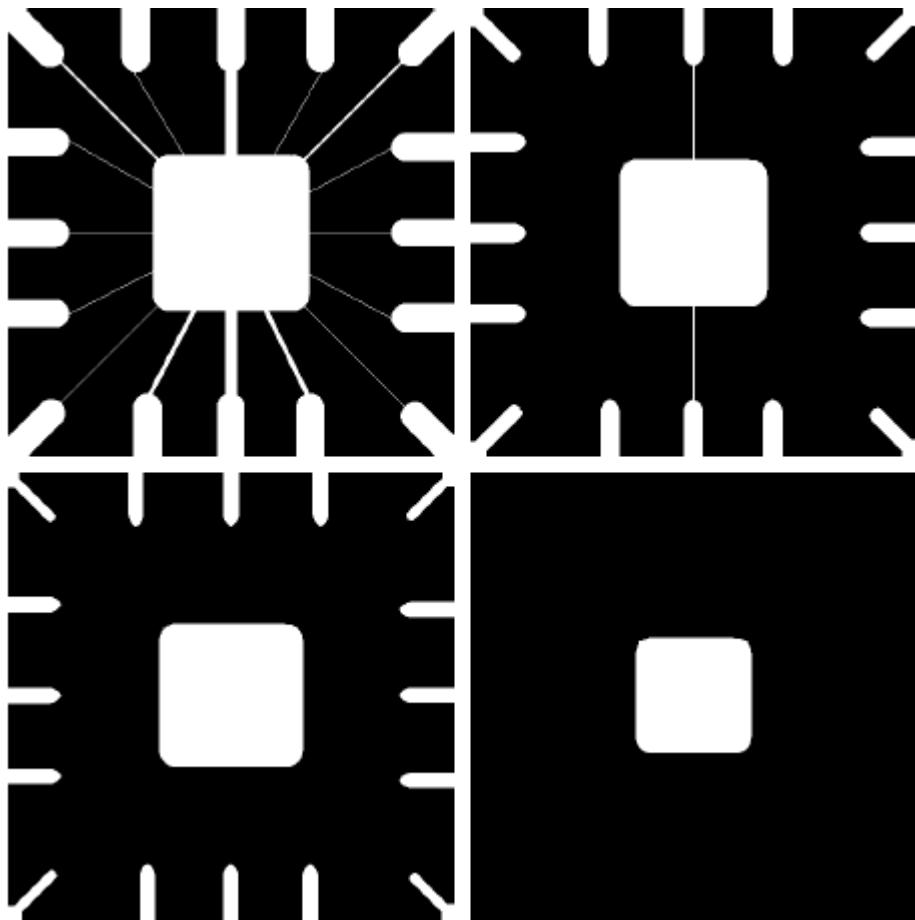


Figure 2. Effect of erosion using a 3X3 square structural element B.

Erozija – primer



- Recimo da želimo da uklonimo tanke linije sa slike
 - Wire-bond mask 486×486
- Erozija primenom strukturnog elementa sa svim 1 veličine:
 - 11×11 (vertikalne linije nisu uklonjene jer su šire od 11 piksela)
 - 15×15
 - 45×45 (dalje uvećavanje strukturnog elementa uklanja veće komponente)

Šta bismo želeli?

- Šta možemo:
 - Dilacija – možemo da premostimo neželjen procepe ali uvećavamo objekte
 - Erozija – uklanjamo neželjene tanke linije ali smanjujemo objekte
- Ali, želeli bismo približno iste veličine objekata na slici uz uklanjanje sitnih problema
 - Otvaranje – prvo erozija pa onda dilacija
 - Zatvaranje – prvo dilacija pa onda erozija

Otvoranje

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

- Koristimo *isti strukturni element* da izvršimo prvo eroziju pa onda dilaciju
- Efekat: Izglađivanje (*smoothing*) ivica objekta: uklanjanje tankih mostova i eliminisanje tankih struktura

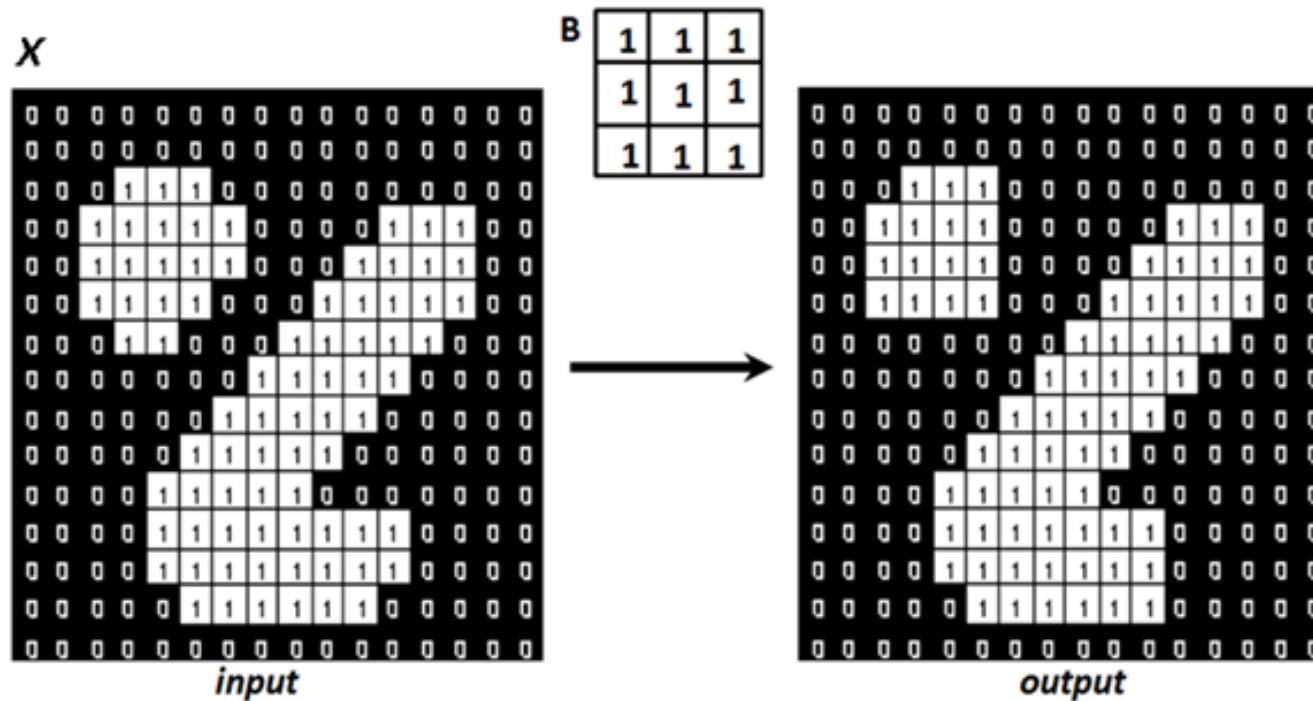
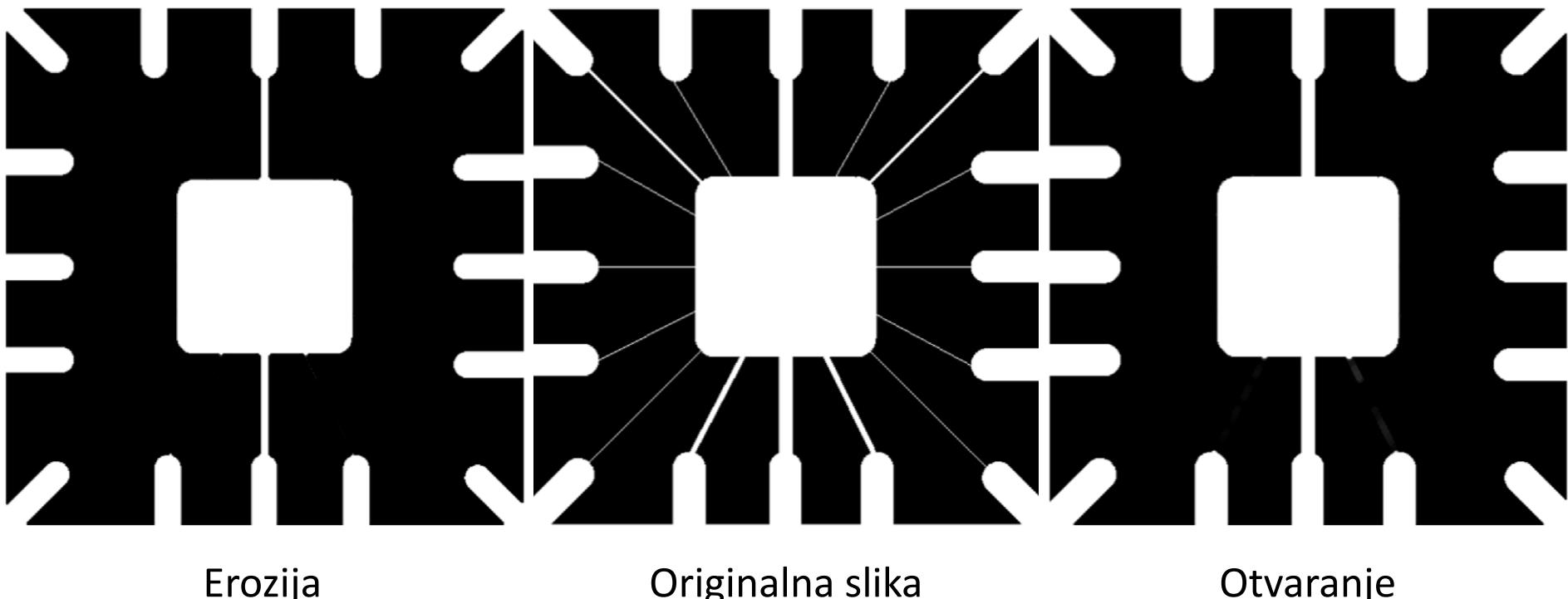


Figure 3. Effect of **opening** using a 3X3 square structural element B .

Erozija i Otvaranje



Erozija

Originalna slika

Otvarenje

Linije tanje od strukturnog elementa su uklonjene, ali su objekti na slici smanjeni

Linije tanje od strukturnog elementa su uklonjene, a veličina objekata sačuvana

Zatvaranje (*Closing*)

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

- *Koristimo isti struktturni element* da izvršimo prvo dilaciju pa onda eroziju
 - Efekti: Izglađivanje objekata dodavanjem piksela: smanjuje rupe u objektima manje od veličine struktturnog elementa, povezuje strukture odvojene malim prekidima,...

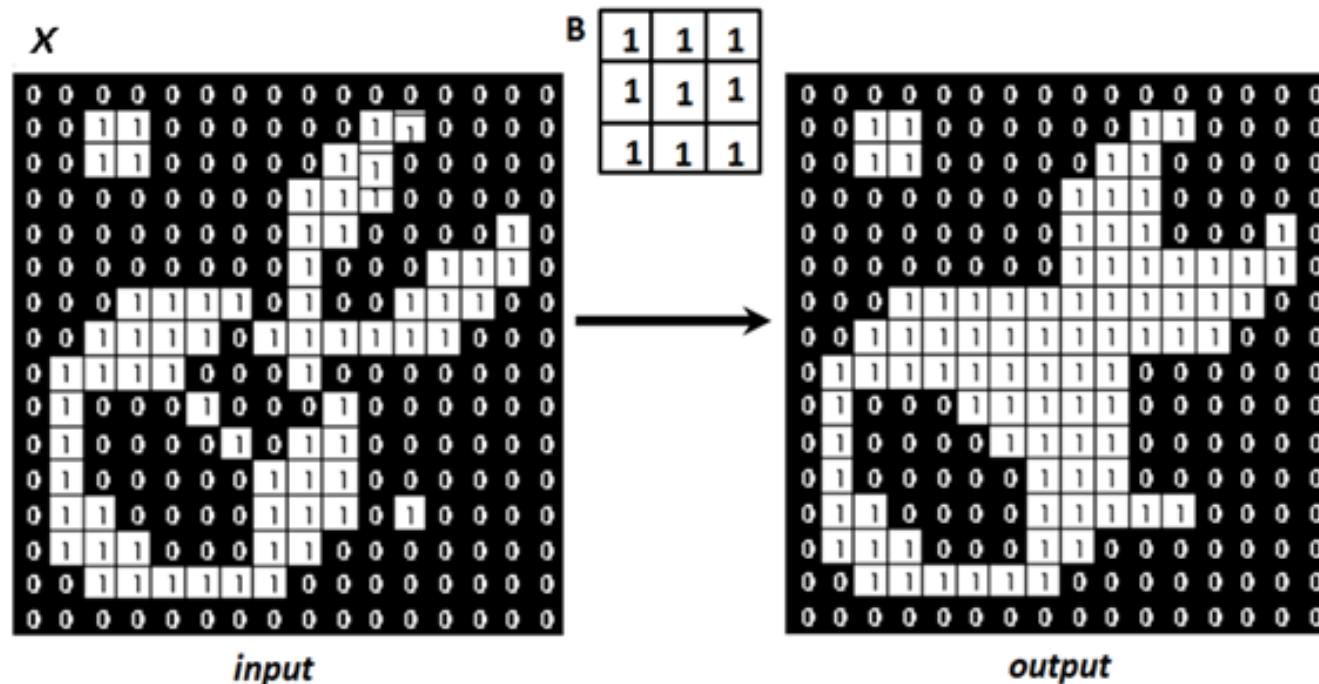


Figure 4. Effect of closing using a 3X3 square structural element B.

Dilacija i zatvaranje

Historically, certain computer programs were written using

Dilacija

Historically, certain computer programs were written using

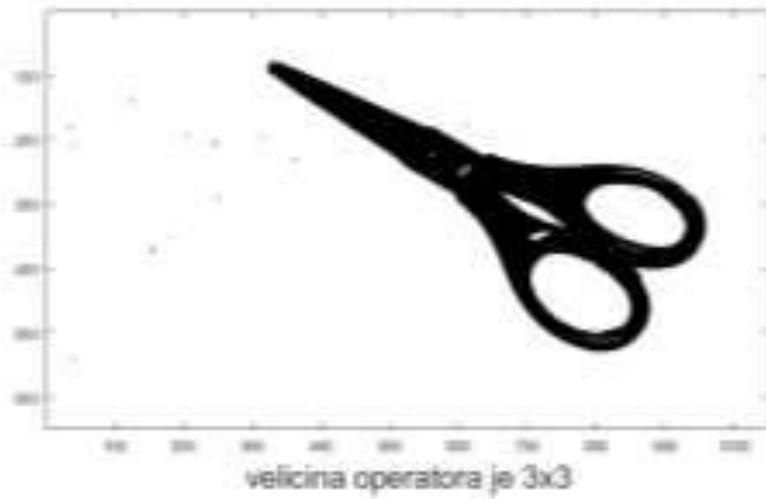
Originalna slika

Historically, certain computer programs were written using

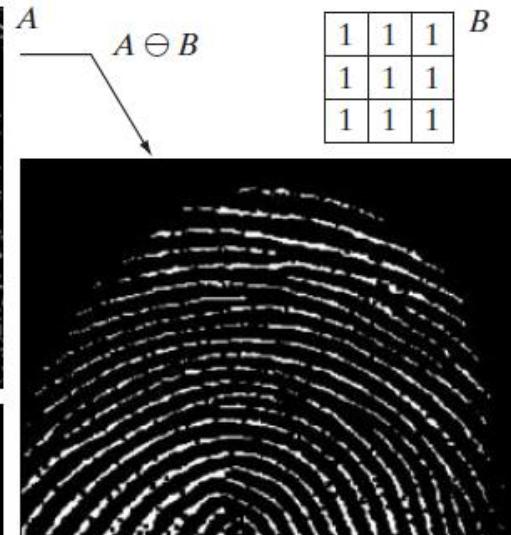
Zatvaranje

Zatvaranje

- Morfološkom operacijom zatvaranja možemo da izbacimo tekst sa slike



Primer primene



$$(A \ominus B) \oplus B = A \circ B$$
$$(A \circ B) \oplus B = [(A \circ B) \ominus B] \oplus B = (A \circ B) \bullet B$$

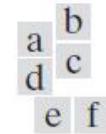


FIGURE 9.11

- (a) Noisy image.
- (b) Structuring element.
- (c) Eroded image.
- (d) Opening of A.
- (e) Dilation of the opening.
- (f) Closing of the opening.
(Original image courtesy of the National Institute of Standards and Technology.)

Uticaj struktturnog elementa

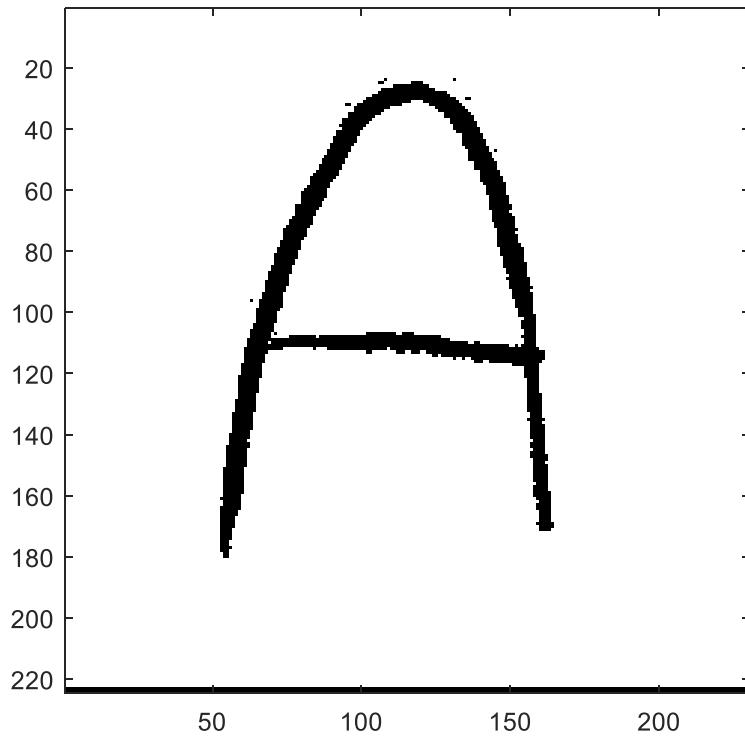
- I kod otvaranja i kod zatvaranja, jačina izgleđivanja zavisi od veličine struktturnog elementa
 - Veličinu treba odrediti u skladu sa veličinom objekata od interesa
 - Za velike objekte, veći struktturni element je preferiran, dok je za manje neophodan manji radi očuvanja finih struktturnih detalja objekta
- Poželjan oblik struktturnog elementa takođe zavisi od geometrijskih oblika objekata na slici
 - Na medicinskim slikama obično postoji veoma malo pravih linija i objekata, pa se preferira oblik diska
 - Na satelitskim snimcima imamo objekte poput zgrada, puteva,... Zato bi bilo dobro da je struktturni element u obliku pravougaonika (odgovarajuće orientacije)

Detekcija ivica

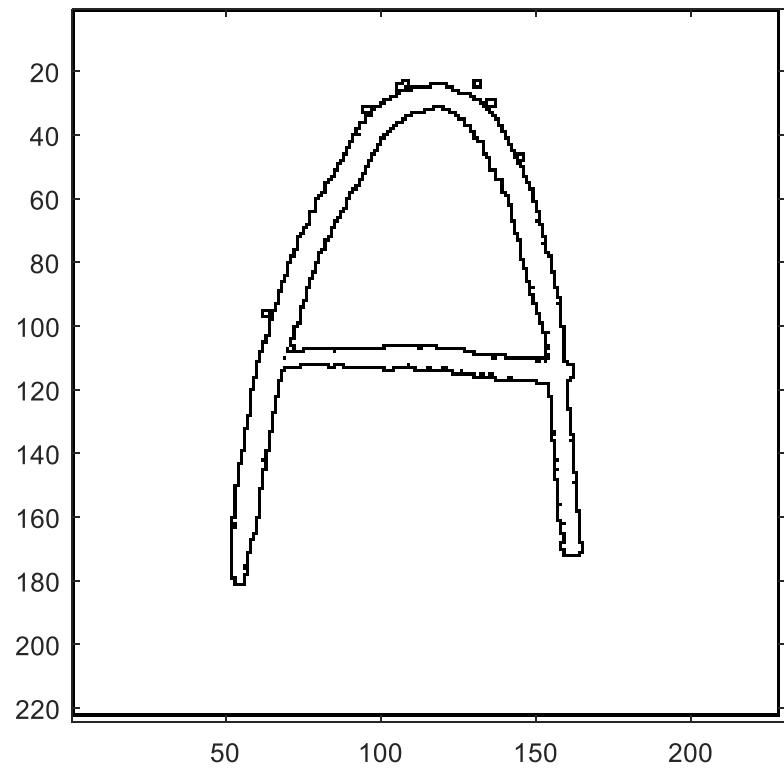
- Ima mnogo kombinacija dilacije i erozije
- Možemo ih iskoristiti da npr. pronađemo ivice objekta

Detekcija ivica

$f =$

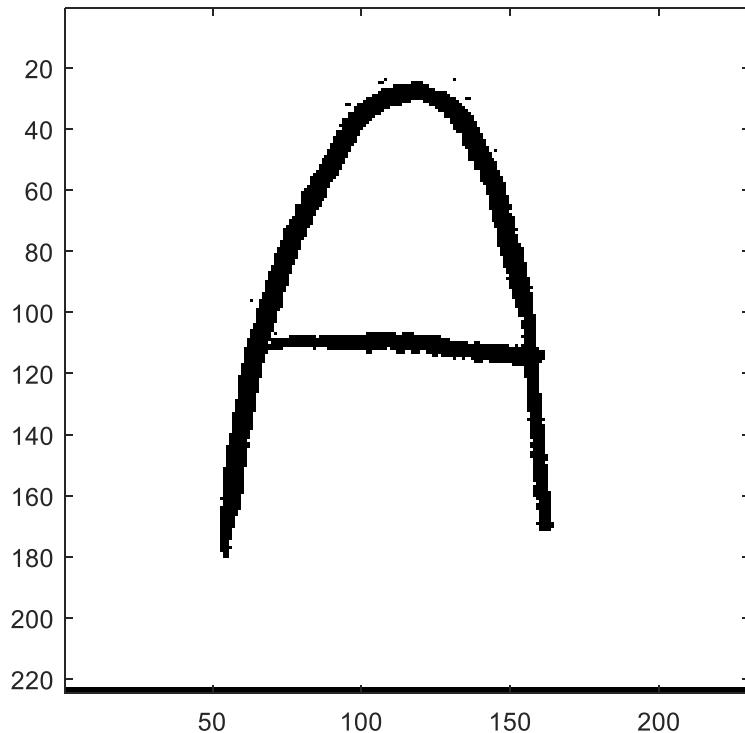


Detekcija ivica₁ = Dilacija - f

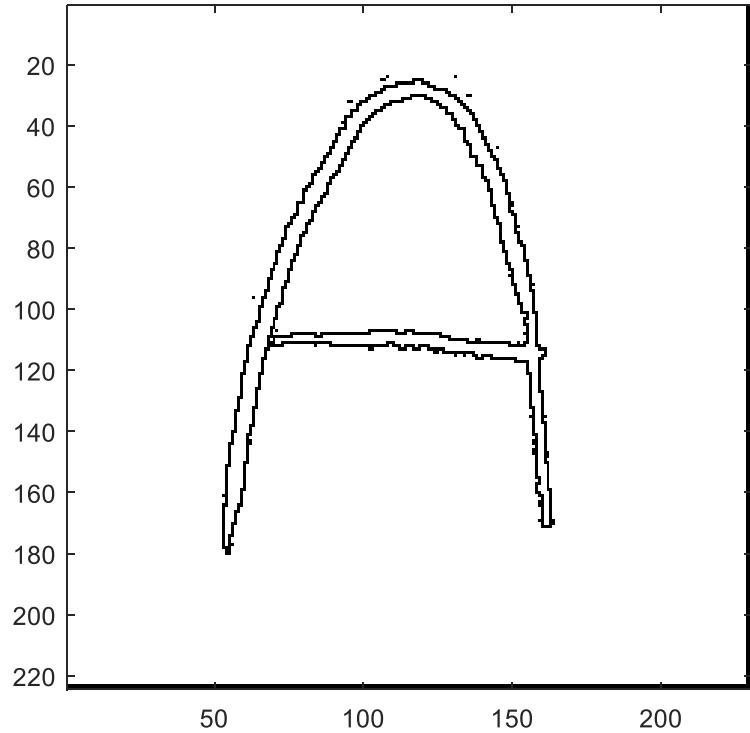


Detekcija ivica

$f =$



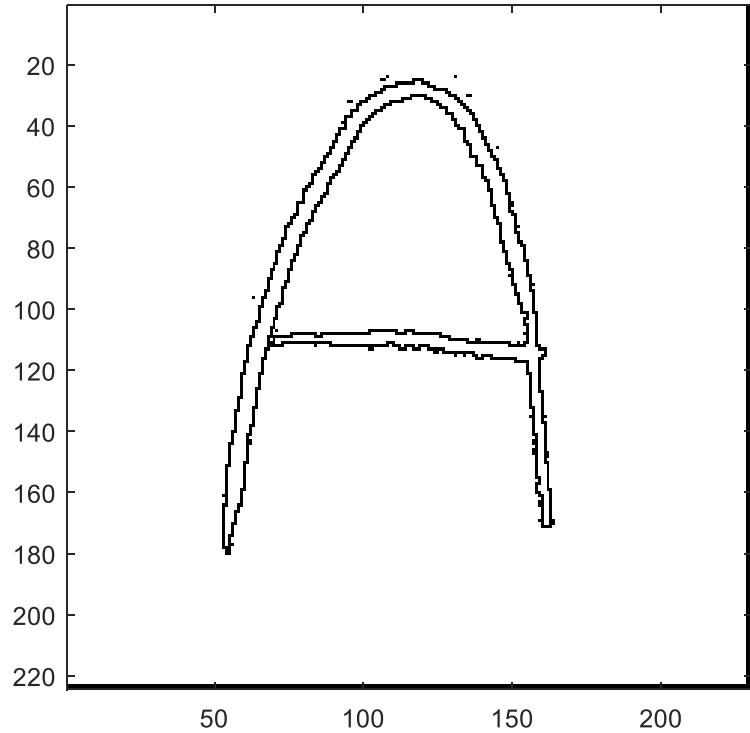
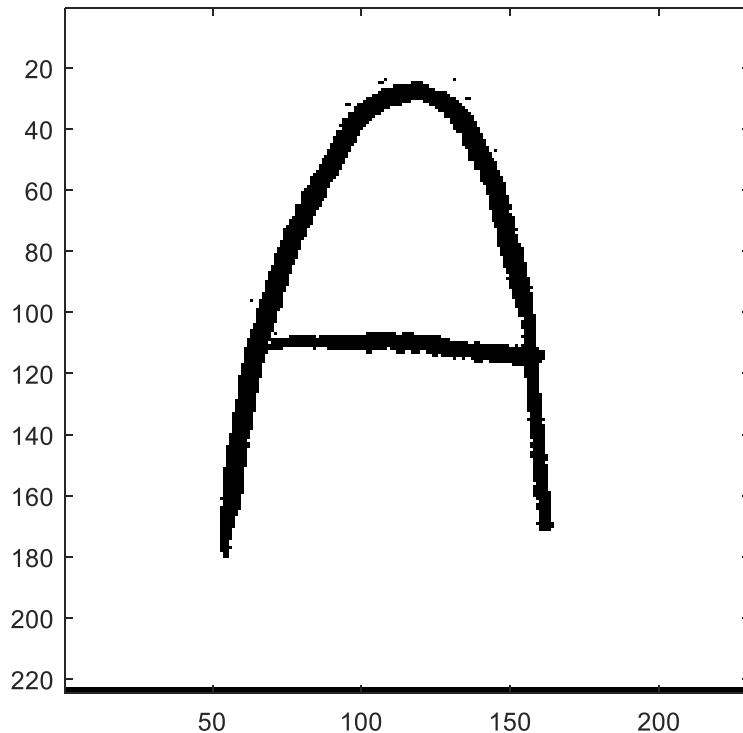
Detekcija ivica₂ = $f - \text{Erozija}$



Detekcija ivica

Detekcija ivica₃ = Dilacija – Erozija

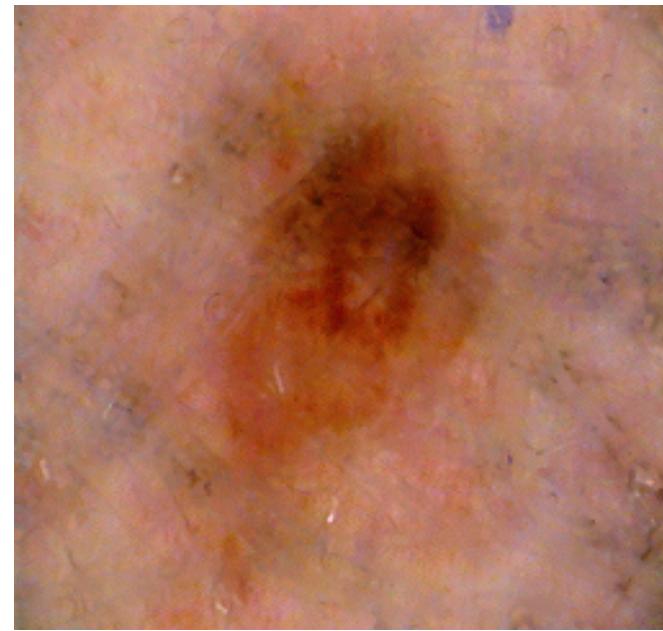
$f =$



Proširenje na *grayscale* slike

- Dilacija – maksimum piksela susedstva
- Erozija – minimum piksela susedstva

Primena – *DullRazor* tehnika



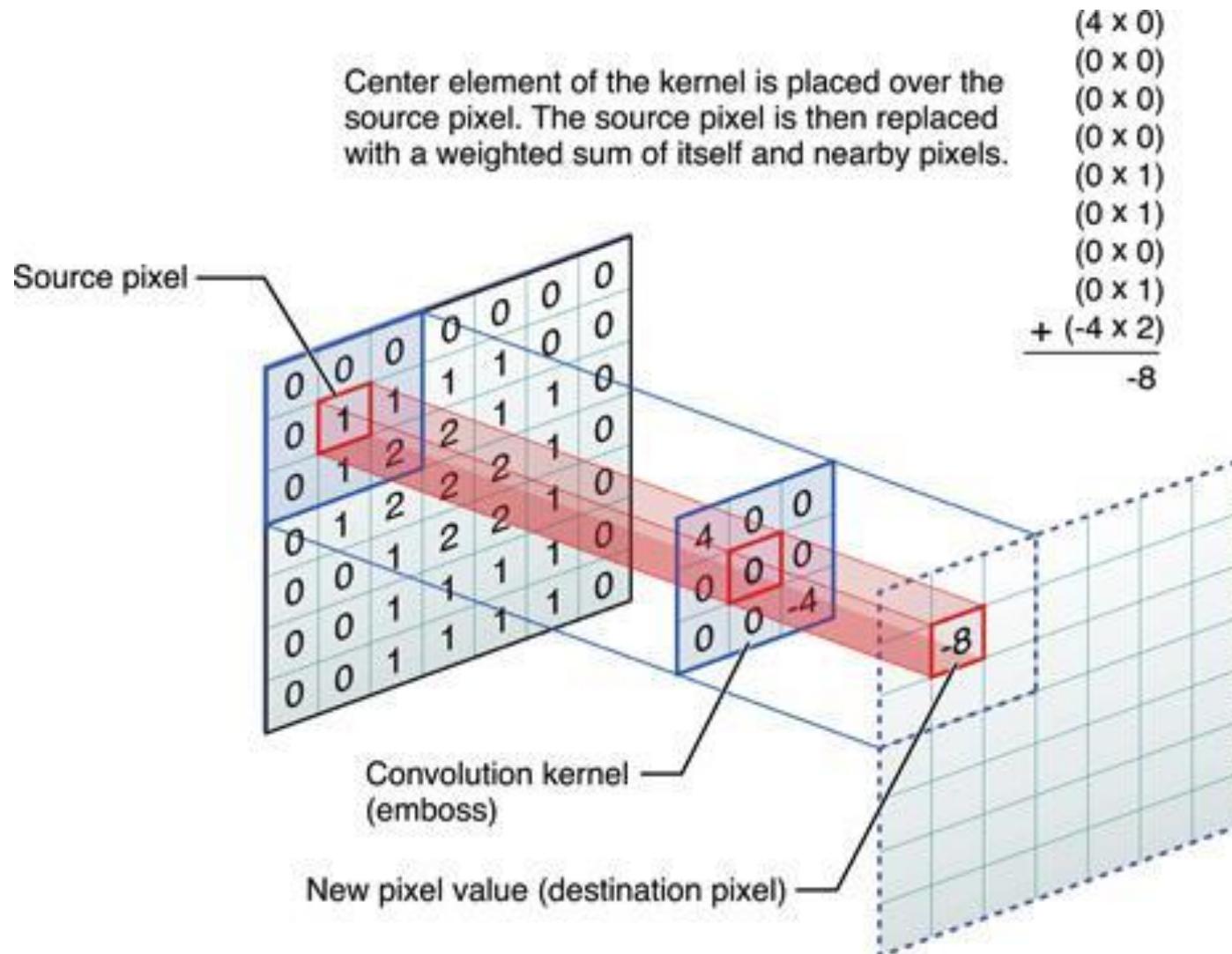
- Klasifikacija mladeža na benigne i melanome
- *DullRazor* – kombinacija dilacije, erozije, oduzimanja slika, uklanjanja šuma, primene maske,...
- Lee, Tim, et al. "Dullrazor®: A software approach to hair removal from images." *Computers in biology and medicine* 27.6 (1997): 533-543
- Kiani, Kimia, and Ahmad R. Sharafat. "E-shaver: An improved DullRazor® for digitally removing dark and light-colored hairs in dermoscopic images." *Computers in biology and medicine* 41.3 (2011): 139-145.

Konvolucija

Konvolucija

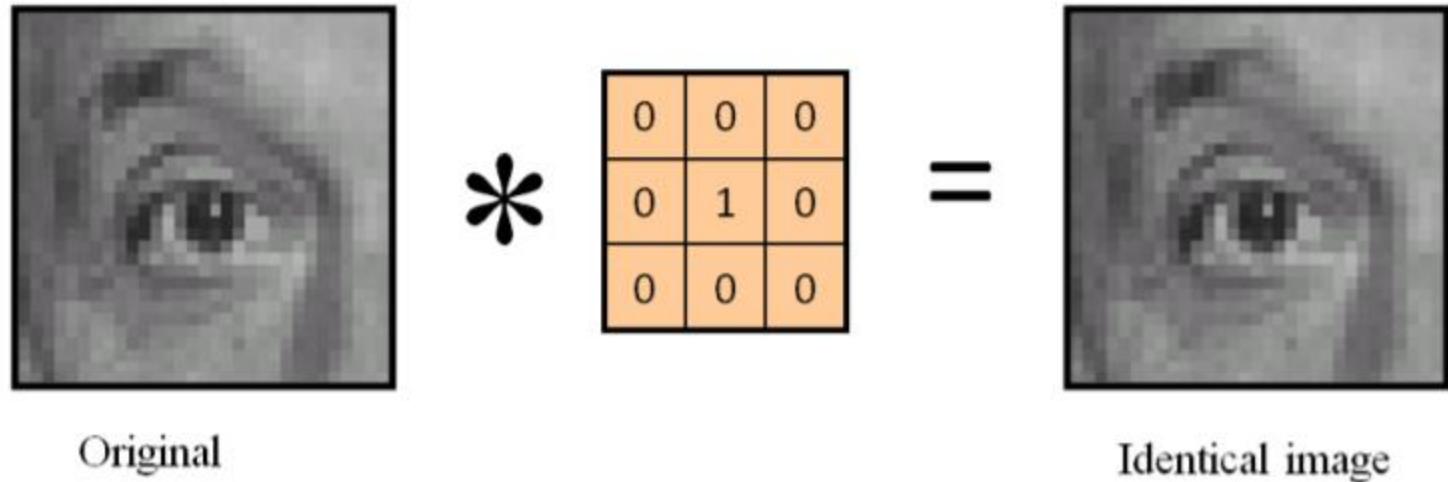
- Konvolucija je matematička operacija definisana nad dve funkcije koja proizvodi treću funkciju
- Pošto smo definisali slike kao funkcije, onda je možemo primeniti na slikama
- Konvolucija je važna operacija za *signal processing*
 - Može se primeniti za dva signala (1D) ili dve slike (2D)
 - Razmišljajte o jednom signalu kao ulaznom, a drugom (kernelu) kao o filteru koji se primenjuje da bismo dobili izlaz

Konvolucija



Konvolucija

- „Prevlačimo“ kernel preko svakog piksela na slici
- Vrednost piksela na izlaznoj slici dobijamo tako što množimo odgovarajuće vrednosti ulazne slike i kernela, a zatim sve umnoške saberemo
- Npr. sledeći kernel ništa ne menja u originalnoj slici:



- Često se ova metoda zove *weighted sliding window*
- Ulaznu sliku možemo padovati nulama kako bi izlazna bila iste veličine

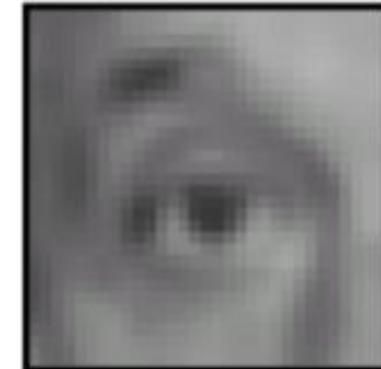
Konvolucija

- *Mean filter (low-pass filter)*



Original

$$\ast \quad \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} =$$



Blur (with a mean filter)

- *Shift left*



Original

$$\ast \quad \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} =$$



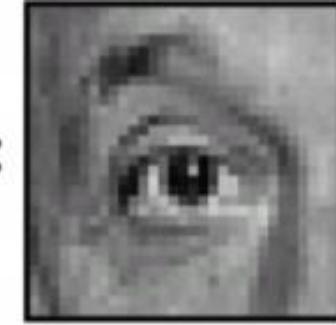
Shifted left
By 1 pixel

Konvolucija

- *Sharpening (high-pass filter)*



$$\ast \left(\begin{matrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{matrix} - \frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \right) =$$



Sharpening filter
(accentuates edges)

