

Mateusz Dec

Projekt Zaliczeniowy

Celem projektu było zaprojektować a następnie zaimplementować sekwencję operacji przetwarzania obrazów umożliwiającą wyznaczenie liczby obiektów (ziaren) znajdujących się na obrazie wejściowym.

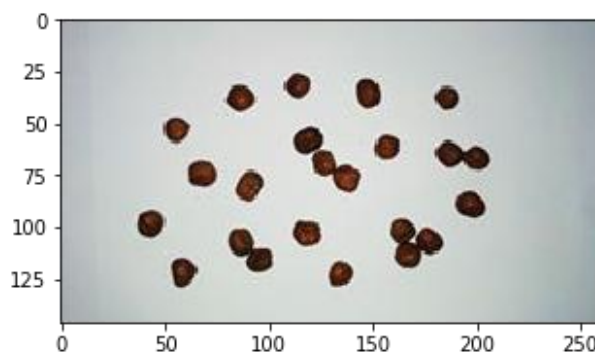
Zdjęcia zostały wykonane smartfonem (Xiaomi Redmi4X). Algorytm został sprawdzony na 15 zdjęciach z różną ilością obiektów oraz w różnej odległości od siebie.

Poprawność algorytmu oceniam na 100%, przeprowadzony algorytm na wszystkich zdjęciach działa poprawnie i nie wykrył żadnych błędów.

Etapy działania algorytmu:

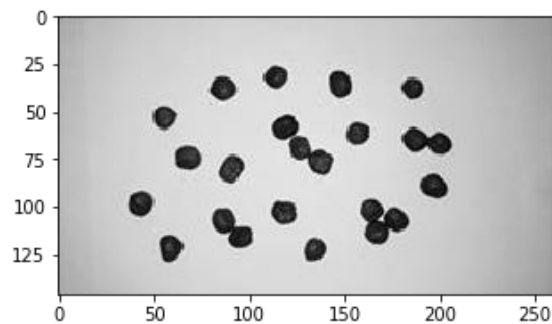
1. Pierwszym krokiem było wczytanie obrazu wejściowego za pomocą funkcji **imread** z biblioteki skimage modułu io.

Obraz wejściowy



- Następnie obraz wejściowy za pomocą funkcji `rgb2grey` z biblioteki `skimage` z modułu `color`, jest konwertowany na obraz wieloodcieniowy w skali szarości.

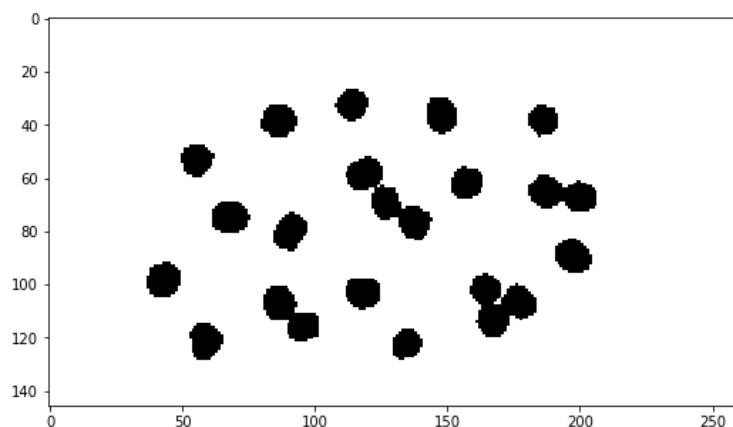
Analizowany obraz



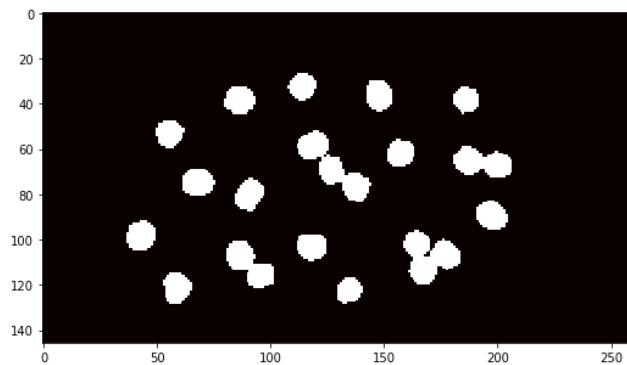
- Kolejnym etapem jest wykonanie binaryzacji obrazu wykorzystując próg wyliczeniowy za pomocą funkcji `threshold_otsu` z biblioteki `filters`.

Prog binaryzacji wynosi: 0.4843349425551471

Obraz po binaryzacji



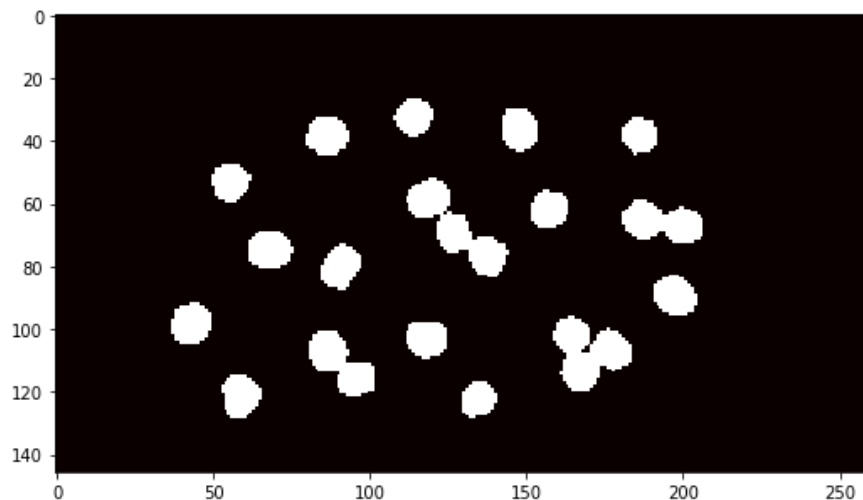
4. Kolejnym krokiem jest oddzielenie ziaren od tła oraz obliczenie jaki % obiekty zajmują obszar na zdjęciu.



```
def count_black_white(image):  
    a,b = image.shape  
    black = 0  
    white = 0  
  
    for i in range(a):  
        for j in range(b):  
            if image[i,j] == 1:  
                white+=1  
            elif image[i,j] == 0:  
                black +=1  
    return (white*100)/(white+black)
```

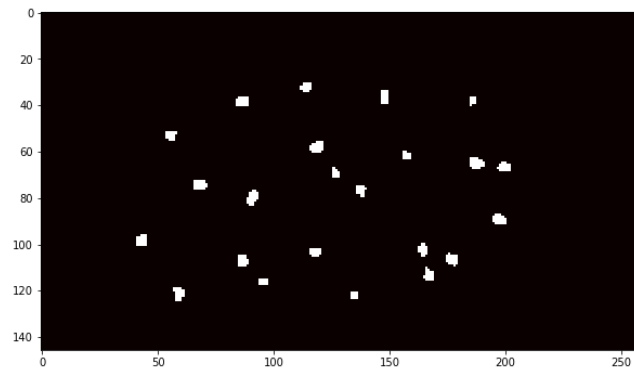
1-Funkcja obliczająca % obiektów na obrazie

*1- Działanie funkcji polega na przechodzeniu po całym obrazie pixel po pixelu zliczanie punktów białych i czarnych a następnie zwraca ilość białych punktów pomnożona przez 100 i dzielenie przez sumę punktów białych i czarnych.

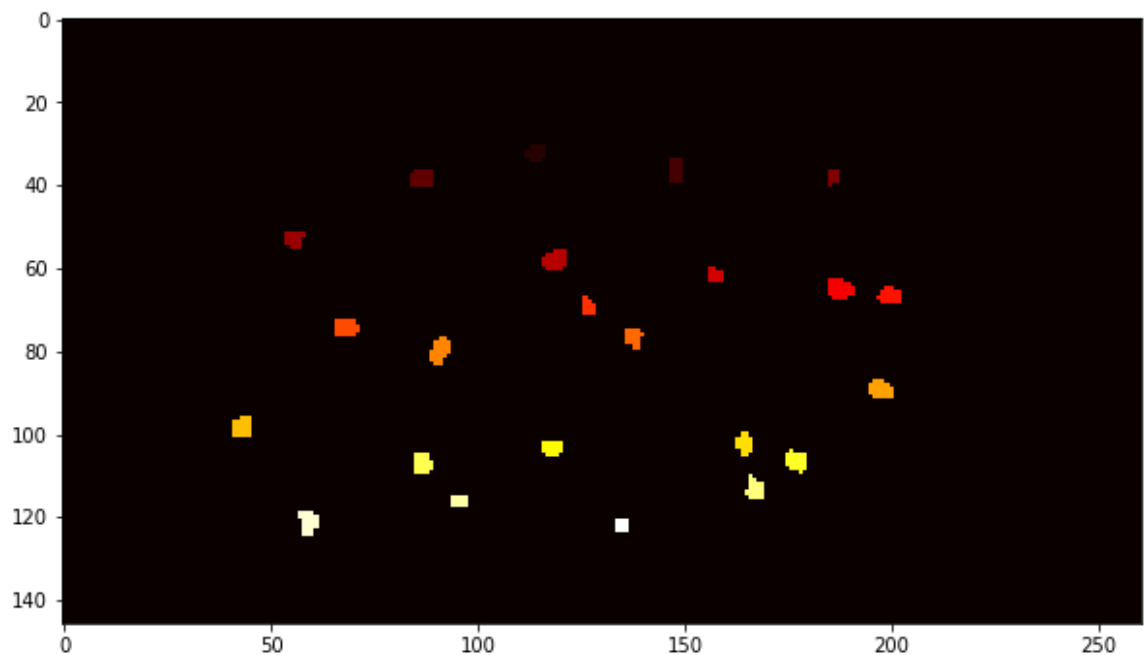


Analizowane obiekty stanowią 7.526373799401669 % obszaru zdjęcia

5. Kolejnym etapem jest wykonanie 4 erozji na kopi obrazu , aby oddzielić sklejone ziarna od siebie, erozja wykonywana jest za pomocą funkcji **binary_erosion** z pakietu **morphology**.



6. Następnie wykonywana jest indeksacja obrazu za pomocą **ndimage** z biblioteki **scipy** oraz obliczenie ile jest obiektów na obrazie za pomocą **ndi.label**



Ilość obiektów: 23

7. Ostatnim etapem jest obliczanie współczynników:

1) Liczby punktów na każdym obiekcie:

```
def Figury_punkty(obiekt, numer):  
  
    punkt = []  
    for i in range(obiekt.shape[0]):  
        for j in range(obiekt.shape[1]):  
            if obiekt[i,j] == numer:  
                punkt.append((i,j))  
  
    return punkt
```

2) Środki ciężkości obiektów:

```
def srodek_ciezkosci(punkt):  
    a=0  
    b=0  
    for (i,j) in punkt:  
        a = a + j  
        b = b + i  
    a = a/len(punkt)  
    b = b/len(punkt)  
  
    return [b, a]
```

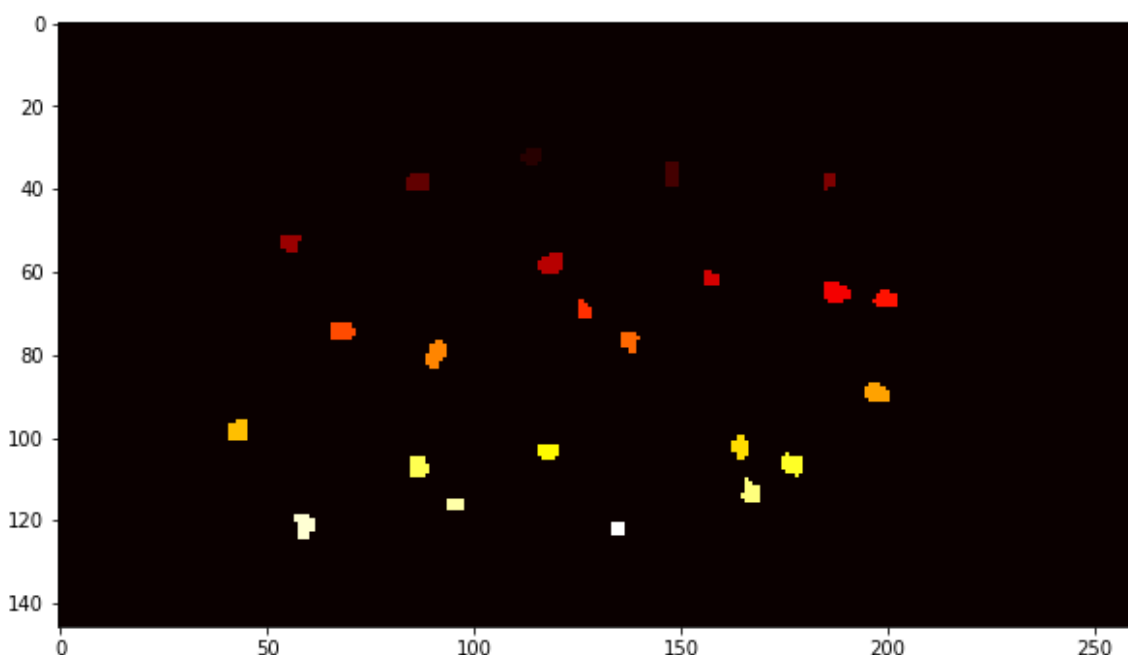
3) Blair-Bliss:

```
def Blair_Bliss(punkty):  
  
    p = len(punkty)  
    a,b = srodek_ciezkosci(punkty)  
  
    k = 0  
    for punkt in punkty:  
        k = k + distance.euclidean(punkt,(a,b))**2  
  
    return p/(math.sqrt(2*math.pi*k))
```

4) Feret:

```
def Feret(punkty):  
  
    a = [x for (y,x) in punkty]  
    b = [y for (y,x) in punkty]  
  
    aa = max(a) - min(a)  
    bb = max(b) - min(b)  
  
    return float(bb)/float(aa)
```

Projekt zaliczeniowy



Ilość obiektów: 23

```
Liczba punktów: 17 Srodek ciezkosci: [32.411764705882355, 114.11764705882354] Blair-Bliss: 1.0012352853750737 Feret: 0.75
Liczba punktów: 18 Srodek ciezkosci: [36.5, 148.0] Blair-Bliss: 0.8941342143133769 Feret: 2.5
Liczba punktów: 23 Srodek ciezkosci: [38.56521739130435, 86.6086956521739] Blair-Bliss: 0.9611832205630371 Feret: 0.6
Liczba punktów: 10 Srodek ciezkosci: [38.2, 185.9] Blair-Bliss: 0.9821280021861475 Feret: 1.5
Liczba punktów: 16 Srodek ciezkosci: [53.3125, 55.75] Blair-Bliss: 0.9798399728863811 Feret: 0.75
Liczba punktów: 24 Srodek ciezkosci: [58.125, 118.79166666666667] Blair-Bliss: 0.974249547101205 Feret: 0.8
Liczba punktów: 14 Srodek ciezkosci: [61.714285714285715, 157.35714285714286] Blair-Bliss: 0.9862316797569728 Feret: 1.0
Liczba punktów: 28 Srodek ciezkosci: [65.03571428571429, 187.60714285714286] Blair-Bliss: 0.9591126799806995 Feret: 0.6666666666666666
Liczba punktów: 18 Srodek ciezkosci: [66.77777777777777, 199.77777777777777] Blair-Bliss: 0.9576979707990727 Feret: 0.6
Liczba punktów: 12 Srodek ciezkosci: [69.41666666666667, 126.83333333333333] Blair-Bliss: 0.9285100169667038 Feret: 2.0
Liczba punktów: 22 Srodek ciezkosci: [74.5, 68.27272727272727] Blair-Bliss: 0.970034611895148 Feret: 0.6
Liczba punktów: 19 Srodek ciezkosci: [76.73684210526316, 137.73684210526315] Blair-Bliss: 0.9837526549151365 Feret: 1.0
Liczba punktów: 24 Srodek ciezkosci: [79.91666666666667, 90.95833333333333] Blair-Bliss: 0.9265153637135087 Feret: 1.5
Liczba punktów: 25 Srodek ciezkosci: [89.2, 197.44] Blair-Bliss: 0.9867556740661929 Feret: 0.8
Liczba punktów: 23 Srodek ciezkosci: [98.17391304347827, 43.130434782608695] Blair-Bliss: 0.9899388074060169 Feret: 1.0
Liczba punktów: 19 Srodek ciezkosci: [102.42105263157895, 164.57894736842104] Blair-Bliss: 0.9684205063946099 Feret: 1.6666666666666667
Liczba punktów: 18 Srodek ciezkosci: [103.33333333333333, 118.0] Blair-Bliss: 0.995820125495468 Feret: 0.75
Liczba punktów: 21 Srodek ciezkosci: [106.42857142857143, 177.0952380952381] Blair-Bliss: 0.9808658747315155 Feret: 1.25
Liczba punktów: 22 Srodek ciezkosci: [107.04545454545455, 86.72727272727273] Blair-Bliss: 0.9981410805697998 Feret: 1.0
Liczba punktów: 20 Srodek ciezkosci: [113.1, 167.2] Blair-Bliss: 0.9469147621168987 Feret: 1.25
Liczba punktów: 12 Srodek ciezkosci: [116.0, 95.5] Blair-Bliss: 0.9982225900859755 Feret: 0.6666666666666666
Liczba punktów: 23 Srodek ciezkosci: [121.26086956521739, 59.08695652173913] Blair-Bliss: 0.9450878521316485 Feret: 1.25
Liczba punktów: 9 Srodek ciezkosci: [122.0, 135.0] Blair-Bliss: 1.0364824484140063 Feret: 1.0
```