Computer Vision: OpenCV

Mihailo Grbić i Slobodan Jenko

Matematička gimnazija
NEDELJA$^{\text{v5.0}}_{\text{INFORMATIKE}}$

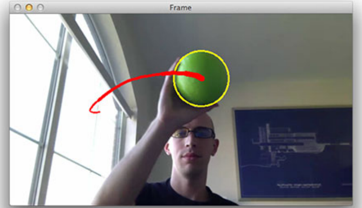
18. decembar 2018.

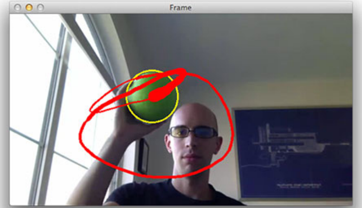
Šta je Computer Vision

- Obrada slike vs Kompjuterska vizija

Primene CV-a

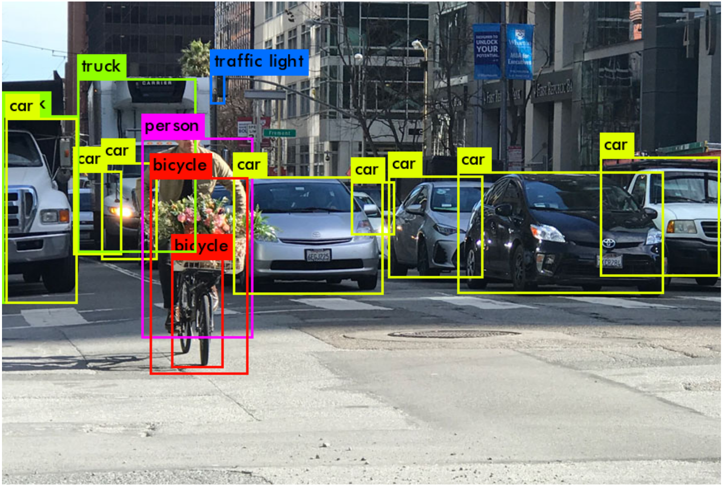
- Praćenje objekata





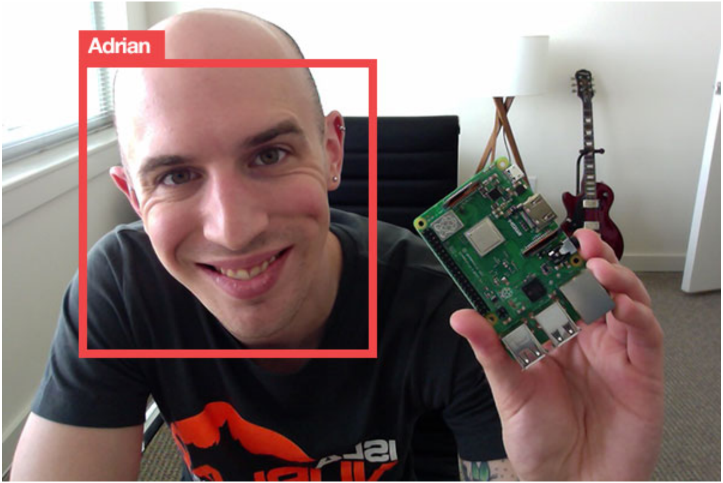
Primene CV-a

- Praćenje objekata kod samovozećih automobila



Primene CV-a

- Face Recognition

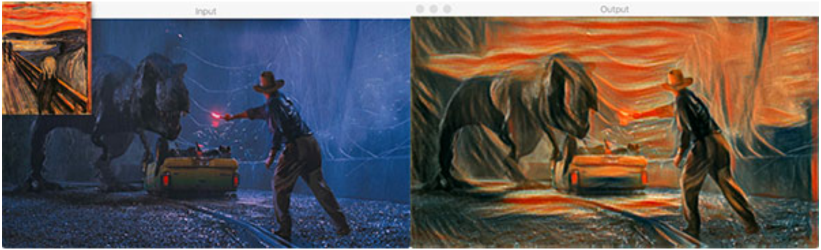


Primene CV-a

- Opisivanje slika

Magije CV-a

- Style Transfer



Magije CV-a

- Night Vision

(a) Camera output with ISO 8,000 (b) Camera output with ISO 409,600 (c) Our result from the raw data of (a)

Glavni problem CV-a

- Hajde da napravimo naš samovozeći automobil

Glavni problem CV-a

- Šta ćemo sad?

Glavni problem CV-a

- A sad?

Glavni problem CV-a

- A tek sad?

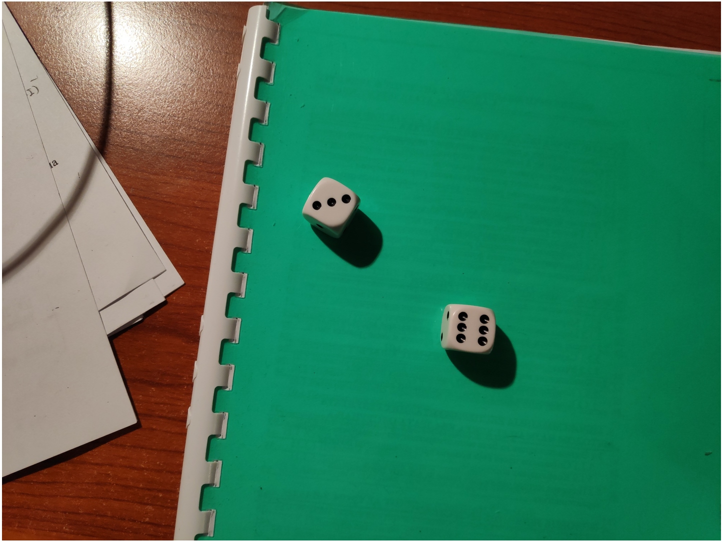
Glavni problem CV-a

- ...

Glavni problem CV-a

- Glavni problem CV-a = Specifičnost algoritama, a hiljade slučajeva
- Rešenje = Sveoubuhvatni i invarijantni algoritmi

Broj na kockicama



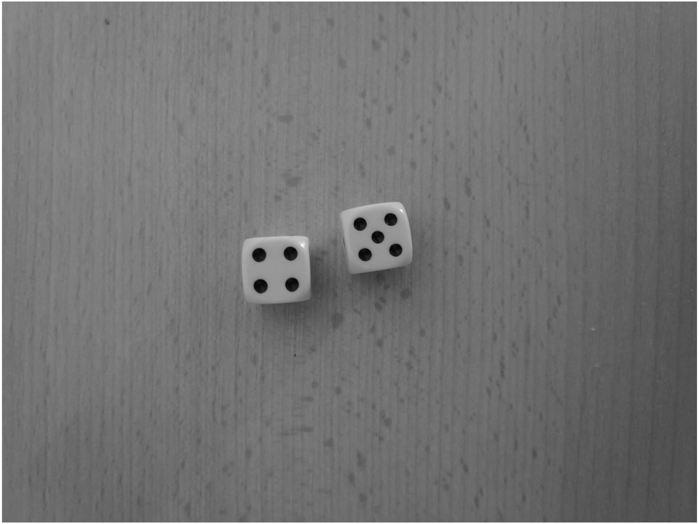
Kako će funkcionisati radionica?

- 5 Challenge-a
- Rastuća težina - sve više specifičnih slučajeva
- Menjate funkciju $my\_function$ u Python skripti $Dice\_Solver.py$
- Vaš kod testirate skriptom $Tester.py$ i unošenjem odgovarajućeg broja challenge-a

Primer slike

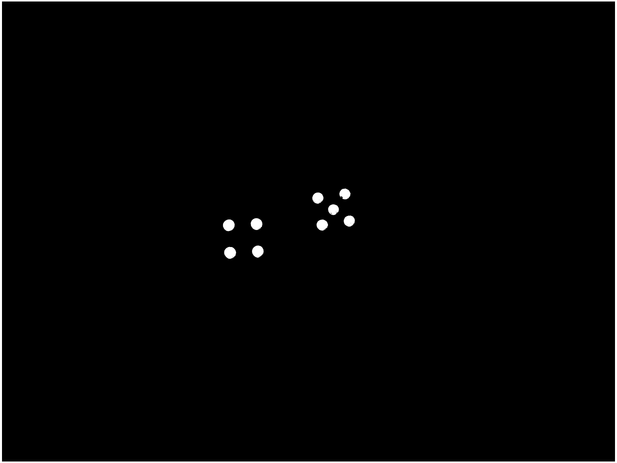
Images/Challenge1.jpg

Grayscale

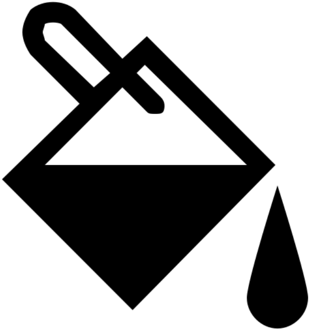


Binarizacija

- if Illumination<x then white; else black

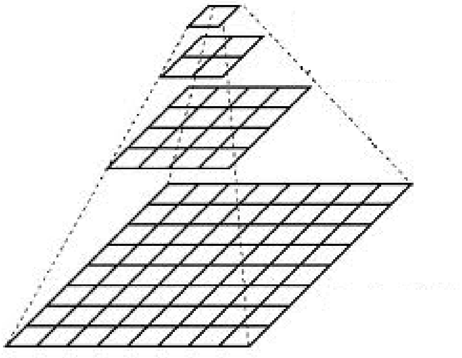


Flood Fill



Flood Fill

Pyramid Down



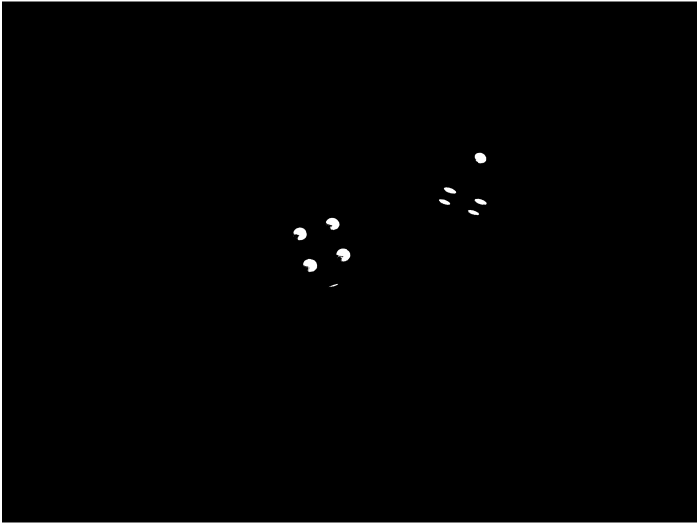
Rešenje

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
ret,thresh = cv2.threshold(gray, 35, 255, cv2.THRESH_BINARY_INV)

thresh=cv2.pyrDown(thresh)
thresh=cv2.pyrDown(thresh)

mask=np.zeros((thresh.shape[0]+2, thresh.shape[1]+2),np.uint8)

counter = 0
for i in range(thresh.shape[0]):
    for j in range(thresh.shape[1]):
        if thresh[i][j]==255 and mask[i][j]==0:
            counter+=1
            cv2.floodFill(thresh, mask,(j,i), 0)
```

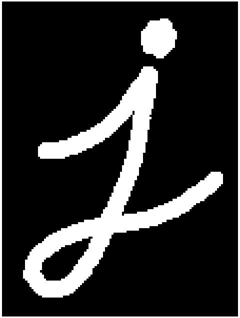
Primer slike

Images/Challenge2.jpg

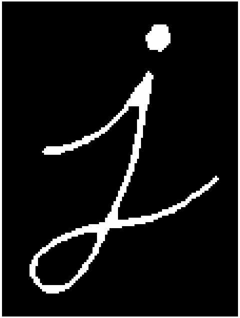
Binarizacija?



Morfološke operacije



(a) Original



(b) Erozija

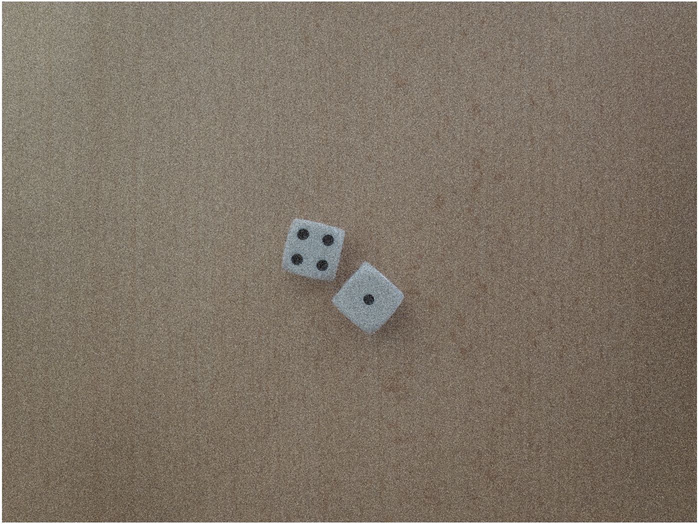
(c) Dilatacija

Morfološke operacije

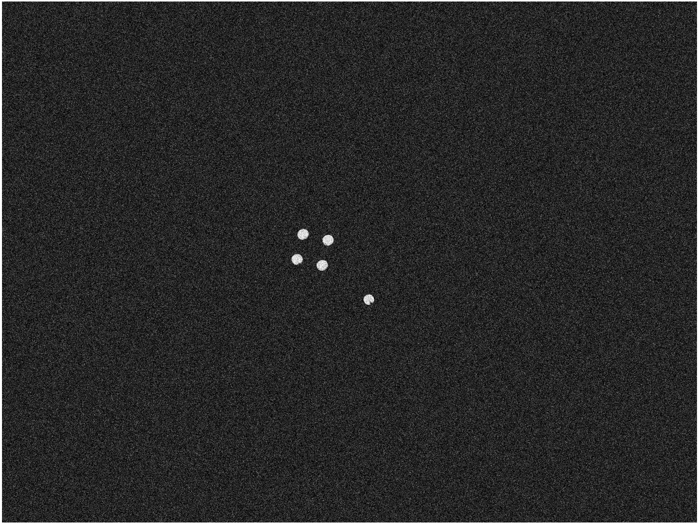
(a) Zatvaranje

(b) Otvaranje

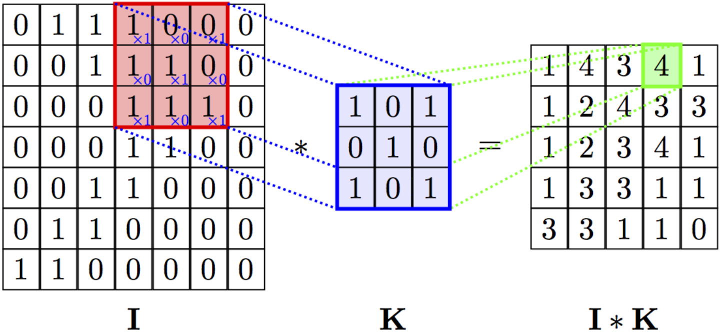
Šum na slikama



Šum na slikama



Konvolucioni filtri



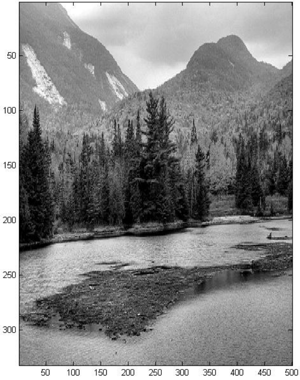
Konvolucioni filtri

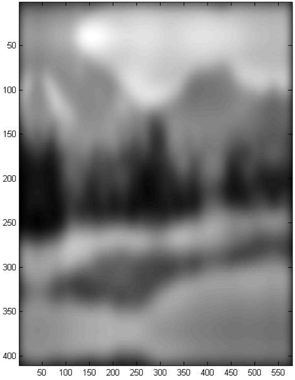
$$K = \frac{1}{K_{width} \cdot K_{height}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ . & . & . & \cdots & 1 \\ . & . & . & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$$

Gausov filtar

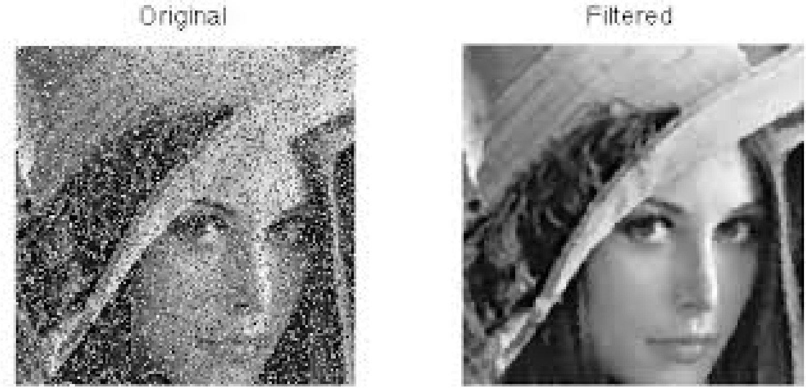
$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Gausov filtar



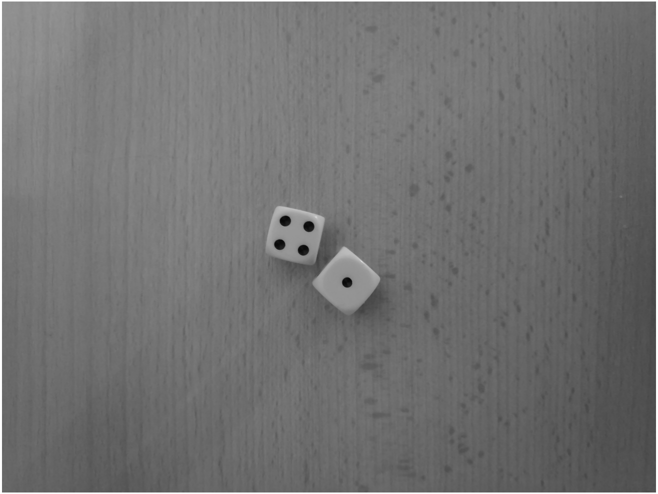


Medijanski filtar



Gausov vs medijanski

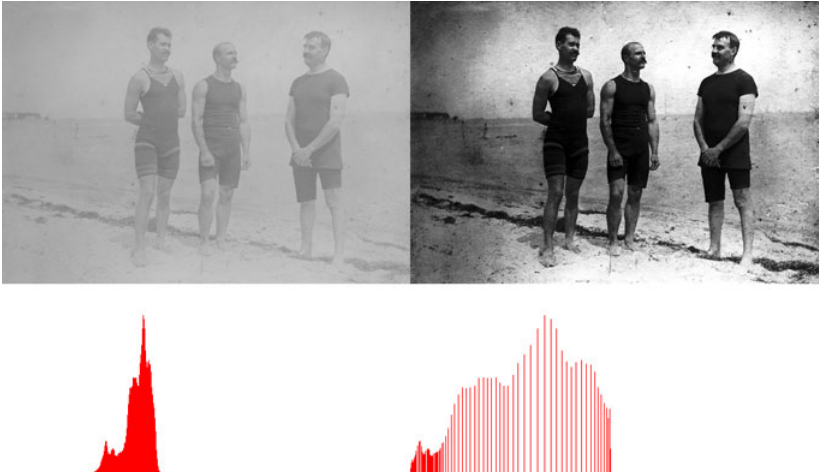
Odšumljena slika



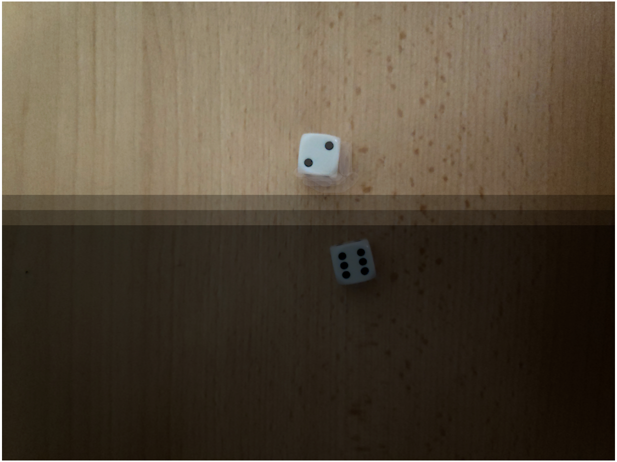
Slike u mračnijim/svetlijim okruženjima

Slike u mračnijim/svetlijim okruženjima

Histogram equalization

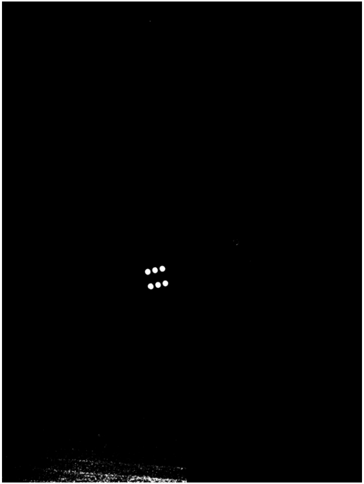


Slike u mračnijim/svetlijim okruženjima



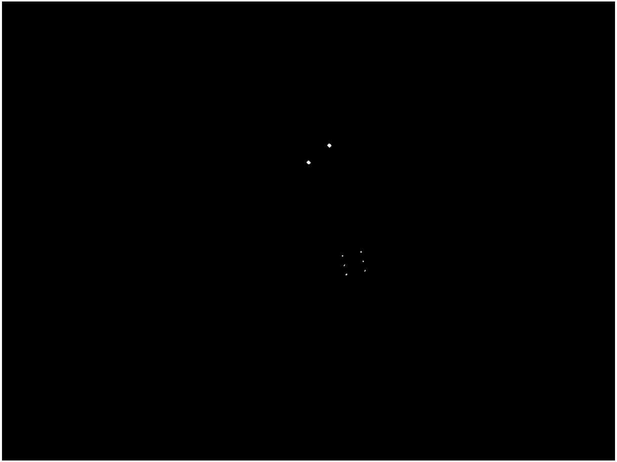
Običan Threshold

(a) slabo

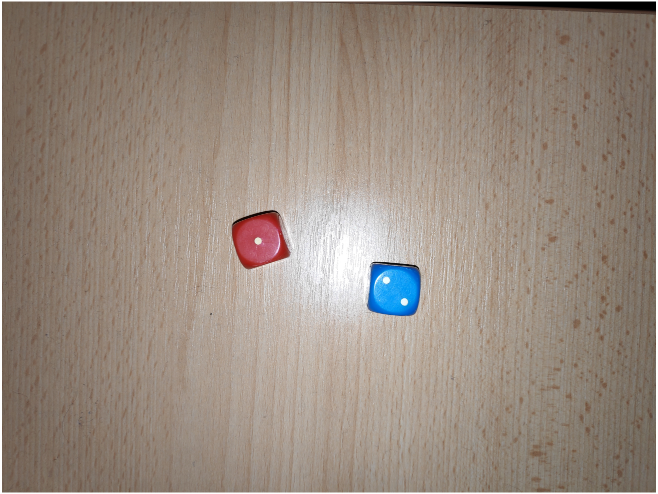


(b) jako

Adaptivni threshold



Kockice koje nisu crno-bele



Sobel edge detection

(a) Original

(b) Rezultat

Sobel edge detection

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I$$

(a) Horizontalni

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

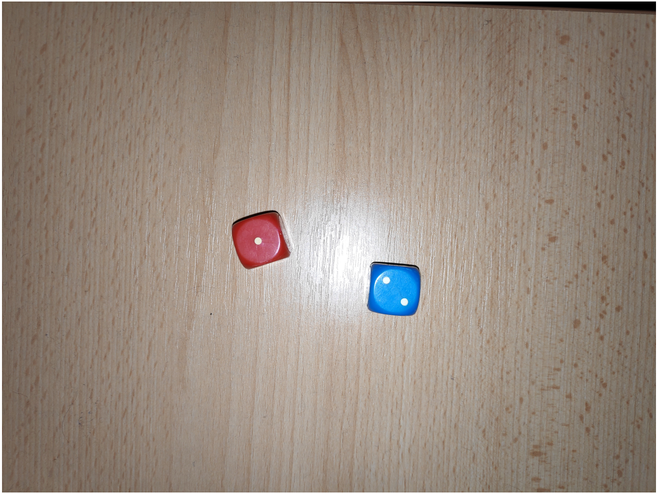
(b) Vertikalni

Canny edge detection

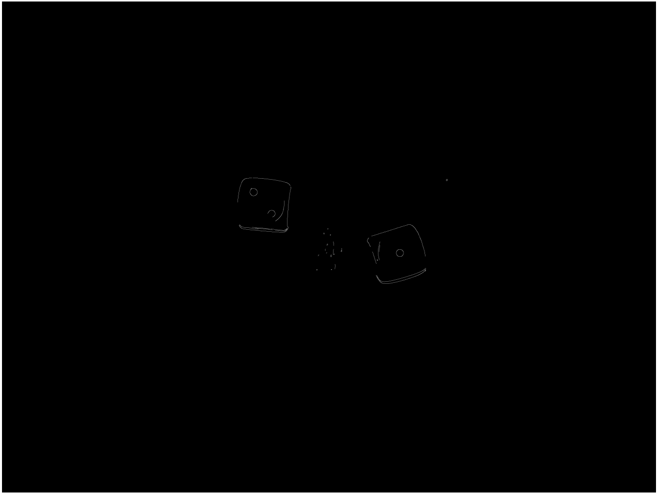
(a) Sobel

(b) Canny

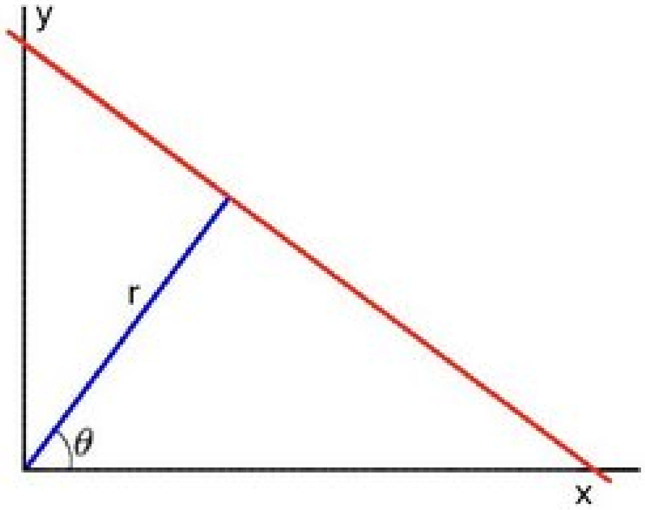
Canny edge detection



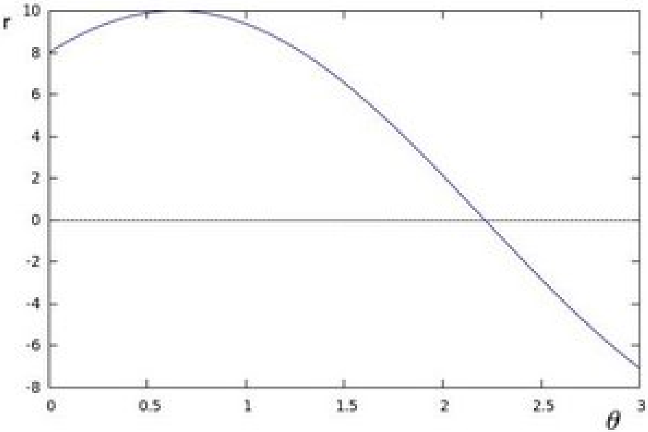
Canny edge detection



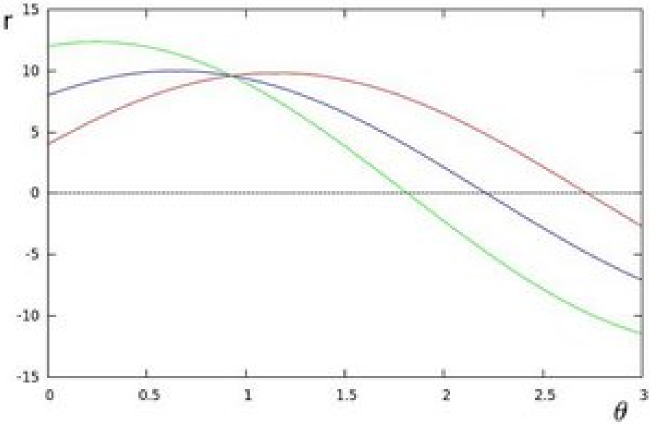
Hough Line Transform



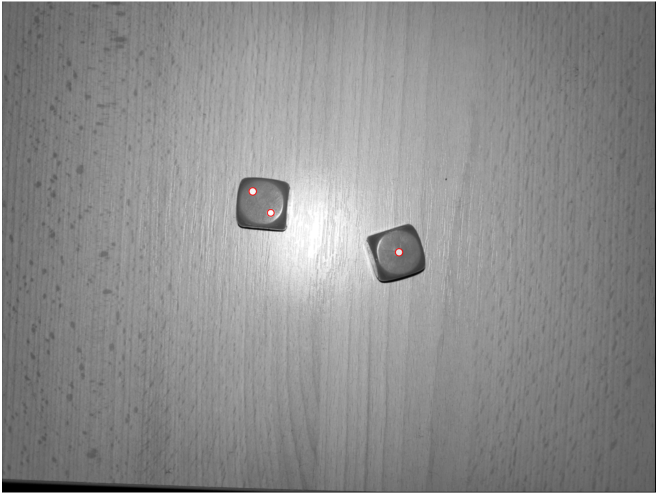
Hough Line Transform



Hough Line Transform



Hough Circle Transform



Zaključak

- Dalji rad

Zaključak

- Hvala na pažnji!