

Procesarea imaginilor digitale

Ene Cristian Ștefan

1. Noțiuni teoretice

1.1 Reprezentarea imaginilor

Imaginile digitale sunt structuri de date matriciale. Elementele ce compun imaginea se numesc pixeli și aceștia conțin informația despre cât de intensă este culoare. Se disting 3 tipuri de imagini:

- Imagini în scală de gri;
- Imagini color;
- Imagini binare

Imaginile în scală de gri, pixelii descriu diferite nuanțe de gri. În cazul imaginilor color, fiecare pixel este definit prin 3 intensități separate, câte una pentru fiecare culoare primară de lumină: roșu, verde, albastru.

Imaginile binare conțin numai alb sau negru, pixelii pot să ia doar două valori: 0 sau 1.

1.2 Histograme de imagini

O reprezentare de tip histogramă indică distribuția pixelilor în funcție de intensitate. Se pot realiza histogramme pentru fiecare canal (roșu, verde, albastru) sau o histogramă care să le cuprindă pe toate.

1.3 Convoluția matricilor

Prelucrarea imaginilor se realizează cu ajutorul operației de convoluție care are următoare formulă:

$y[m,n] = x[m,n] * h[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[m-i,n-j]$	(1)
--	-----

Unde:

- h = nucleu (kernel)
- x = matricea de intrare
- y = matricea de ieșire

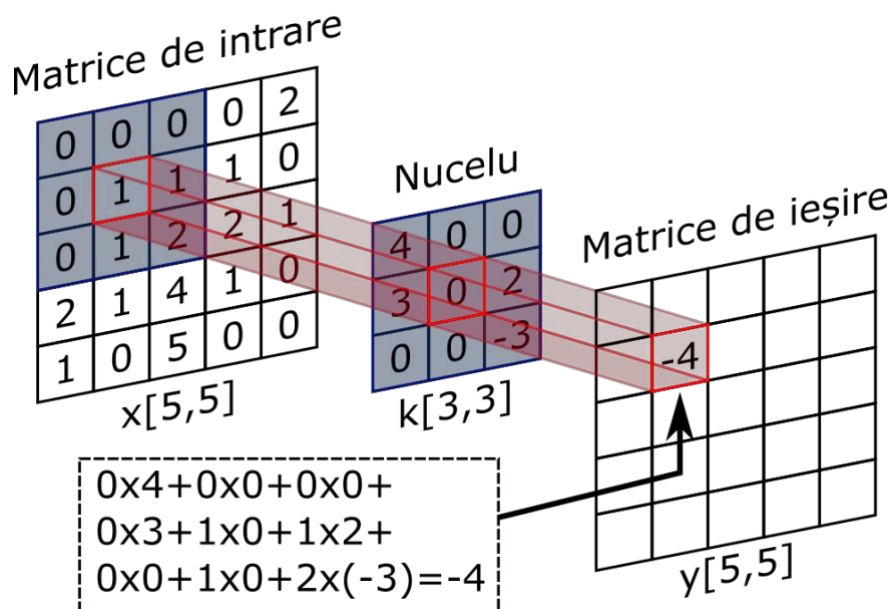


Figura 1. Exemplu de convoluție a matricilor

1.4 Filtre de imagine

Filtrele de imagine sunt aplicații ale convoluției. Un filtru este definit prin nucleu. Exemplu de filtre:

Tabelul 1. Exemple de filtre

Denumire	Nucleu	Efect
Contur(Outline)	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	Păstrează doar conturul elementelor din imagine
Reliefare (Emboss)	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$	Accentuează diferențele dintre pixeli, oferind iluzia adâncimii în imagine
Ascuțire (Sharpen)	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	Opusul filtrului de estompare
Estompare gaussiană 3x3	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	Slăbirea intensității unei culori sau a unor contururi

Pentru a folosi filtrele pe imagini color este necesar ca filtrul să se aplice pe fiecare canal de culoare.

2. Rezolvarea cerințelor

1. Încărcați o imagine color și reprezentați cele 3 canale de culoare primară.

```
import cv2
import matplotlib.pyplot as plt
I=cv2.imread('Imagine1.png',1)
I=cv2.cvtColor(I, cv2.COLOR_BGR2RGB)
R=I[:, :,0] #Salveaza componenta 0 pentru fiecare element din matrice
G=I[:, :,1] #Salveaza componenta 1 pentru fiecare element din matrice
B=I[:, :,2] #Salveaza componenta 2 pentru fiecare element din matrice
plt.figure(0)
plt.subplot(2,2,1)
plt.imshow(R, cmap='gray', vmin=0, vmax=255)
plt.title('Canal R')
plt.axis('off')
plt.subplot(2,2,2)
plt.imshow(G, cmap='gray', vmin=0, vmax=255)
plt.title('Canal G')
plt.axis('off')
plt.subplot(2,2,3)
plt.imshow(B, cmap='gray', vmin=0, vmax=255)
plt.title('Canal B')
plt.axis('off')
plt.subplot(2,2,4)
plt.imshow(I)
plt.title('Imagine color')
plt.axis('off')
plt.tight_layout()
plt.show()
```

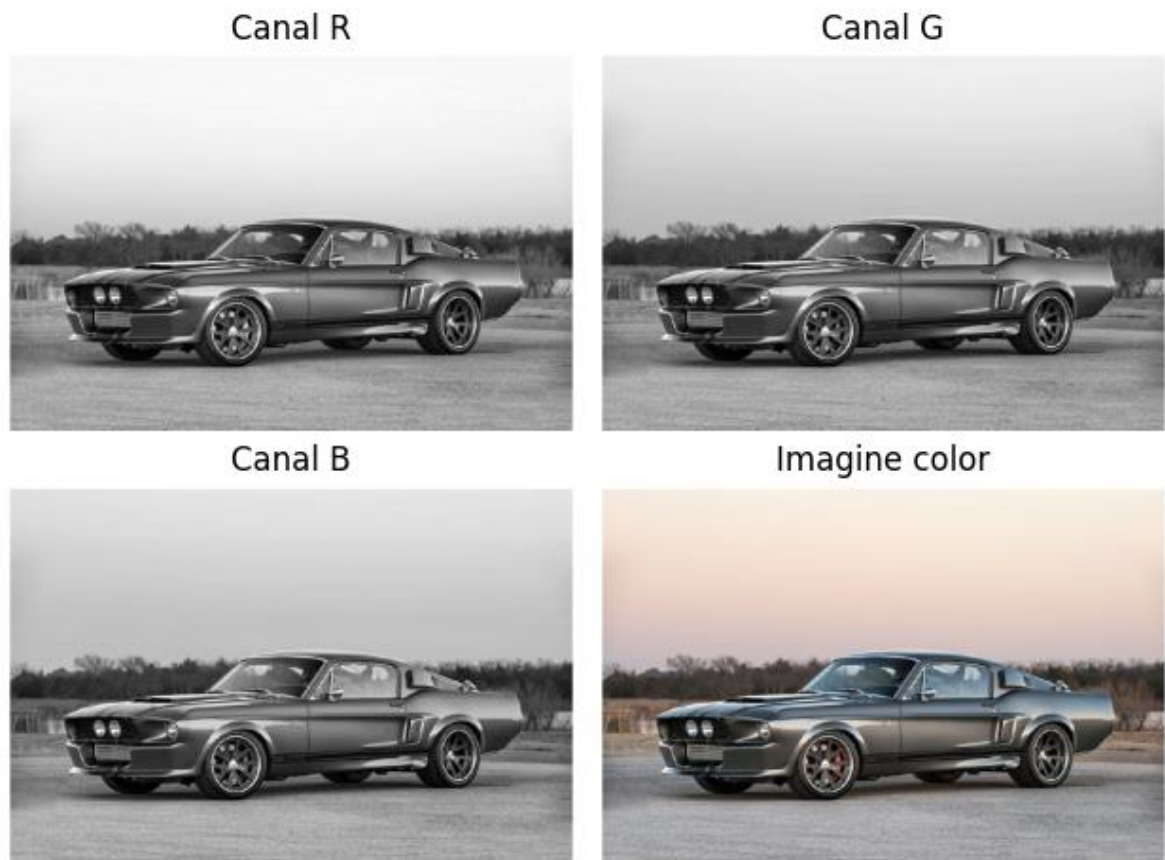


Figura 2. Rezultat cerința 1

2. Realizați reprezentările de tip histogramă pentru cele 3 canale de culoare primară pentru întreaga imagine.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
I=cv2.imread('Imagine1.png',1)
I= cv2.cvtColor(I, cv2.COLOR_BGR2RGB)
R=I[:, :, 0]
G=I[:, :, 1]
B=I[:, :, 2]
plt.subplot(2,2,1)
plt.hist(R.ravel(),256,[0,256])
plt.title('Histograma Rosu')
plt.subplot(2,2,2)
plt.hist(G.ravel(),256,[0,256])
plt.title('Histograma Verde')
plt.subplot(2,2,3)
plt.hist(B.ravel(),256,[0,256])
plt.title('Histograma Albastru')
plt.subplot(2,2,4)
plt.hist(I.ravel(),256,[0,256])
```

Prelucrarea digitală a semnalelor

```
plt.title('Histograma totala')  
plt.tight_layout()  
plt.show()
```

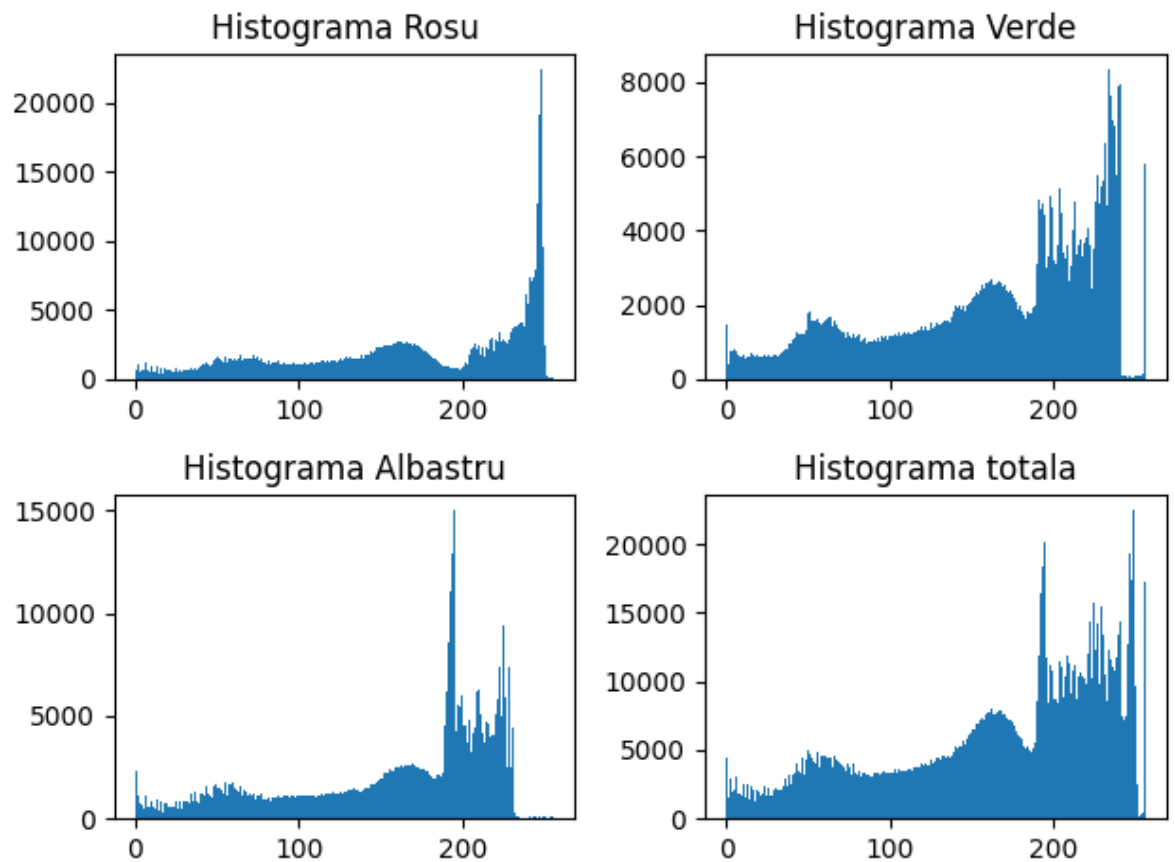


Figura 3. Rezultat cerința 2

3. Transformați imaginea în scală gri și realizați afișarea acesteia.

```
import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
I=cv2.imread('Imagine1.png',1)  
I= cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)  
plt.plot()  
plt.imshow(I,cmap = 'gray',vmin=0, vmax=255)  
plt.title('Imagine Gri')  
plt.axis('off')  
plt.tight_layout()  
plt.show()
```

Imagine Gri



Figura 4. Rezultat cerința 3

4. Aplicați un filtru gaussian 3x3 imaginii.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
I=cv2.imread('Imagine1.png',1)
I= cv2.cvtColor(I, cv2.COLOR_BGR2RGB)
G3 = 1/16*np.array([[1, 2, 1],[2, 4, 2],[1, 2, 1]])#nucleu gaussian 3x3
F = np.zeros((I.shape))
F = cv2.filter2D(src=I, ddepth=-1, kernel=G3)
plt.imshow(F.astype('uint8'))
plt.axis('off')
plt.show()
```



Figura 5. Rezultat cerința 4

5. Aplicați un filtru de ascuțire (sharpen) imaginii.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
I=cv2.imread('Imagine1.png',1)
I= cv2.cvtColor(I, cv2.COLOR_BGR2RGB)
SH = np.array([[0, -1, 0],[-1, 5, -1],[0, -1, 0]])
F = np.zeros((I.shape))
F = cv2.filter2D(src=I, ddepth=-1, kernel=SH)
plt.imshow(F.astype('uint8'))
plt.axis('off')
plt.show()
```



Figura 6. Rezultat cerința 5

6. Aplicați un filtru de contur imaginii.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
I=cv2.imread('Imagine1.png',1)
I= cv2.cvtColor(I, cv2.COLOR_BGR2RGB)
SH = np.array([[ -1,  -1,  -1],[ -1,  8,  -1],[ -1,  -1,  -1]])
F = np.zeros((I.shape))
F = cv2.filter2D(src=I, ddepth=-1, kernel=SH)
plt.imshow(F.astype('uint8'))
plt.axis('off')
plt.show()
```



Figura 7. Rezultat cerința 6

7. Aplicați un filtru definit de voi imaginii

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
I=cv2.imread('Imagine1.png',1)
I= cv2.cvtColor(I, cv2.COLOR_BGR2RGB)
SH = np.array([[ -2,  -1,  0],[ -1,  1,  1],[ 0,  1,  2]])
F = np.zeros((I.shape))
F = cv2.filter2D(src=I, ddepth=-1, kernel=SH)
plt.imshow(F.astype('uint8'))
plt.axis('off')
plt.show()
```




Figura 8. Rezultat cerința 7