

Metode de îmbunătățire a imaginilor cu lumină scăzută

Transformări de Nuanțe de Gri și Egalizarea Histogramelor

Membrii echipei

Manager: Lascu Anamaria-Alexandra

Dezvoltator: Marte Daniela

Cercetător: Moț Patricia Alexandra

Nume îndrumător: Camelia Costina Florea

Data predării proiectului (zi/lună/an): 13/12/2023

Cuprins

1. Abstract
2. Introducere
3. Fundamentare teoretică
4. Implementarea soluției adoptate
5. Rezultate experimentale
6. Concluzii
7. Bibliografie

1. Abstract

Acest proiect se concentrează asupra îmbunătățirii imaginilor cu iluminare redusă, abordând transformările în tonuri de gri și egalizarea histogramelor. Iluminarea redusă, cauzată de condiții precum timpul de noapte, furtuni sau spații interioare, duce la imagini cu histogramă îngustă, luminozitate medie scăzută și susceptibilitate crescută la zgomot de captare. Scopul principal este îmbunătățirea contrastului global și local, precum și a luminozității medii, fără a amplifica zgomotul sau a reduce claritatea, în special în cazul imaginilor color.

Pentru realizarea proiectului am utilizat limbajul de programare Python, folosindu-ne de mediul de dezvoltare Jupyter din Anaconda. Aceste funcții operează asupra componentei de luminozitate a imaginii, permitând ajustarea parametrilor de către utilizator în unele cazuri. Rezultatul constă într-o imagine îmbunătățită în comparație cu imaginea de intrare.

Cuvinte cheie: îmbunătățire imagini, iluminare redusă, transformări în tonuri de gri, egalizare histogramă

- Date de intrare:
 - Imagini cu iluminare redusă.
 - Histogramele imaginilor inițiale
- Date de ieșire:
 - Imagini îmbunătățite în aspect global și local, fără amplificarea zgomotului sau reducerea clarității.
 - Histogramele imaginilor îmbunătățite

2. Introducere

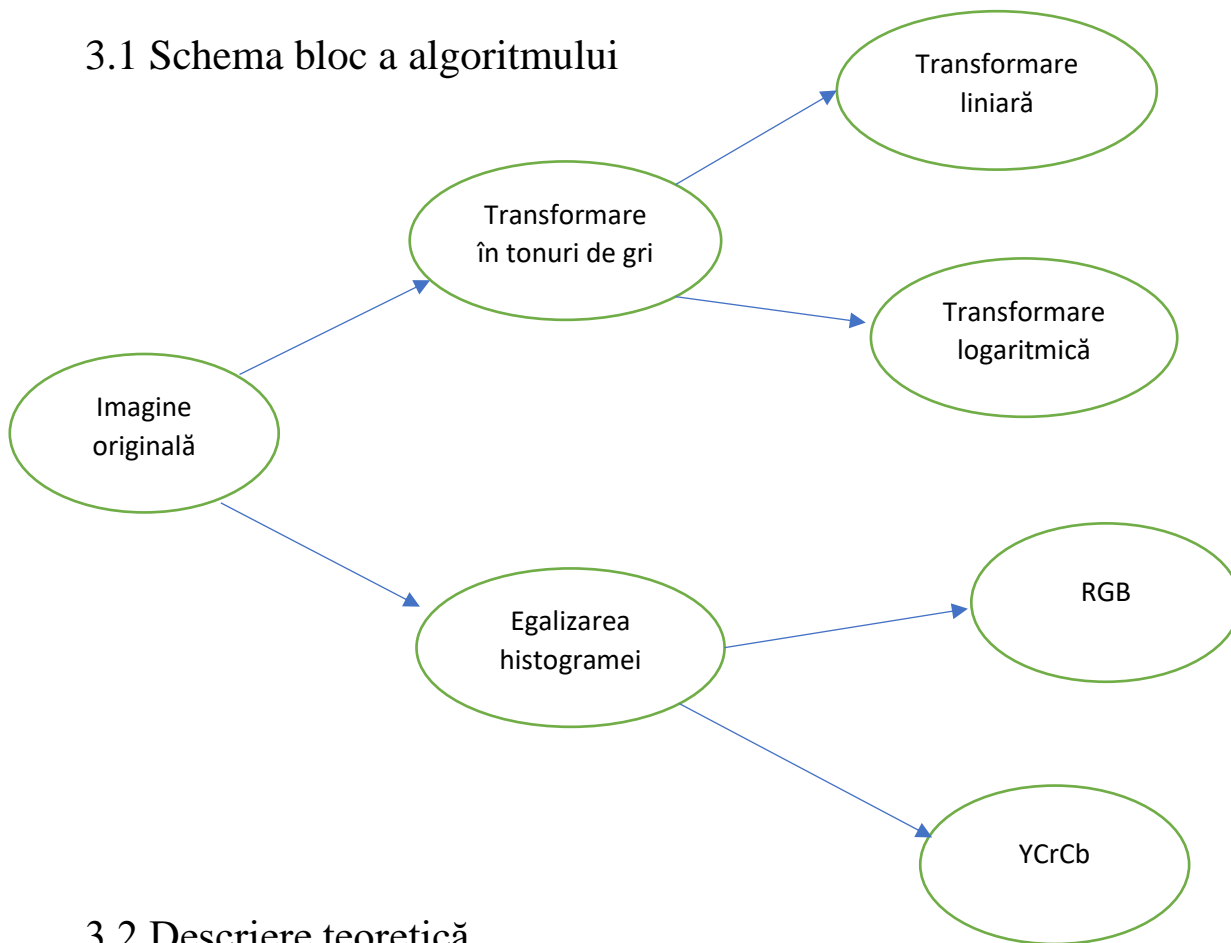
În cadrul proiectului, am abordat problema îmbunătățirii imaginilor cu iluminare redusă, concentrându-ne pe două metode distincte.

Prima metodă implică transformări în tonuri de gri, inclusiv transformări liniare și non-liniare, pentru optimizarea contrastului. În plus, am introdus o abordare inovatoare folosind transformări logaritmice pentru a evidenția detalii esențiale. A doua metodă explorează algoritmul de egalizare a histogramelor pe canalele RGB și pe canalul de intensitate în spațiul YCrCb.

Rezultatele preliminare evidențiază îmbunătățiri semnificative ale calității imaginilor, oferind perspective valoroase pentru optimizarea performanței sistemelor în condiții de iluminare redusă.

3. Fundamentare teoretică

3.1 Schema bloc a algoritmului



3.2 Descriere teoretică

Proiectul propune două metode fundamentale pentru îmbunătățirea imaginilor cu iluminare redusă: Transformări în Tonuri de Gri și Egalizarea Histogramelor.

➤ Transformări în Tonuri de Gri:

a. Transformări liniare:

- Aplicarea unei transformări liniare pe fiecare canal al imaginii (roșu, verde, albastru) pentru a ajusta intervalul de luminozitate. Această ajustare se face pentru a optimiza distribuția luminanței și a contrastului în fiecare canal.

- Transformările liniare ajustează intervalul de intensitate al imaginii conform ecuației generale:

$$g(x, y) = \frac{f(x, y) - f_{min}}{f_{max} - f_{min}} \times (g_{max} - g_{min}) + g_{min} \quad [1]$$

Unde:

- $g(x,y)$ reprezintă pixelul rezultat,
- $f(x,y)$ este pixelul inițial,
- f_{min}, f_{max} reprezintă valorile minime și, respectiv, maxime ale intensităților din imaginea inițială
- g_{min}, g_{max} reprezintă valorile minime și, respectiv, maxime ale intensităților în imaginea rezultată

Apoi, imaginea rezultată este normalizată pentru a aduce valorile pixelilor într-un interval specific, acesta fiind între 0 și 255.

b. Transformări neliniare (Logaritmice):

- Deoarece s-a putut observa ineficacitatea transformărilor liniare în îmbunătățirea imaginilor cu iluminare redusă, utilizăm o transformare logaritmică pentru a îmbunătăți contrastul și vizibilitatea detaliilor în zonele întunecate.

- Transformările logaritmice evidențiază detalii în zonele întunecate, urmând ecuația:

$$g(x, y) = \log(1 + c \times f(x, y)) \quad [1]$$

- Unde c este o constantă de scalare.

➤ Egalizarea Histogramelor:

a. Egalizarea Histogramelor pe Canale RGB:

-Egalizarea histogramei este realizată pe fiecare canal RGB pentru a obține o distribuție uniformă a intensităților. Această abordare are ca rezultat o îmbunătățire globală a contrastului și a luminozității în imagine, utilizând formula:

$$g(x, y) = \frac{f(x,y) - \min}{\max - \min} \times 255 \quad [1]$$

- Unde \min și \max reprezintă valorile minime și, respectiv, maxime ale intensităților pe canal.

-Observarea schimbărilor în intensitate și contrast pe fiecare canal RGB permite evaluarea impactului egalizării histogramei asupra aspectului general al imaginii.

b. Egalizarea Histogramelor pe Canalul de Intensitate în Spațiul YCrCb:

Convertirea imaginii în spațiul de culoare YCrCb pentru a separa informațiile de luminanță (Y), culoare roșie minus luminanță (Cr), și culoare albastră minus luminanță (Cb), și aplicarea egalizării histogramei asupra canalului de luminanță pentru a accentua detaliile.

Egalizarea histogramei pe canalul de luminanță se efectuează în spațiul YCrCb, menținând culorile originale, folosind formula:

$$g(x,y)=\text{equalizeHist}(\text{luminance}(x,y))$$

- Unde `equalizeHist` este funcția de egalizare a histogramei, iar `luminance(x,y)` reprezintă canalul de luminanță al imaginii.

- După egalizarea histogramei, imaginea este convertită înapoi la formatul RGB pentru a putea fi comparată cu imaginea originală și pentru a evidenția îmbunătățirile aduse de acest proces.

Această fundamentare teoretică oferă o înțelegere detaliată a metodelor propuse pentru îmbunătățirea imaginilor cu iluminare redusă. Transformările în tonuri de gri și egalizarea histogramei reprezintă abordări complementare, punând accent pe ajustarea contrastului și uniformizarea distribuției de intensități. Aceste tehnici formează baza algoritmilor implementați în proiect pentru a obține rezultate vizuale superioare în condiții de iluminare redusă.

4. Implementarea soluției adoptate

Implementarea proiectului a fost realizată în Python, folosind mediul de programare Jupyter Notebook din platforma Anaconda.

Librăriile utilizate

```
import cv2
from matplotlib import pyplot as plt
import numpy as np
```

- cv2: utilizată pentru manipularea imaginilor
- pyplot: utilizată pentru reprezentarea de grafice și figuri
- numpy: utilizată pentru calculul numeric în Python, pentru manipularea matricilor

Citire și afișare imagine inițială

```
img = cv2.imread("poza2.jpeg", cv2.IMREAD_COLOR)

plt.imshow(img)
plt.title('Image with low light (input)')
plt.show()
```

- Funcția *cv2.imread* citește imaginea de intrare. Aceasta furnizează imaginea în formatul BGR, adică fiecare pixel este reprezentat de 3 valori: intensitatea canalului albastru, verde și roșu.
- Ulterior, imaginea e afișată folosind funcția *plt.imshow* din biblioteca Matplotlib. Aceasta preia imaginea citită și o afișează într-un plot. Titlul imaginii este setat cu ajutorul funcției *plt.title*, iar apoi imaginea este arătată pe ecran cu *plt.show()*.

Afișare imagine pentru fiecare canal (R, G, B)

```
plt.figure()
plt.imshow(img[:, :, 0], cmap = 'gray')
plt.suptitle('Red channel of input image')
plt.show()
```

În mod similar se afișează și pentru canalele Verde și Albastru.

Histograma canalelor (Roșu, Verde, Albastru)

- Afișarea histogramelor pentru fiecare canal al imaginii de intrare.

```
plt.hist(img[:, :, 0].ravel(), 256, [0, 255])
plt.suptitle('Linear histogram of red channel.')
plt.show()
```

În mod similar se afișează și pentru canalele Verde și Albastru.

Implementare transformări în tonuri de gri pentru fiecare canal

I) Transformări liniare

Aplicarea unei transformări liniare și normalizarea imaginii pentru a optimiza intervalul de luminozitate.

Apply linear transformation $g(x, y) = \frac{f(x, y) - f_{min}}{f_{max} - f_{min}} (g_{max} - g_{min}) + g_{min}$ on image.

```
[ ] img_normalized = cv2.normalize(img, None, 0, 255,
cv2.NORM_MINMAX, dtype=cv2.CV_8U)
```

Aplicarea imaginilor și histogramele procesate

```
plt.figure()
plt.imshow(img_normalized[:, :, 0], cmap = 'gray')
plt.suptitle('Red channel of linear gray scaled image')
plt.show()
```

```
plt.hist(img_normalized[:, :, 0].ravel(), 256, [0, 255])
plt.suptitle('Linear hitogram of red channel.')
plt.show()
```

În mod similar se afișează și pentru canalele Verde și Albastru.

Comparație între imaginea inițială și cea modificată

```
plt.figure()

plt.imshow(img)
plt.title('Original image')
plt.show()

plt.imshow(img_normalized)
plt.title('Image with low light (Linear gray transformation)')
plt.show()
```

II) Transformări neliniare (Logaritmice)

Aplicarea unei transformări logaritmice pentru a îmbunătăți contrastul în zonele întunecate

```
# Apply log transformation method
c = 255 / np.log(1 + np.max(img))
log_image = c * (np.log(img + 1))

# Specify the data type so that
# float value will be converted to int
log_image = np.array(log_image, dtype = np.uint8)
```

[4], [5]

Asemănător cu implementarea transformării liniare, vom afișa imaginile procesate, histogramele acestora și vom face o comparație între imaginea inițială și cea nouă.

Implementarea egalizării histogramei

I) Egalizare pe canale RGB

Aplicarea algoritmului de egalizare a histogramei pentru fiecare canal RGB

```
R, G, B = cv2.split(img)

output1_R = cv2.equalizeHist(R)
output1_G = cv2.equalizeHist(G)
output1_B = cv2.equalizeHist(B)
```

Afișarea imaginilor procesate și histograma acestora

```
plt.hist(output1_R.ravel(), 256, [0,255])
plt.suptitle('Linear histogram of red channel.')
plt.show()

plt.figure()
plt.imshow(output1_R, cmap = 'gray')
plt.suptitle('Red channel of linear gray scaled image')
plt.show()
```

Similar pentru canalele Verde și Albastru

Comparație între imaginea inițială și cea modificată

```
img_eq_hist = cv2.merge((output1_R, output1_G, output1_B))

plt.figure()

plt.imshow(img)
plt.title('Original image')
plt.show()

plt.imshow(img_eq_hist)
plt.title('Image with equalized hist')
plt.show()
```


II) Egalizare în Spațiul YCrCb

Conversie la Spațiul de Culoare YCrCb

- Utilizarea funcției `cv2.cvtColor` pentru a converti imaginea la spațiul de culoare YCrCb (Luminance, Chroma red, Chroma blue)

```
img_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)
```

Afișarea Imaginii pe Canalul de Intensitate

- Afișarea imaginii pe canalul de intensitate al spațiului YCrCb.

```
plt.figure()
plt.imshow(img_hsv[:, :, 0], cmap = 'gray')
plt.suptitle('Intensity channel of image')
plt.show()
```

Aplicare Egalizarea Histogramei pe Canalul de Intensitate

- Aplicarea funcției `cv2.equalizeHist` pe canalul de intensitate al imaginii în spațiul YCrCb.

```
img_hsv[:, :, 0] = cv2.equalizeHist(img_hsv[:, :, 0])
```

Afișarea imaginii pe canalul de Intensitate după egalizare

```
plt.figure()
plt.imshow(img_hsv[:, :, 0], cmap = 'gray')
plt.suptitle('Intensity channel of image')
plt.show()
```

Afișarea histogramei canalului de Intensitate după egalizare

```
plt.hist(img_hsv[:, :, 0].ravel(), 256, [0, 255])
plt.suptitle('Linear hitogram of intensity channel.')
plt.show()
```

Conversie înapoi la formatul RGB și comparație între imaginea de intrare și cea de ieșire

- Utilizarea funcției `cv2.cvtColor` pentru a converti imaginea înapoi la formatul RGB.

```
img_converted_back = cv2.cvtColor(img_hsv, cv2.COLOR_YCrCb2BGR)

plt.figure()

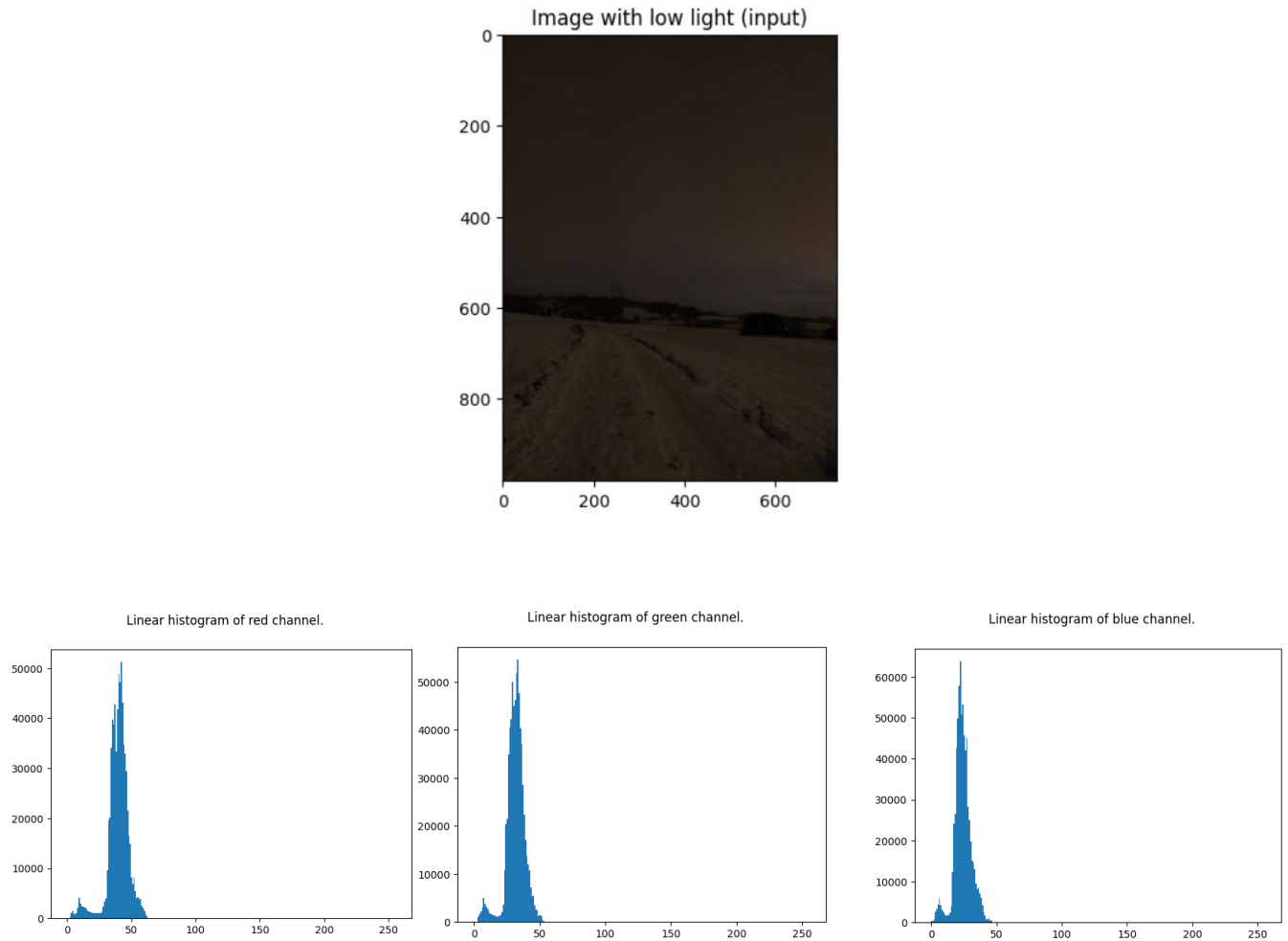
plt.imshow(img)
plt.title('Original image')
plt.show()

plt.imshow(img_converted_back)
plt.title('Image with intensity channel hist equalized')
plt.show()
```

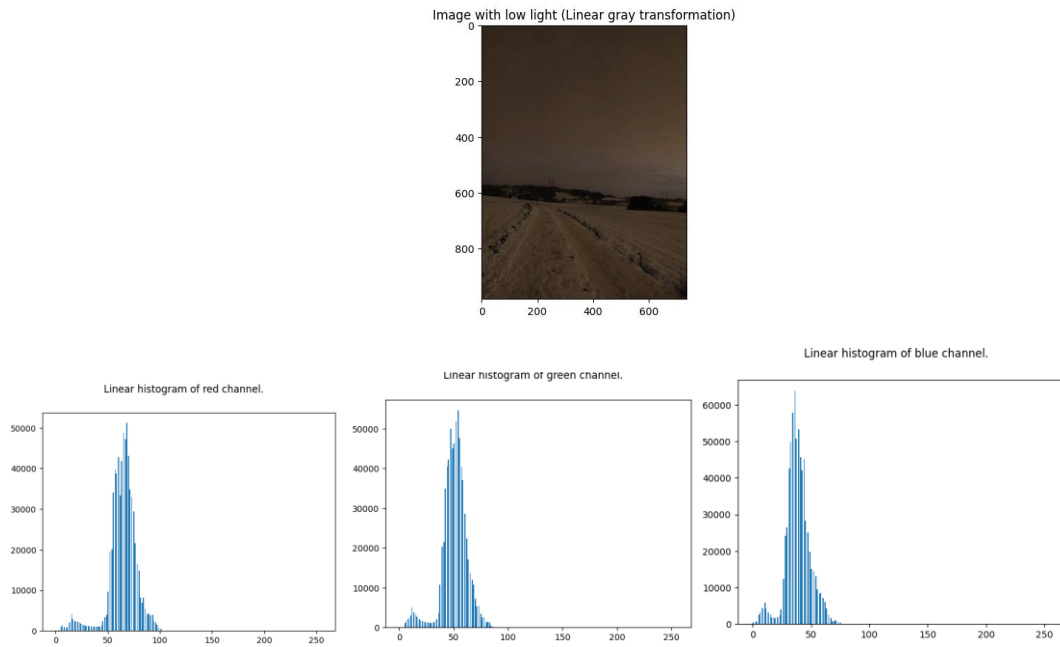
5. Rezultate experimentale

Prima imagine

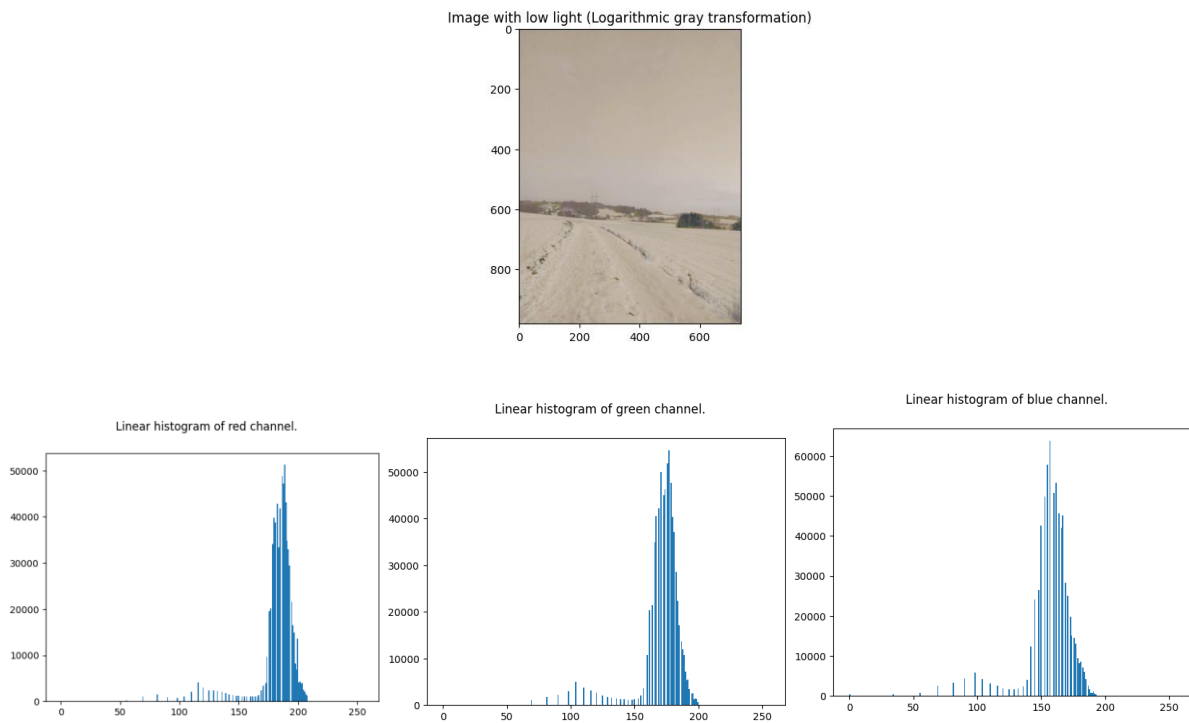
- Imaginea inițială și histogramele pe cele 3 canale (R, G, B)



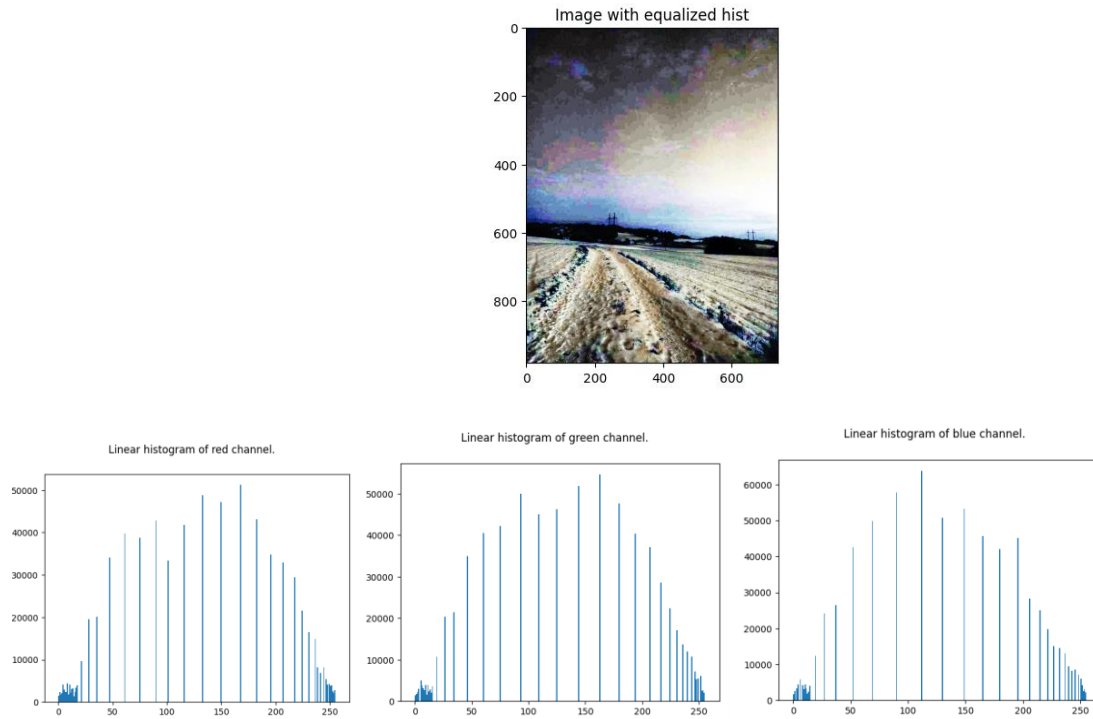
- Imaginea îmbunătățită și histogramele R, G, B pentru transformarea liniară



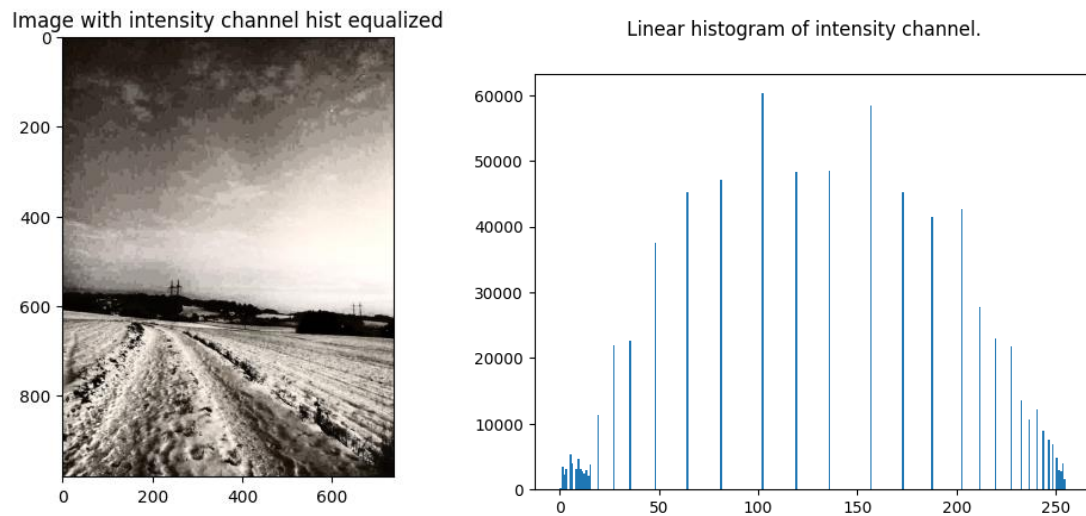
- Imaginea îmbunătățită și histogramele pentru transformarea logaritmică



- Imaginea îmbunătățită și histogramele pentru egalizarea histogramei pe canale RGB

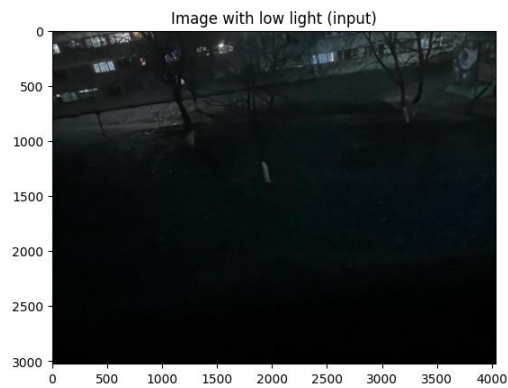


- Imaginea îmbunătățită și histograma pentru egalizarea histogramei pe canalul de intensitate

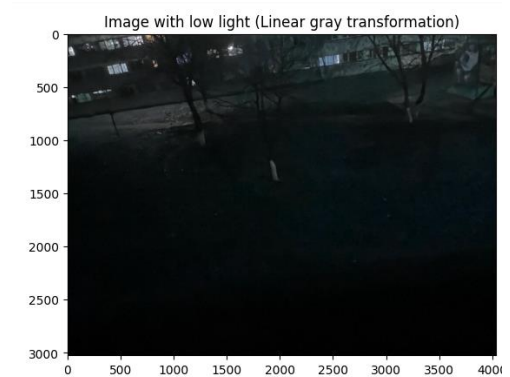


A doua imagine

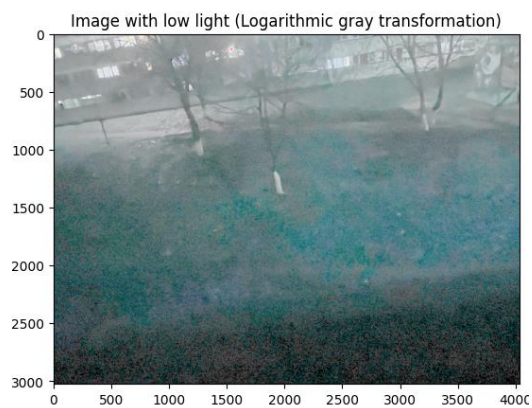
Imaginea inițială



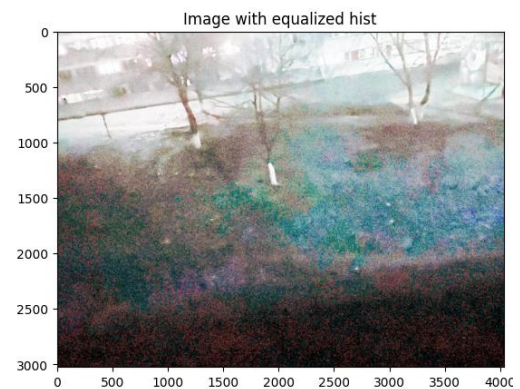
Tranformarea liniară Gray



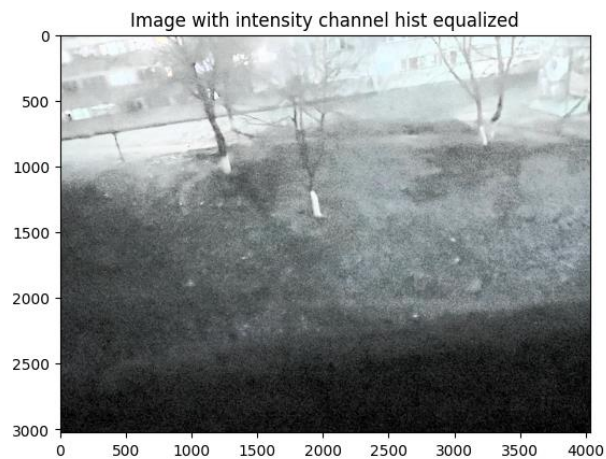
Transformarea logaritmică Gray



Egalizarea Histogramei pe canale RGB



Egalizarea Histogramei pe canale de intensitate (YCrCb)



A treia imagine

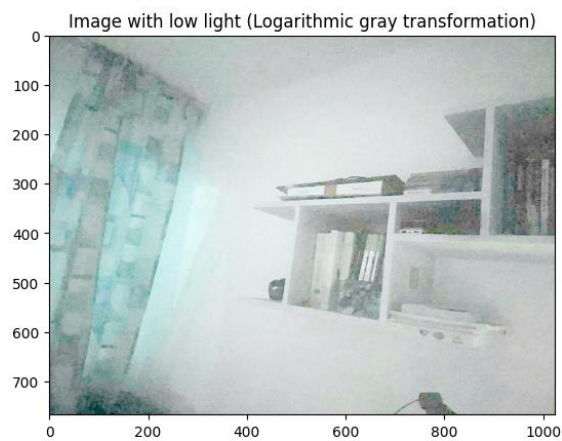
Imaginea inițială



Transformarea liniară Gray



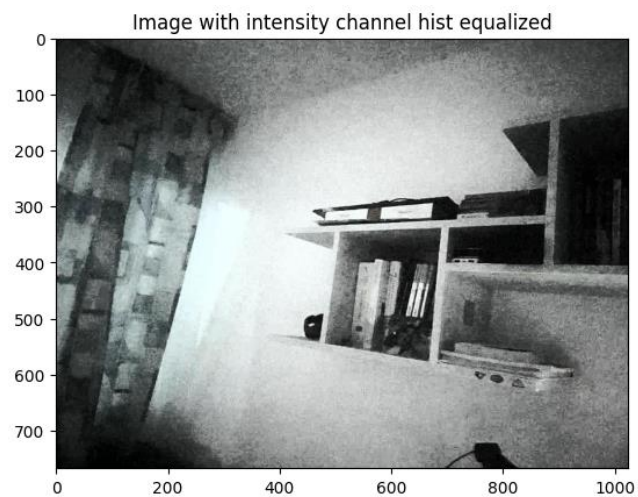
Transformarea logaritmică Gray



Egalizarea Histogramei pe canale RGB

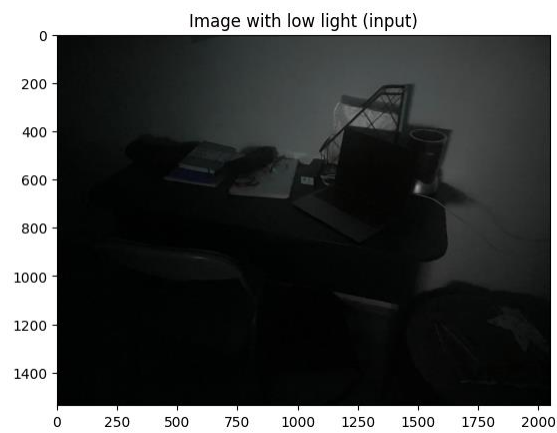


Egalizarea Histogramei pe canale de intensitate (YCrCb)

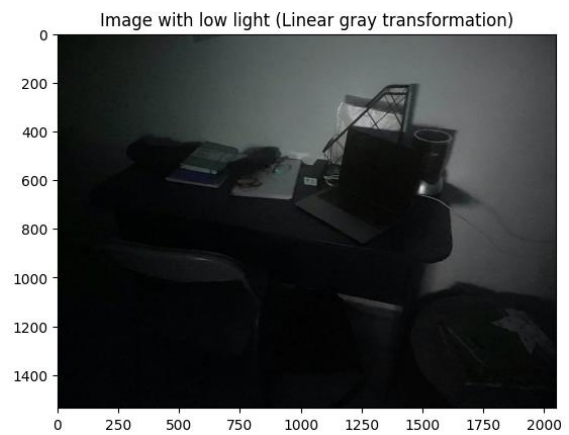


A patra imagine

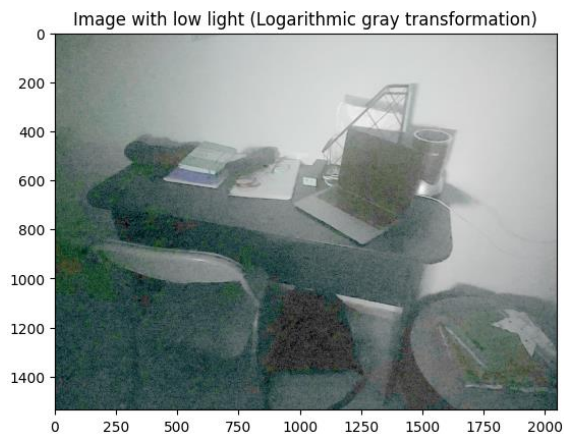
Imaginea inițială



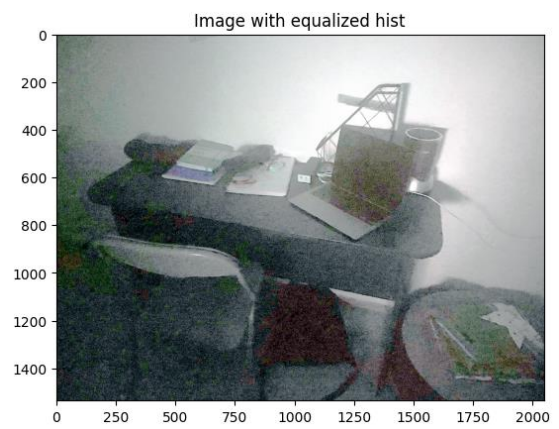
Transformarea liniară Gray



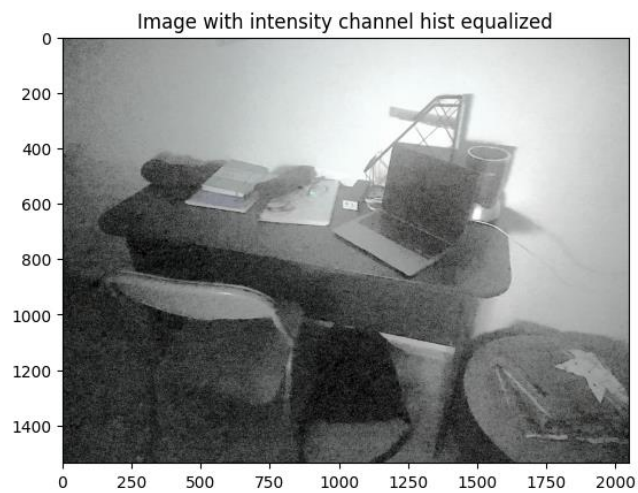
Transformarea logaritmică Gray



Egalizarea Histogramei pe canale RGB

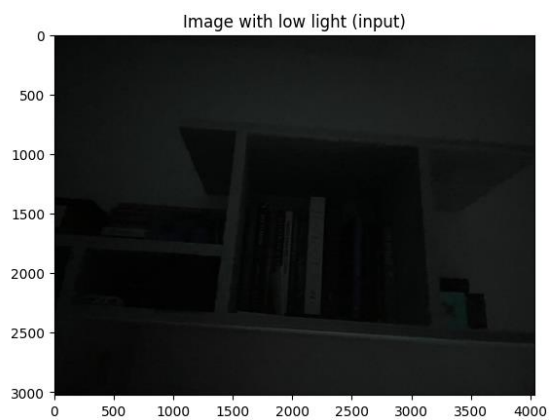


Egalizarea Histogramei pe canale de intensitate (YCrCb)



A cincea imagine

Imaginea inițială



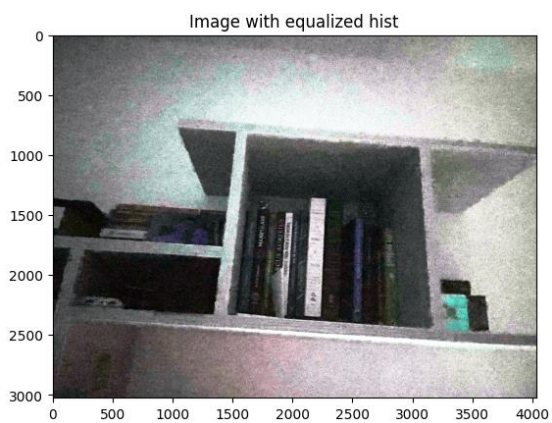
Tranformarea liniară Gray



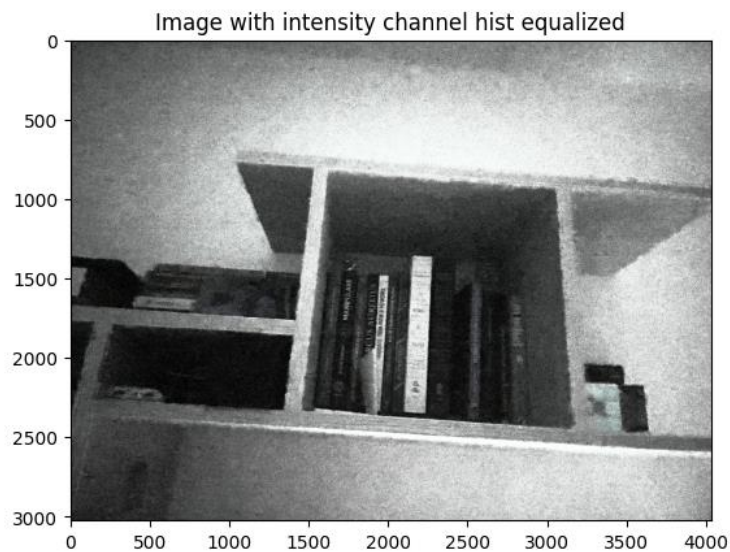
Transformarea logaritmică Gray



Egalizarea Histogramei pe canale RGB



Egalizarea Histogramei pe canale de intensitate (YCrCb)



6. Concluzii

Proiectul a explorat două abordări principale pentru îmbunătățirea imaginilor cu iluminare redusă: Transformări în Tonuri de Gri și Egalizarea Histogramelor, pe canalele RGB și în spațiul YCrCb.

În urma rezultatelor experimentale, s-a constatat că transformările liniare nu au furnizat rezultatele așteptate în contextul imaginilor cu iluminare redusă. În schimb, transformările logaritmice și egalizarea histogramelor, aplicate pe canalele RGB și în spațiul YCrCb, au evidențiat o îmbunătățire considerabilă a contrastului și a detaliilor în zonele întunecate. Aceste metode au oferit rezultate superioare, contribuind semnificativ la vizibilitatea și claritatea imaginilor.

7. Bibliografie

- [1] [Wang, Wencheng, Xiaojin Wu, Xiaohui Yuan, and Zairui Gao . "An experiment -based review of low -light image enhancement methods . " Ieee Access 8 \(2020 \) : 87884 -87917](#)
- [2] H. Wang, Y. Zhang, and H. Shen, “Review of image enhancement algorithms,” (in Chinese), Chin. Opt., vol. 10, no. 4, pp. 438–448, 2017.
- [3] S. Park, K. Kim, S. Yu, and J. Paik, “Contrast enhancement for low-light image enhancement: A survey,” IEIE Trans. Smart Process. Comput., vol. 7, no. 1, pp. 36–48, Feb. 2018.
- [4] Image Processing — Gray Level Transformation: Part 5. Internet: <https://kavinithisara.medium.com/image-processing-gray-level-transformation-part-5-549b9b282ad3>
- [5] Log transformation of an image using Python and OpenCV. Internet: <https://www.geeksforgeeks.org/log-transformation-of-an-image-using-python-and-opencv/>