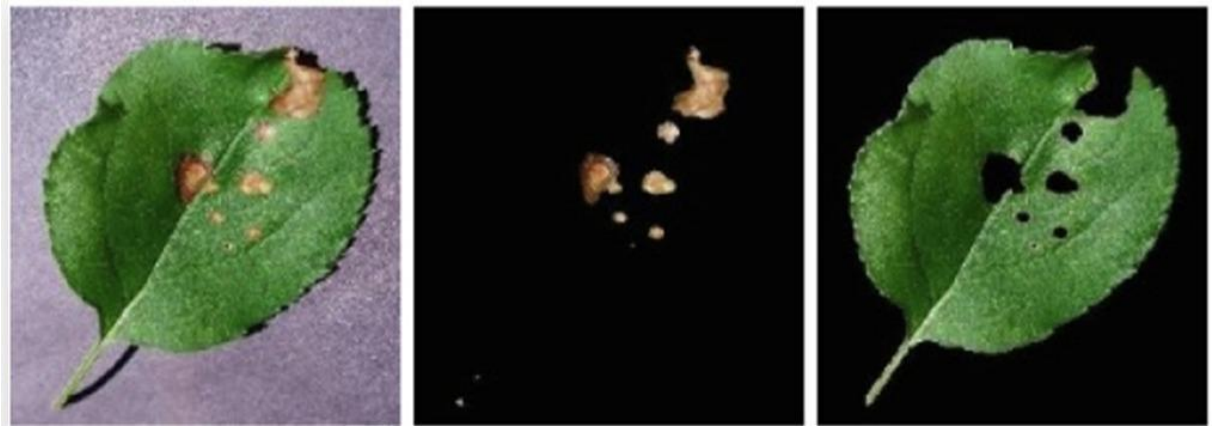
Detectarea bolii frunzelor



Membrii echipei:

- Dura Georgiana-Emanuela
- Rus Rebeca-Emanuela

Profesor coordonator:

- Florea Camelia Costina

Cuprins

Abstract 3
Introducere 4
Fundamentare teoretică 5
Segmentarea imaginii 5
Histograma de culoare 6
Implementare 7
Rezultate experimentale 18
Concluzie 23
Bibliografie 24

Abstract

Acest proiect este realizat în Python și abordează tema detectării bolii frunzelor, mai exact detectarea zonelor, mai exact a petelor colorate de pe frunză care indică existența unei boli.

Acest proiect este realizat în spațiul de culoare L*a*b* și în spațiul HSV.

În realizarea acestui proiect s-a utilizat procesul de segmentare a imaginilor, cu ajutorul unor măști, dorind a se obține imagini diferite, rezultate din imaginea originală, în care să se evidențieze zonele de interes: partea ce indică faptul că frunza are o boală, partea sănătoasă a frunzei, respectiv fundalul imaginii cu frunză.

Pentru a obține rezultatele dorite, am testat algoritmul pe mai multe imagini de intrare cu diverse frunze și am determinat valorile necesare generării măștilor utilizate în procesul de segmentare a imaginii, atât în spațiul L*a*b*, cât și în spațiul HSV.

Datele de intrare sunt reprezentate de imaginea cu o frunză care are pete ce indică o boală, iar ca rezultate se obțin următoarele imagini: imaginea originală care a fost citită, o imagine în care este evidențiată partea bolnavă a frunzei de intrare (petele colorate), o altă imagine în care este prezentată doar partea sănătoasă a frunzei (fără fundal și fără partea bolnavă) și o imagine în care se regăsește doar fundalul extras din imaginea de intrare. Asupra acestor imagini obținute se aplică funcția „scatter”, iar prin concatenarea orizontală a acestor histograme rezultă o imagine ce conține toate cele 3 histograme concatenate.

Introducere

Detectarea bolii frunzelor este un proces foarte important în domeniul agriculturii, deoarece prin identificarea existenței unei boli prin intermediul imaginilor cu frunze se pot efectua la timp tratamente de combatere a bolilor respective. [Journal Hindawi2022]

Conceptul de detecție a bolilor prin intermediul imaginilor cu frunze a fost puternic dezvoltat în ultima perioadă și se dorește a fi mai puternic dezvoltat cu scopul de a îmbunătăți agricultura din toată lumea. [Journal Hindawi2022]

Algoritmul implementat în cadrul acestui proiect se bazează pe prelucrare de imagini, mai exact pe segmentarea imaginilor în zone de interes: zonele bolnave, respectiv partea sănătoasă a frunzei și fundalul.

Segmentarea imaginilor este un proces des folosit în prelucrarea imaginilor, ce reprezintă împărțirea imaginii în zone de interes și urmărește extragerea, identificarea sau recunoașterea unui anumit obiect dintr-o imagine. Există diverse metode de segmentare a imaginilor. O posibilă clasificare ar fi:

- metode de segmentare orientate pe regiuni
- metode de segmentare orientate pe contururi. [PIL10]

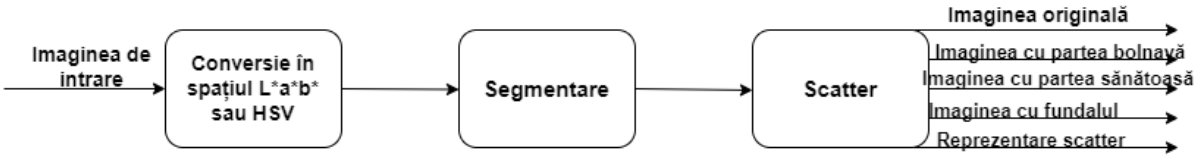
Acest proiect se bazează pe metode de segmentare orientate pe regiuni, pentru a segmenta imaginea originală în imagini distincte ce urmăresc zonele de interes: partea sau părțile bolnave ale frunzei, partea sănătoasă a frunzei, respectiv partea de fundal. În general, operația de segmentare orientată pe regiuni urmărește extragerea din imagine a zonelor (regiunilor) ocupate de diferite obiecte prezente în scenă, mai exact a unui set de parametri ale căror valori nu se modifică în diferitele puncte ce aparțin regiunii considerate.[Vertan1999]

În cadrul acestui proiect am lucrat în spațiul L*a*b* și în spațiul HSV. Spațiul L*a*b* are 3 canale de culoare: L*, care reprezintă luminanța, a* reprezentând coordonata roșu/verde și b*, reprezentând coordonata galben/albastru. [DEX-TEX2010]

Similar, spațiul HSV are alte 3 canale de culoare: H, care reprezintă nuanța; S, care reprezintă saturația și V, care reprezintă valoarea (corespondenta luminozității din spațiul L*a*b*). Spațiul HSV reprezintă mai bine modul în care oamenii se referă la culori decât spațiul de culoare RGB, astfel abordarea HSV este disponibilă în multe programe software de editare a imaginilor „high-end”.[eYewated]

Fundamentare teoretică

Algoritmul utilizat este realizat în două spații de culoare distincte: L*a*b*, respectiv HSV și presupune segmentarea imaginii de intrare, utilizând măști, în imagini distincte ce conțin zonele de interes ale frunzei, și anume: partea bolnavă, partea sănătoasă și fundalul. Asupra imaginilor rezultate se aplică funcția „scatter”.



Segmentarea imaginii

Spațiul L*a*b*

Pentru început, convertim imaginea din format RGB: I_{RGB} într-o imagine în spațiul L*a*b*: I_{Lab}, pe care o descompunem în 3 canale de culoare: I_L, I_a și I_b. [Journal Hasan2022] L* reprezintă canalul de luminanță (L = 0 reprezintă negru și poate lua valori până la 100, care reprezintă alb), a* reprezintă culoarea pe axa roșie-verde (cu „+a*” se indică roșu, respectiv cu „-a” se indică verde), iar b* reprezintă culoarea pe axa galben-albastru („+b*” indică galben și „-b*” indică albastru). [DEX-TEX2010]

În cadrul proiectului am folosit numai canalele I_a și I_b, pentru care am determinat, prin încercări multiple pe diverse imagini, și am setat un prag minim, respectiv un prag maxim pentru fiecare din cele 2 canale utilizate pentru detecția zonei sănătoase. În continuare, am creat o mască folosind pragurile determinate, iar în final am aplicat masca obținută pe imaginea originală, folosind operația „și” logic pe biți. [Journal Hasan2022] Acest proces se repetă pentru zona bolnavă, respectiv pentru fundal.

Spațiul HSV

Similar lucrului în spațiul L*a*b*, convertim imaginea din format RGB: I_{RGB} într-o imagine în spațiul HSV: I_{HSV}, pe care o descompunem în 3 canale de culoare: I_H, I_S și I_V. Canalul H reprezintă nuanța și este exprimată ca un număr de la 0 la 360 de grade, S reprezintă saturația (cantitatea de culoare gri) și poate lua valori de la 0 la 100% și V, care reprezintă valoarea (corespondenta luminozității din spațiul L*a*b*). Valoarea funcționează împreună cu saturația și descrie luminozitatea sau intensitatea culorii, de la 0-100%, unde 0 este complet negru, iar 100 este cea mai strălucitoare și reprezintă cea mai mare culoare.[eYewated]

În acest caz, am utilizat doar canalele de culoare: I_H și I_S, pentru care am determinat, prin încercări multiple pe diverse imagini, și am setat un prag minim, respectiv un prag maxim. Cu ajutorul acestor limite, am creat o mască, apoi am inițializat o altă mască cu

valori nule, de dimensiunea primei măști, a căror valori le modificăm în funcție de valorile zonei parcurse, urmând să se aplice masca rezultată pe imaginea originală, folosind operația „și” logic pe biți, pentru a obține imaginea cu zona sănătoasă. Acest proces se repetă pentru zona bolnavă, respectiv pentru fundal.

Histograma de culoare

Histograma imaginii, numită și funcția de densitate de probabilitate a variabilei aleatoare discrete, reflectă distribuția în imagine a proprietății fizice înregistrate, mai exact a unei componente (sau a unui canal) din spațiul de culoare utilizat. O altă definiție a histogramei ar fi probabilitatea de apariție în imagine a diferitelor nivele de gri posibile.[Vertan1999]

Spațiul L*a*b*

Deoarece imaginile obținute cu zonele de interes rezultate în urma segmentării imaginii de intrare sunt în spațiul RGB, convertim iar imaginile din spațiul RGB în spațiul L*a*b*: I_{Lab}. Imaginea nou rezultată este descompusă în cele 2 canale de culoare utilizate: I_a, respectiv I_b, urmând să calculăm „scatter-ul” folosind canalele a*, respectiv b*. Acest proces este utilizat pentru toate cele 3 imagini ce conțin zonele de interes segmentate și, în final, cele 3 caracteristici ale imaginilor segmentate sunt concatenate orizontal. Rezultatul obținut este caracteristic histogramei de culoare corespunzător celor 3 imagini segmentate I_{RGB}.[Journal Hasan2022]

Spațiul HSV

În cazul utilizării spațiului HSV, „scatter-ul” l-am reprezentat tot utilizând spațiul L*a*b*, deci modul de lucru este identic cu cel prezentat mai sus.

Implementare

Pentru implementarea algoritmului am utilizat limbajul Python, prin intermediul căruia am segmentat imaginea de intrare în alte 3 imagini distincte, ce surprind: partea bolnavă a frunzei, partea sănătoasă, respectiv fundalul. În cadrul acestui proiect am lucrat în două spații de culoare distincte, și anume: spațiul L*a*b*, respectiv spațiul HSV.

Pași parcurși în implementarea programului:

- Importare biblioteci:
  - ***cv2*** -> citirea imaginii cu ajutorul metodei „imread”
  - ***numpy*** -> pentru lucrul cu vectori
  - ***matplotlib.pyplot*** -> pentru reprezentarea imaginii [PNI2023]
- Citirea imaginii din Google Drive

```
# Citirea imaginii RGB
image_rgb = cv2.imread('/content/drive/MyDrive/Proiect PNI/frunza.jpg')
```

Implementarea în spațiul L*a*b*

- **Conversia imaginii citite în spațiul de culoare L*a*b*** [PNI2023]

```
# Convertirea imaginii în spațiul de culoare LAB
image_lab = cv2.cvtColor(image_rgb, cv2.COLOR_BGR2LAB)
```

- Împărțirea imaginii în cele 3 canale: L, a și b

```
# Extrage canalele L, a și b
channel_l, channel_a, channel_b = cv2.split(image_lab)
```

- Afișarea imaginii originale, a imaginii convertite în spațiul de culoare L*a*b* și a celor 3 canale: L, a și b [PNI2023]

  -eliminarea valorilor numerice de pe axe:

```
plt.axis('off')
```

- Calcul histograme corespunzătoare fiecărui canal [PNI2023]

```
# Calculează histogramele pentru fiecare canal
hist_l = cv2.calcHist([channel_l], [0], None, [256], [0, 256])
hist_a = cv2.calcHist([channel_a], [0], None, [256], [0, 256])
hist_b = cv2.calcHist([channel_b], [0], None, [256], [0, 256])
```

- Afișare histograme obținute [PNI2023]

```
# Afișează histogramele
plt.figure(figsize=(10, 5))

plt.subplot(1, 3, 1)
plt.plot(hist_l, color='gray')
plt.title('Histograma pentru canalul L')

plt.subplot(1, 3, 2)
plt.plot(hist_a, color='g')
plt.title('Histograma pentru canalul a')

plt.subplot(1, 3, 3)
plt.plot(hist_b, color='b')
plt.title('Histograma pentru canalul b')

plt.show()
```

- **Segmentarea imaginii pe zone de interes**

1. **Partea sănătoasă**

   - Setarea pragului inferior, respectiv a pragului superior pentru componenta a* [Vertan1999]

```
# Definirea pragurilor pentru componenta a
a_lower, a_upper = 70, 125
```

   - Crearea măștii folosind cele 2 componente de culoare: a* și b*

```
# Crearea măștii folosind componentele a și b
mask= cv2.inRange(image_lab, (0, a_lower, 140), (255, a_upper, 255))
```

   - Aplicarea măștii create pe imaginea de intrare, folosind operația „și” logic pe biți[Journal Hindawi2022]

```
# Aplicarea măștii pe imagine
result_LABs = cv2.bitwise_and(image_rgb, image_rgb, mask=mask)
```

   - Afișarea imaginii originale și a imaginii ce evidențiază zona sănătoasă[PNI2023]

```
# Afișarea imaginii și a rezultatului
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image_rgb, cv2.COLOR_BGR2RGB))
plt.title('Imagine originală')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(result1, cv2.COLOR_BGR2RGB))
plt.title('regiunea sănătoasă')

plt.axis('off')
plt.show()
```

2. **Partea bolnavă (zonele bolnave)**

- Setarea pragurilor inferioare, respectiv superioare pentru componentele a* și b* [Vertan1999]

```
# Definirea pragurilor pentru componentele a și b
a_lower, a_upper = 125, 160 # Puteți ajusta aceste
b_lower, b_upper = 140,190  # Puteți ajusta aceste
```

- Crearea măștii folosind cele 2 componente de culoare: a* și b*

```
# Crearea măștii folosind componentele de culoare a și b
mask = cv2.inRange(image_lab, (0, a_lower, b_lower), (255, a_upper,  b_upper))
```

- Aplicarea măștii create pe imaginea de intrare, folosind operația „și” logic pe biți[Journal Hindawi2022]

```
# Aplicarea măștii pe imagine
result_LABb = cv2.bitwise_and(image_rgb, image_rgb, mask=mask)
```

- Afișarea imaginii originale și a imaginii ce evidențiază zona bolnavă[PNI2023]

```
# Afișarea imaginii și a rezultatului
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image_rgb, cv2.COLOR_BGR2RGB))
plt.title('Imagine originală')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(result_LABb, cv2.COLOR_BGR2RGB))
plt.title('regiunea bolnava')
plt.axis('off')

plt.show()
```

3. **Fundalul**

- Setarea pragurilor inferioare, respectiv superioare pentru componentele a* și b* [Vertan1999]

```
# Definirea pragurilor pentru componentele a și b
a_lower, a_upper = 100, 160 # Puteți ajusta aceste
b_lower, b_upper =90,140 # Puteți ajusta aceste val
```

- Crearea măștii folosind cele 2 componente de culoare: a* și b*

```
# Crearea măștii folosind componentele de culoare a și b
mask = cv2.inRange(image_lab, (0, a_lower, b_lower), (255, a_upper,  b_upper))
```

- Aplicarea măștii create pe imaginea de intrare, folosind operația „și” logic pe biți[Journal Hindawi2022]

```
# Aplicarea măștii pe imagine
result_LABf = cv2.bitwise_and(image_rgb, image_rgb, mask=mask)
```

- Afișarea imaginii originale și a imaginii ce evidențiază zona de fundal[PNI2023]

```
# Afișarea imaginii și a rezultatului
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image_rgb, cv2.COLOR_BGR2RGB))
plt.title('Imagine originală')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(result_LABf, cv2.COLOR_BGR2RGB))
plt.title('Evidențierea zonei de fundal')
plt.axis('off')

plt.show()
```

- Afișarea imaginii originale și a imaginilor rezultate în urma segmentării: a zonei bolnave, a zonei sănătoase, respectiv a fundalului, obținute lucrând în spațiul L*a*b*[PNI2023]

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 4, 1)
plt.imshow(cv2.cvtColor(image_rgb, cv2.COLOR_BGR2RGB))
plt.title('Imaginea Originală')
plt.axis('off')

plt.subplot(1, 4, 2)
plt.imshow(cv2.cvtColor(result_LABb, cv2.COLOR_BGR2RGB))
plt.title('regiunea bolnavă')
plt.axis('off')

plt.subplot(1, 4 , 3)
plt.imshow(cv2.cvtColor(result_LABs, cv2.COLOR_BGR2RGB))
plt.title('regiunea sănătoasă')
plt.axis('off')

plt.subplot(1, 4, 4)
plt.imshow(cv2.cvtColor(result_LABf, cv2.COLOR_BGR2RGB))
plt.title('regiunea de fundal')
plt.axis('off')
plt.show()
```

- **Generarea histogramelor de culoare (a „scatter-ului”) pentru imaginile cu cele 3 zone de interes**
  - Conversia imaginii cu zona bolnavă în spațiul de culoare L*a*b*[PNI2023]

```
# Converteste imaginea in spatiul de culoare LAB
lab_image_disease = cv2.cvtColor(result_LABb, cv2.COLOR_BGR2LAB)
```

  - Extragerea valorilor pentru componentele a* și b* necesare funcției „scatter” [PNI2023]

```
# Extrage valorile pentru a și b
a_values = lab_image_disease[:, :, 1].ravel()  # Componenta a
b_values = lab_image_disease[:, :, 2].ravel()  # Componenta b
```

  - Crearea și afișarea „scatter plot-ului” [Journal Hindawi2022]

```
# Crearea scatter plot-ului
plt.figure(figsize=(8, 8))
plt.scatter( a_values, b_values, c='y', marker='o', alpha=0.5)
plt.title('Scatter Plot în spațiul LAB pentru regiunea bolnava')
plt.xlabel('Componenta a')
plt.ylabel('Componenta b')
plt.grid(True)
plt.show()
```

  - Etapele enunțate mai sus pentru imaginea cu zona bolnavă se repetă pentru imaginea cu zona sănătoasă, și respectiv pentru imaginea cu fundalul
- **Concatenarea pe orizontală a celor 3 „scatter plot-uri” obținute anterior și afișarea rezultatului**[Journal Hindawi2022]

```
# Crearea scatter plot-ului
plt.figure(figsize=(8, 8))
plt.scatter(a_values, b_values, c='y', marker='o', alpha=0.5)
plt.scatter(a_values1, b_values1, c='g', marker='*', alpha=0.5)
plt.scatter(a_values2, b_values2, c='gray', marker='+', alpha=0.5)
plt.title('Scatter Plot în spațiul LAB final')
plt.xlabel('Componenta a')
plt.ylabel('Componenta b')
plt.grid(True)
plt.show()
```

Implementarea în spațiul HSV

- **Conversia imaginii citite în spațiul de culoare HSV**[PNI2023]

```
#convertim imaginea din rgb in hsv
image_hsv = cv2.cvtColor(image_rgb, cv2.COLOR_BGR2HSV)
```

- Împărțirea imaginii în cele 3 canale: H, S și V

```
#extragem canalele H S si V

channel_h, channel_s, channel_v = cv2.split(image_hsv)
```

- Afișarea imaginii originale, a imaginii convertite în spațiul de culoare HSV și a celor 3 canale: H, S și V[PNI2023]
- Calcul histograme corespunzătoare fiecărui canal[PNI2023]

```
# Calculează histogramele pentru fiecare canal
hist_h = cv2.calcHist([channel_h], [0], None, [256], [0, 256])
hist_s = cv2.calcHist([channel_s], [0], None, [256], [0, 256])
hist_v = cv2.calcHist([channel_v], [0], None, [256], [0, 256])
```

- Afișare histograme obținute[PNI2023]

```
# Afișează histogramele
plt.figure(figsize=(10, 5))

plt.subplot(1, 3, 1)
plt.plot(hist_h, color='gray')
plt.title('Histograma pentru canalul H')

plt.subplot(1, 3, 2)
plt.plot(hist_s, color='g')
plt.title('Histograma pentru canalul S')

plt.subplot(1, 3, 3)
plt.plot(hist_v, color='b')
plt.title('Histograma pentru canalul V')

plt.show()
```

- **Segmentarea imaginii pe zone de interes**
  1. **Partea sănătoasă**
     - Setarea pragurilor pentru componentele H și S [Vertan1999]

```
# Defineste limitele culorilor pentru c
lower_bound = np.array([30, 60, 0])  #
upper_bound = np.array([90, 255, 255])
```

     - Creare mască folosind limitele definite anterior

```
# Creează o mască binară folosind limitele culorilor
mask = cv2.inRange(hsv, lower_bound, upper_bound)
```

     - Identificare regiuni conectate

```
# Identifică regiunile conectate (componente conectate)
num_labels, labels, stats, centroids = cv2.connectedComponentsWithStats(mask, connectivity=8)
```

     - Inițializare mască cu valori nule

```
# Inițializează o mască goală pentru
disease_mask = np.zeros_like(mask)
```

- Parcurgere regiuni și adăugare valori în masca nulă [Vertan1999]

```
# Parcurge regiunile și le adaugă la mască
for i in range(1, num_labels):  # începe de la 1 pentru a exclude fun
    area = stats[i, cv2.CC_STAT_AREA]
    # Ajustează acest prag pentru a exclude regiunile mici, care ar p
    if area > 1000:
        region_mask = np.where(labels == i, 255, 0).astype(np.uint8)
        disease_mask = cv2.bitwise_or(disease_mask, region_mask)
```

- Aplicarea măștii obținute pe imaginea de intrare, folosind operația „și” logic pe biți[Journal Hindawi2022]

```
# Aplică mască pe imaginea originală pentru a evidenția zona sănătoasă
result2 = cv2.bitwise_and(image_rgb, image_rgb, mask=disease_mask)
```

- Afișarea imaginii originale și a imaginii ce evidențiază zona sănătoasă[PNI2023]

```
# Afișează imaginea originală și rezultatul
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image_rgb, cv2.COLOR_BGR2RGB))
plt.title('Imaginea Originală')

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(result2, cv2.COLOR_BGR2RGB))
plt.title('Regiunea sănătoasă')

plt.show()
```

2. **Partea bolnavă (zonele bolnave)**

- Setarea pragurilor pentru componentele H și S [Vertan1999]

```
lower_bound = np.array([10, 90, 0])
upper_bound = np.array([30, 255, 255])
```

- Creare mască folosind limitele definite anterior

```
mask = cv2.inRange(hsv, lower_bound, upper_bound)
```

- Creare mască Gaussiană și aplicare funcție „threshold”

```
blurred = cv2.GaussianBlur(mask, (5, 5), 0)
_, thresholded = cv2.threshold(blurred, 50, 255, cv2.THRESH_BINARY)
```

- Identificare regiuni conectate

```
#Identifică regiunile conectate (componente conectate)

num_labels, labels, stats, centroids = cv2.connectedComponentsWithStats(thresholded, connectivity=8)
```

- Inițializare mască cu valori nule

```
#Inițializează o mască goală pentru a adăuga

disease_mask = np.zeros_like(thresholded)
```

- Parcurgere regiuni și adăugare valori în masca nulă [Vertan1999]

```
#Parcurge regiunile și le adaugă la mască

min_area_threshold =30
for i in range(1, num_labels):
    area = stats[i, cv2.CC_STAT_AREA]
    if area > min_area_threshold:
        region_mask = np.where(labels == i, 255, 0).astype(np.uint8)
        disease_mask = cv2.bitwise_or(disease_mask, region_mask)
```

- Aplicarea măștii obținute pe imaginea de intrare, folosind operația „și” logic pe biți[Journal Hindawi2022]

```
# Extract diseased region
disease_region = cv2.bitwise_and(image_rgb, image_rgb, mask=disease_mask)
```

- Afișarea imaginii originale și a imaginii ce evidențiază zona bolnavă[PNI2023]

```
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image_rgb, cv2.COLOR_BGR2RGB))
plt.title('Imagine originală')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(disease_region, cv2.COLOR_BGR2RGB))
plt.title('regiunea bolnavă')
plt.axis('off')
plt.show()
```

3. **Fundalul**
   - Setarea pragurilor pentru componentele H și S [Vertan1999]

```
lower_bound = np.array([0, 40, 40])
upper_bound = np.array([70, 255, 255])
```

   - Creare mască folosind limitele definite anterior

```
mask = cv2.inRange(hsv, lower_bound, upper_bound)
```

   - Identificare regiuni conectate

```
#Identifică regiunile conectate (componente conectate)

num_labels, labels, stats, centroids = cv2.connectedComponentsWithStats(mask, connectivity=8)
```

   - Inițializare mască cu valori nule

```
#creaza o masca goala
healthy_mask = np.zeros_like(mask)
```

- Parcurgere regiuni și adăugare valori în masca nulă [Vertan1999]

```
#Parcurge regiunile și le adaugă la mască

for i in range(1, num_labels):
    area = stats[i, cv2.CC_STAT_AREA]
    if area > 1000:
        region_mask = np.where(labels == i, 255, 0).astype(np.uint8)
        healthy_mask = cv2.bitwise_or(healthy_mask, region_mask)
```

- Evidențiere zonă bolnavă prin negarea măștii pentru regiunea sănătoasă

```
# Evidențiem zona bolnavă prin negarea măștii sănătoase
disease_mask = cv2.bitwise_not(healthy_mask)
```

- Aplicarea măștii obținute pe imaginea de intrare, folosind operația „și” logic pe biți[Journal Hindawi2022]

```
result = cv2.bitwise_and(image_rgb, image_rgb, mask=disease_mask)
```

- Afișarea imaginii originale și a imaginii ce evidențiază zona sănătoasă[PNI2023]

```
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image_rgb, cv2.COLOR_BGR2RGB))
plt.title('Imaginea Originală')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(result, cv2.COLOR_BGR2RGB))
plt.title('regiunea de fundal')
plt.axis('off')
plt.show()
```

- Afișarea imaginii originale și a imaginilor rezultate în urma segmentării: a zonei bolnave, a zonei sănătoase, respectiv a fundalului, obținute lucrând în spațiul HSV[PNI2023]

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 4, 1)
plt.imshow(cv2.cvtColor(image_rgb, cv2.COLOR_BGR2RGB))
plt.title('Imaginea Originală')
plt.axis('off')

plt.subplot(1, 4, 2)
plt.imshow(cv2.cvtColor(disease_region, cv2.COLOR_BGR2RGB))
plt.title('Regiunea bolnavă')
plt.axis('off')

plt.subplot(1, 4 , 3)
plt.imshow(cv2.cvtColor(result2, cv2.COLOR_BGR2RGB))
plt.title('Regiunea sănătoasă')
plt.axis('off')

plt.subplot(1, 4, 4)
plt.imshow(cv2.cvtColor(result, cv2.COLOR_BGR2RGB))
plt.title('Regiunea de fundal')
plt.axis('off')
plt.show()
```

- **Generarea histogramelor de culoare (a „scatter-ului") pentru imaginile cu cele 3 zone de interes**
  - Conversia imaginii cu zona bolnavă în spațiul de culoare HSV[PNI2023]

```
# Converteste imaginea in spatiul de culoare HSV
lab_image_disease = cv2.cvtColor(disease_region, cv2.COLOR_BGR2HSV)
```

  - Extragerea valorilor pentru componentele H și S necesare funcției „scatter" [PNI2023]

```
# Extrage valorile pentru H si S
h_values3 = lab_image_disease[:, :, 0].ravel()  # Componenta H
s_values3 = lab_image_disease[:, :, 1].ravel()  # Componenta S
```

❖ Crearea și afișarea „scatter plot-ului” [Journal Hindawi2022]

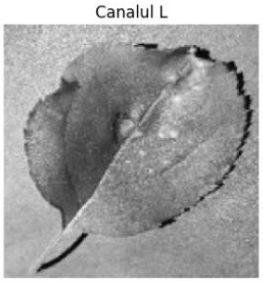
```
# Crearea scatter plot-ului
plt.figure(figsize=(8, 8))
plt.scatter( h_values3, s_values3, c='y', marker='o', alpha=0.5)
plt.title('Scatter Plot pentru regiunea bolnava')
plt.xlabel('Componenta H')
plt.ylabel('Componenta S')
plt.grid(True)
plt.show()
```

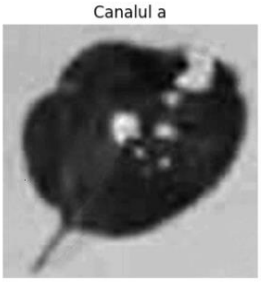
- Etapele enunțate mai sus pentru imaginea cu zona bolnavă se repetă pentru imaginea cu zona sănătoasă, și respectiv pentru imaginea cu fundalul.

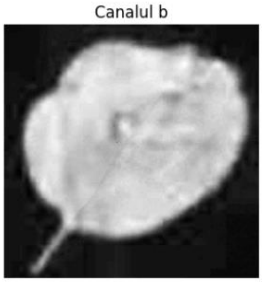
➢ **Concatenarea pe orizontală a celor 3 „scatter plot-uri” obținute anterior și afișarea rezultatului**[Journal Hindawi2022]

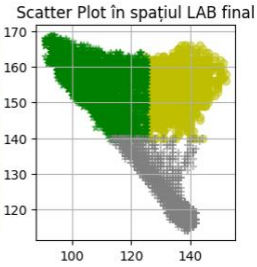
```
# Crearea scatter plot-ului
plt.figure(figsize=(8, 8))
plt.scatter(h_values3, s_values3, c='y', marker='o', alpha=0.5)
plt.scatter(h_values4, s_values4, c='g', marker='*', alpha=0.5)
plt.scatter(h_values5, s_values5, c='gray', marker='+', alpha=0.5)
plt.title('Scatter Plot în spațiul HSV final')
plt.xlabel('Componenta H')
plt.ylabel('Componenta S')
plt.grid(True)
plt.show()
```

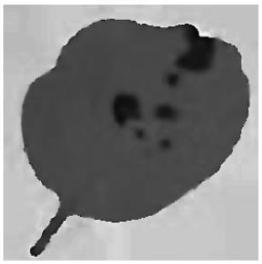
Rezultate experimentale

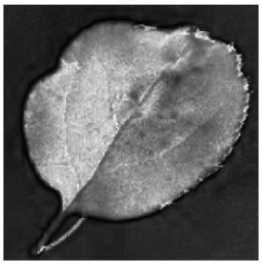


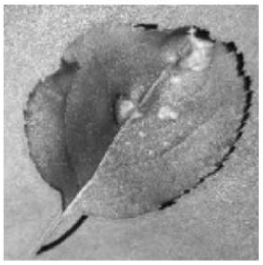


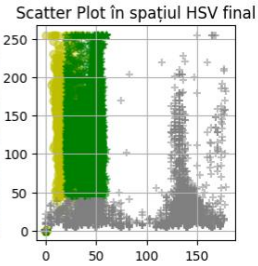


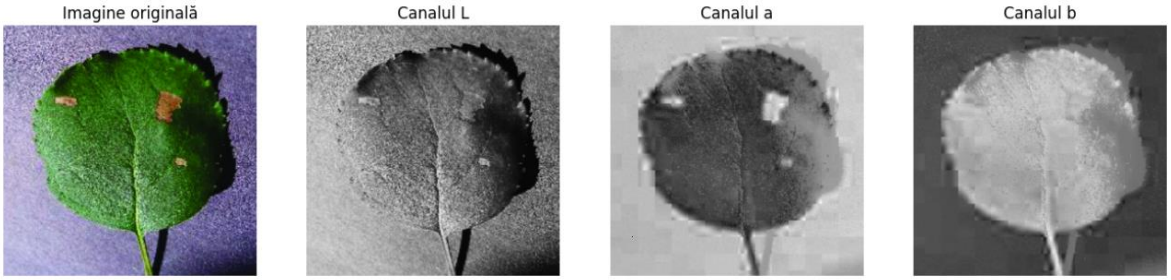




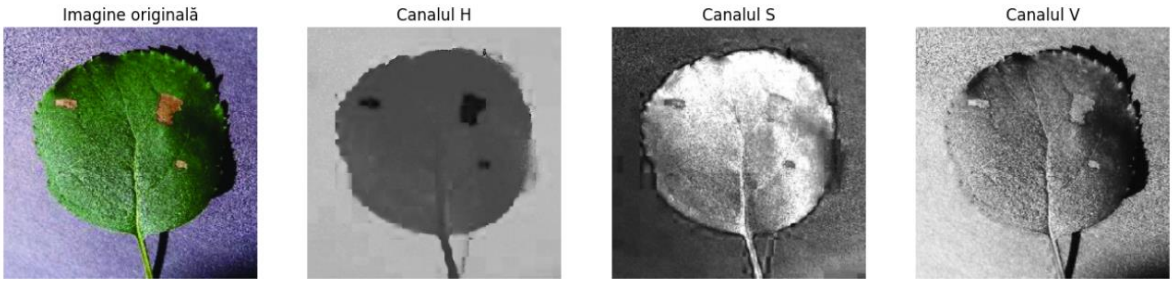






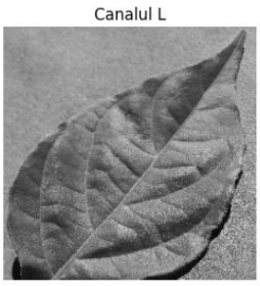

Imagine originală
Canalul L
Canalul a
Canalul b

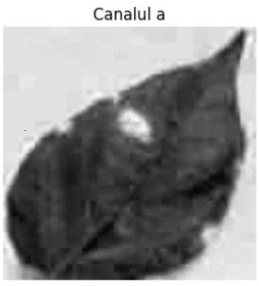

Imaginea Originală
regiunea bolnavă
regiunea sănătoasă
regiunea de fundal
Scatter Plot în spațiul LAB final
200
180
160
140
120
100
80
100
120
140
160

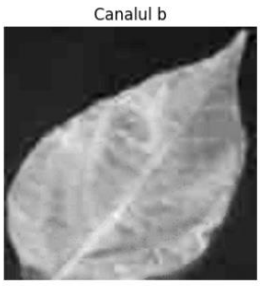

Imagine originală
Canalul H
Canalul S
Canalul V


Imaginea Originală
Regiunea bolnavă
Regiunea sănătoasă
Regiunea de fundal
Scatter Plot în spațiul HSV final
250
200
150
100
50
0
0
50
100
150


Imagine originală


Canalul L


Canalul a

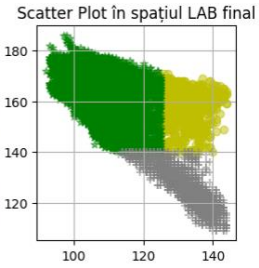

Canalul b


Imaginea Originală

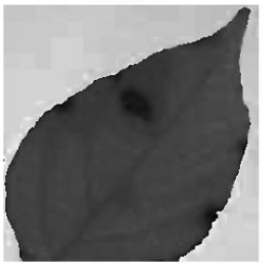

regiunea bolnavă


regiunea sănătoasă


regiunea de fundal


Scatter Plot în spațiul LAB final
180
160
140
120
100
120
140


Imagine originală


Canalul H


Canalul S

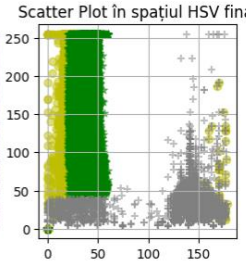

Canalul V


Imaginea Originală

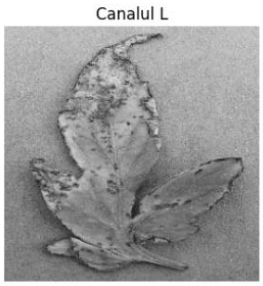

Regiunea bolnavă

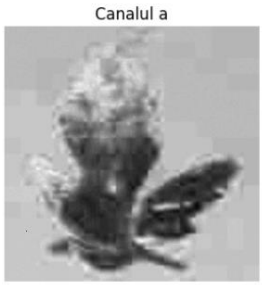

Regiunea sănătoasă

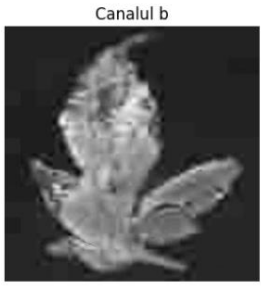

Regiunea de fundal


Scatter Plot în spațiul HSV final
250
200
150
100
50
0
0
50
100
150


Imagine originală


Canalul L


Canalul a

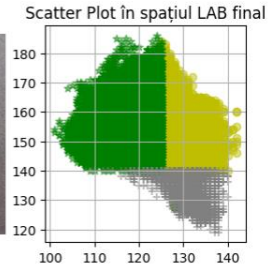

Canalul b


Imaginea Originală

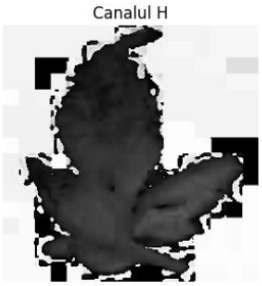

regiunea bolnavă

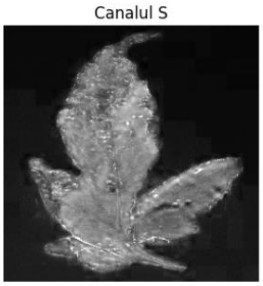

regiunea sănătoasă

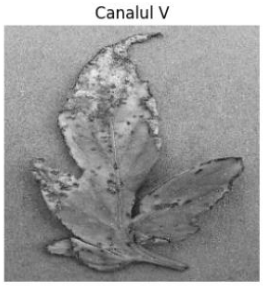

regiunea de fundal


Scatter Plot în spațiul LAB final
180
170
160
150
140
130
120
100
110
120
130
140


Imagine originală


Canalul H


Canalul S

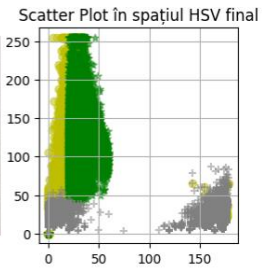

Canalul V

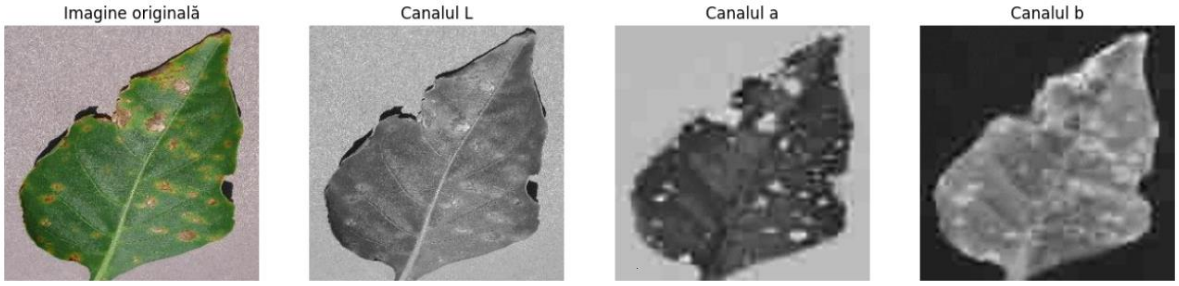

Imaginea Originală


Regiunea bolnavă

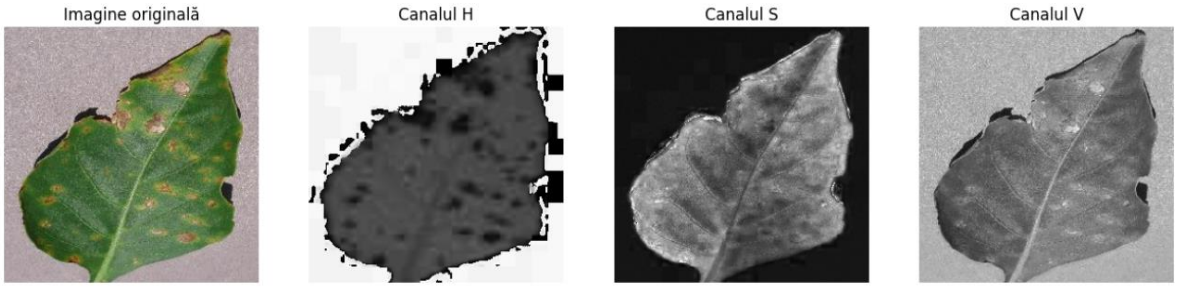

Regiunea sănătoasă


Regiunea de fundal


Scatter Plot în spațiul HSV final
250
200
150
100
50
0
0
50
100
150


Imagine originală
Canalul L
Canalul a
Canalul b


Imaginea Originală
regiunea bolnavă
regiunea sănătoasă
regiunea de fundal
Scatter Plot în spațiul LAB final
180
170
160
150
140
130
120
100
120
140


Imagine originală
Canalul H
Canalul S
Canalul V


Imaginea Originală
Regiunea bolnavă
Regiunea sănătoasă
Regiunea de fundal
Scatter Plot în spațiul HSV final
250
200
150
100
50
0
0
50
100
150

Concluzie

În concluzie, rezultatele sunt puțin diferite, în funcție de spațiul de culoare utilizat (L*a*b*, respectiv HSV). Această diferență constă în gradul de acuratețe al segmentării zonelor de interes din imaginea originală.

De asemenea, apar diferențe și în reprezentarea „scatter plot-ului”. În cazul utilizării spațiului de culoare HSV, gama de valori pentru componente H, S și V caracteristice imaginilor cu cele 3 zone de interes: zona bolnavă, zona sănătoasă și zona de fundal, nu este atât de largă ca și în cazul utilizării spațiului L*a*b*, de unde rezultă suprapuneri ale valorilor reprezentative zonei bolnave cu cele a zonei sănătoase și, respectiv, a fundalului, deci există suprapuneri ale valorilor caracteristice celor 3 imagini în reprezentarea „scatter plot-ului”.

Totodată, în funcție de spațiul de culoare utilizat, cele 3 canale reprezentative au semnificații diferite, ceea ce implică setarea de praguri distincte utilizate în segmentarea imaginii originale. Astfel, rezultatele obținute pot fi diferite în funcție de pragurile alese pentru fiecare spațiu de culoare.

Bibliografie

[Journal Hindawi2022] Ashutosh Kumar Singh , SVN Sreenivasu , U.S.B. K. Mahalaxmi , Himanshu Sharma , Dinesh D. Patil , and Evans Asenso, *Journal of Food Quality*, Volume 2022: https://downloads.hindawi.com/journals/jfq/2022/2845320.pdf

[PIL10] Lucrarea10, Segmentarea imaginilor, https://www.miv.ro/ro/documentatie/pi/PIlab10.pdf

[Vertan1999] Autor: Constantin VERTAN, *PRELUCRAREA ȘI ANALIZA IMAGINILOR, 1999* , pag. 110-129

[DEX-TEX2010] Ultima actualizare: 14 Mai 2010, https://www.dex-tex.info/dictionartextil/id.Diferen%C5%A3a+de+culoare+CIE+L%2Aa%2Ab%2A/i.html

[eYewated] https://ro.eyewated.com/care-este-modelul-de-culoare-hsv/ , Jacci Howard Bear

[Journal Hasan2022] Sharad Hasan, Sarwar Jahan , Md. Imdadul Islam, *Journal of King Saud University – Computer and Information Sciences*, 8 July 2022

[PNI2023] Laborator 1, Prelucrarea numerică a imaginilor, 2023