

Accesare pixeli, ponderari, trackbar, masca

November 27, 2018

Catalin Stoean
catalin.stoean@inf.ucv.ro
<http://inf.ucv.ro/~cstoean>

1 Citim si afisam o poza, apoi ii adaugam pixeli albi

```
In [14]: %matplotlib inline
import matplotlib.pyplot as plt #interfata pentru afisare de imagini
import cv2

im = cv2.imread('C:/pic.jpg')
imPILInit = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)

fig = plt.figure()
ax1 = fig.add_subplot(111)
ax1.imshow(imPILInit)
```

Out [14]: <matplotlib.image.AxesImage at 0x24bbfd7c8d0>



```
In [15]: import random
```

```

def zgomot(im, n):
    h, w, c = im.shape
    for k in range(n):
        i = random.randint(0, h - 1)
        j = random.randint(0, w - 1)
        if len(im.shape) == 2: #avem numai h si w, fara c
            im[i, j] = 255
        elif len(im.shape) == 3:
            im[i, j] = (255, 255, 255)
    return

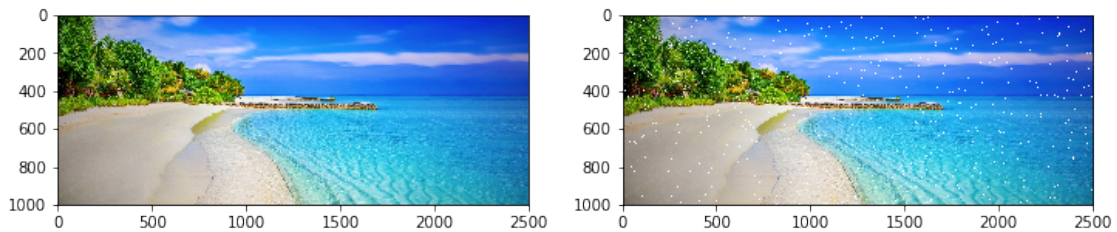
zgomot(im, 25000)

imPIL = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)

fig = plt.figure(figsize=(12, 8))
ax1 = fig.add_subplot(121)
ax1.imshow(imPILInit)
ax1 = fig.add_subplot(122)
ax1.imshow(imPIL)

```

Out [15]: <matplotlib.image.AxesImage at 0x24bc6b886d8>



2 Combinarea a doua poze

```

In [4]: im = cv2.imread('C:/pic.jpg')

imPILInit = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)

batman = cv2.imread('C:/batman.jpg')
h, w, _ = batman.shape

print('w = {}, h = {}'.format(w, h))

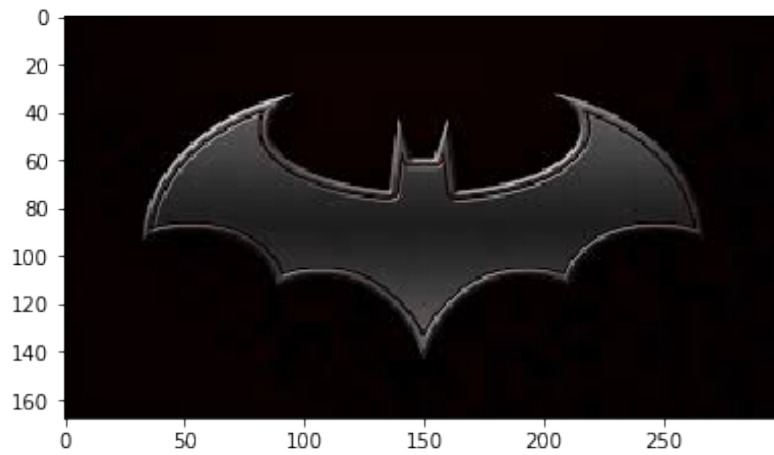
imROI = imPILInit[600:600 + h, 0:w]
imROI = cv2.addWeighted(imROI, 1, batman, 0.9, 0)
imPILInit[600:600 + h, 0:w] = imROI

```

```
fig = plt.figure(figsize=(12, 8))
ax1 = fig.add_subplot(211)
ax1.imshow(batman)
ax1 = fig.add_subplot(212)
ax1.imshow(imPILInit)
```

w = 300, h = 168

Out [4]: <matplotlib.image.AxesImage at 0x2da40d78e10>



3 Thresholding

```
In [16]: logo = cv2.imread('C:/ucv.png', 0)
         h, w = logo.shape
```

```

print('w = {}, h = {}'.format(w, h))

ret,thresh1 = cv2.threshold(logo, 127, 255, cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(logo, 127, 255, cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(logo, 127, 255, cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(logo, 141, 255, cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(logo, 127, 255, cv2.THRESH_TOZERO_INV)
titles = ['Imaginea initiala', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
images = [logo, thresh1, thresh2, thresh3, thresh4, thresh5]
fig = plt.figure(figsize=(8, 8))
for i in range(6):
    plt.subplot(3,2,i+1),plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([],plt.yticks([]))
plt.show()

```

w = 200, h = 208

Original Image



BINARY



BINARY_INV



TRUNC



TOZERO



TOZERO_INV



```
In [17]: im = cv2.imread('C:/pic.jpg')

imPILInit = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)
h, w = logo.shape
imROI = imPILInit[600:600 + h, 0:w]
color = cv2.cvtColor(thresh4, cv2.COLOR_GRAY2BGR)
imROI = cv2.addWeighted(imROI, 1, color, 0.9, 0)
imPILInit[600:600 + h, 0:w] = imROI

fig = plt.figure()
```

```
ax1 = fig.add_subplot(111)
ax1.imshow(imPILInit)
```

Out [17]: <matplotlib.image.AxesImage at 0x24bc742e4e0>



4 Trackbar

```
In [22]: import cv2
```

```
def nothing(x):
    pass
logo = cv2.imread('C:/ucv.png', 0)
numeFereastra = 'Exemplu cu trackbar - ESC pentru inchidere'
cv2.namedWindow(numeFereastra, cv2.WINDOW_NORMAL)

ret,thresh1 = cv2.threshold(logo, 1, 255, cv2.THRESH_BINARY)

cv2.createTrackbar('thresh', numeFereastra, 0, 255, nothing)
cv2.createTrackbar('threshMax', numeFereastra, 0, 255, nothing)

while(1):
    # luam pozitiile de la trackbars
    thresh = cv2.getTrackbarPos('thresh',numeFereastra)
    threshMax = cv2.getTrackbarPos('threshMax', numeFereastra)
    cv2.imshow(numeFereastra, thresh1)

    if cv2.waitKey(1) == 27: #incheiem cu ESC
        break

    ret,thresh1 = cv2.threshold(logo, thresh, threshMax, cv2.THRESH_BINARY)

cv2.destroyAllWindows()
```

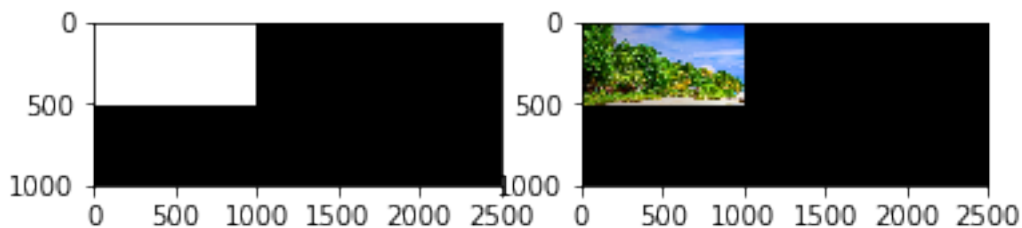
5 Masca in OpenCV

```
In [31]: import numpy as np
         im = cv2.imread('C:/pic.jpg')

         # Cream o imagine neagra
         masca = np.zeros(im.shape, dtype = "uint8")

         # Desenam un dreptunghi alb plin pe imaginea masca
         cv2.rectangle(masca, (0, 0), (1000, 500), (255, 255, 255), -1)
         im = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)
         # Aplicam masca si afisam masca si rezultatul
         #Era OK sa scriem si ca in comentariul de mai jos
         # imagineaMascata = im & masca
         imagineaMascata = cv2.bitwise_and(im, masca)
         fig = plt.figure()
         ax1 = fig.add_subplot(121)
         ax1.imshow(masca)
         ax2 = fig.add_subplot(122)
         ax2.imshow(imagineaMascata)
```

Out [31]: <matplotlib.image.AxesImage at 0x24bc1283dd8>



6 Detectarea obiectului principal dintr-o imagine

```
In [42]: import cv2
         import numpy as np
         import matplotlib.pyplot as plt

         im = cv2.imread("C:/ban.jpg", 0)

         # Aplicam thresholding binar astfel incat fundalul sa fie negru
         # Valoarea pragului de mai jos este foarte importanta pentru un rez bun
         _, imBinara = cv2.threshold(im, 180, 255, cv2.THRESH_BINARY_INV)

         # Facem o copie a imaginii binare.
         imFloodFill = imBinara.copy()
```

```

# Facem masca pentru flood filling
# Dimensiunea trebuie sa fie cu 2 pixeli mai mare decat imaginea.
#[:2] face sa ia doar primele 2 argumente (fara numarul de canale)
h, w = imBinara.shape[:2]
masca = np.zeros((h+2, w+2), np.uint8)

# Inundam pornind de la punctul (0, 0)
#pixelii ce nu sunt afectati sunt cei din interiorul regiunii delimitate
cv2.floodFill(imFloodFill, masca, (0,0), 255)

# In unele cazuri imFloodFill poate deja contine obiectul cautat

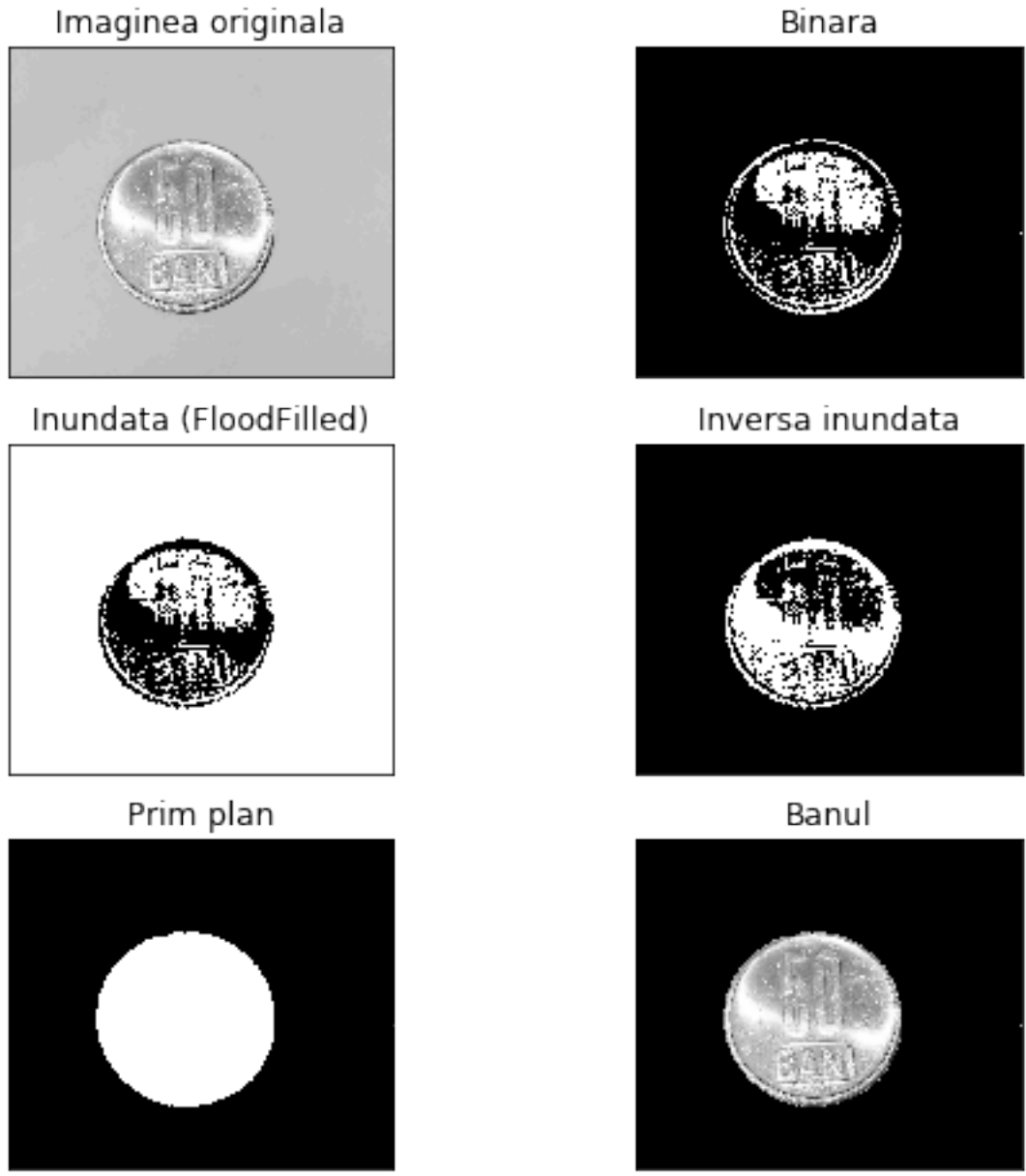
# Inversam imaginea inundata
imFloodFillInv = cv2.bitwise_not(imFloodFill)

'''
Combinam cele doua imagini imBinara si imFloodFill
ca sa obtinem prim planul, iar fundalul sa fie alb.
Se poate scrie si ca mai jos in loc de bitwise_or
imFinala = imBinara | imFloodFill
'''
imPrim = cv2.bitwise_or(imBinara, imFloodFillInv)

imBan = im & imPrim #sau cv2.bitwise_and(im, imPrim)

titles = ['Imaginea originala', 'Binara', 'Inundata (FloodFilled)',
          'Inversa inundata', 'Prim plan', 'Banul']
images = [im, imBinara, imFloodFill, imFloodFillInv, imPrim, imBan]
fig = plt.figure(figsize=(8, 8))
for i in range(6):
    plt.subplot(3,2,i+1),plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([],plt.yticks([]))
plt.show()

```

7 Exerciții

1. Faceti un trackbar pentru exemplul în care se adaugă un logo într-o poză pentru a varia ponderile pentru combinarea pozelor.
2. Faceti un trackbar pentru exemplul în care se caută obiectul din prim plan prin care să se stabilească valoarea pragului de la thresholding și care să permită alegerea THRESH_BINARY sau THRESH_BINARY_INV.