



UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE  
AREQUIPA

ESCUELA PROFESIONAL DE  
CIENCIA DE LA COMPUTACIÓN  
COMPUTACIÓN GRÁFICA

---

## Práctica 1: Thresholding

---

***Pertenece a:***

*Carolina Boniee Chávez López*

*Milagros Celia Cruz Mamani*

*Williams Fidel Nifla Ctasi*

# Práctica 1: Thresholding

## 1. Ejercicios

- 1.1. En los tejidos nervisosos de algunos ratones, las células saludables tienen una intensidad mediana de gris, mientras que las células muertas son más densas y oscuras. Desarrolle un programa que quite las células muertas de los siguientes tejidos

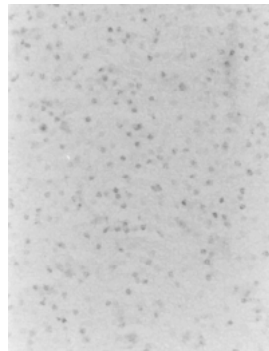


Figura 1: Imagen Fuente

### 1.1.1. Código

#### Algoritmo 1: Código Respuesta

```
1 import matplotlib.pyplot as plt
2 import cv2
3 import numpy as np
4
5
6 def out_celulas_malas(img, imageFile):
7     #_, threshold_binary = cv2.threshold(neg, 90, 150, cv2.THRESH_BINARY)
8     #cv2.imshow('thresholdin', threshold_binary)
9
10    #Obtengo el tamaño de la imagen
11    h, w = img.shape
12
13    #IMREAD_GRAYSCALE = Carga la imagen a escala de Grises.
14    imgGray = cv2.imread(imageFile, cv2.IMREAD_GRAYSCALE)
15    for i in range(h):
16        for j in range(w):
```

```

17         if (img[i][j] >= 193 and img[i][j] <= 195 ):
18             imgGray[i][j] = 255
19         else:
20             imgGray[i][j] = 0
21     cv2.imshow('Sin Celulas Muertas', imgGray)
22
23
24 if __name__ == "__main__":
25
26     imageFile = 'thresh1.png'
27     image = cv2.imread(imageFile, cv2.IMREAD_GRAYSCALE)
28     cv2.imshow('Imagen Fuente', image)
29     out_celulas_malas(image, imageFile)
30
31
32     hist = cv2.calcHist([image], [0], None, [256], [0, 256])
33
34     plt.plot(hist, color='blue')
35     plt.xlabel('Intensidad')
36     plt.ylabel('Pixeles Total')
37     plt.show()
38
39     cv2.destroyAllWindows()

```

### 1.1.2. Resultados

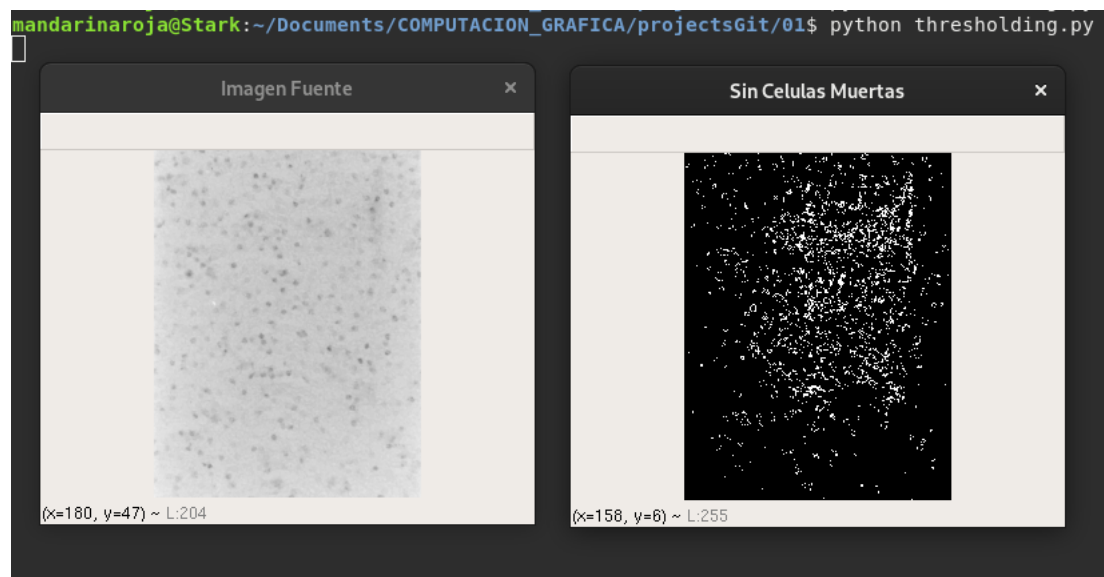


Figura 2: Thresholding

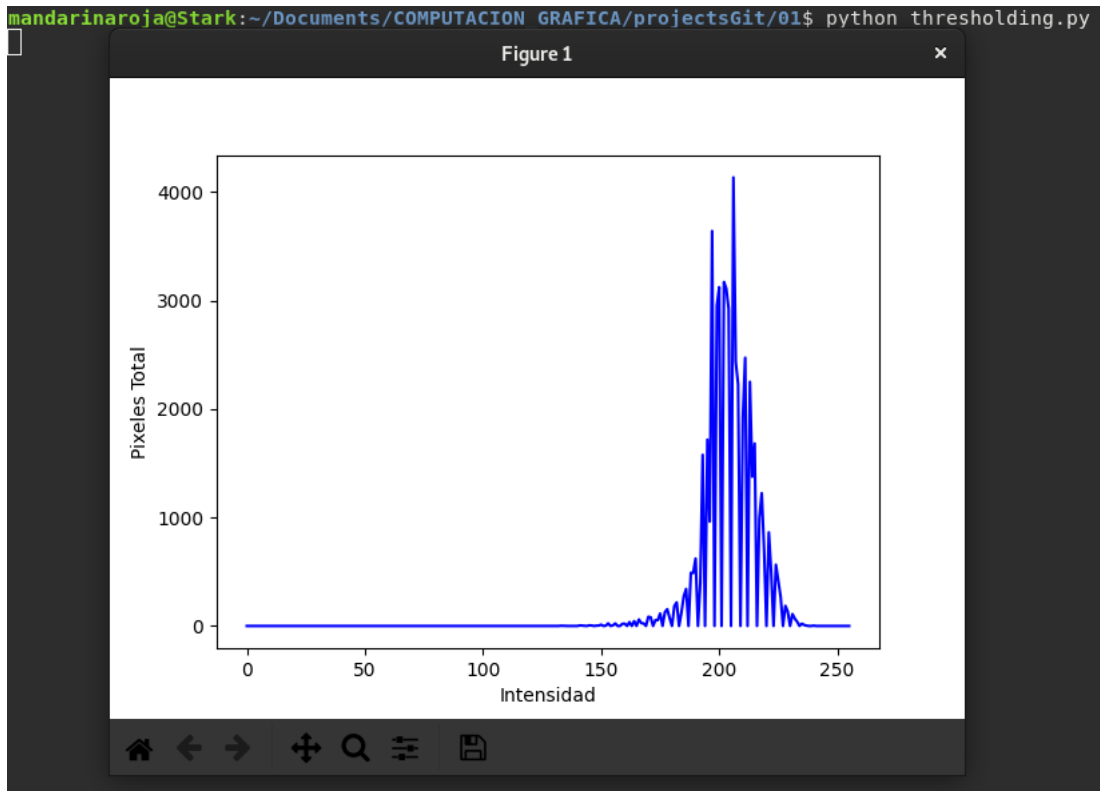


Figura 3: Histograma

1.2. De la imagen anterior, implemente un programa que quite las células saludables.

1.2.1. Código

Algoritmo 2: Código

```

1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 #img = cv2.imread('thresh2.jpg')
6 #gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
7 #neg=255-gray
8 img = cv2.imread('thresh2.png', cv2.IMREAD_GRAYSCALE)
9 cv2.imshow('thresh2', img)
10 #cv2.imshow('gray', gray)
11 #cv2.imshow('neg', neg)
12
13

```

```

14
15 def No_saludables(img):
16     h, w = img.shape
17     gray = cv2.imread('thresh2.png', cv2.IMREAD_GRAYSCALE)
18     for i in range(h):
19         for j in range(w):
20             if (img[i][j] <= 170):
21                 gray[i][j] = 255
22             else:
23                 gray[i][j] = 0
24     cv2.imshow('Sin células saludables', gray)
25
26
27
28
29 No_saludables(img)
30
31 hist = cv2.calcHist([img], [0], None, [256], [0, 255])
32
33 plt.plot(hist, color='gray')
34 plt.xlabel('intensidad de iluminacion')
35 plt.ylabel('cantidad de pixeles')
36 plt.show()
37
38 cv2.waitKey(0)
39 cv2.destroyAllWindows()

```

### 1.2.2. Resultado



Figura 4: Thresholding para quitar células saludables

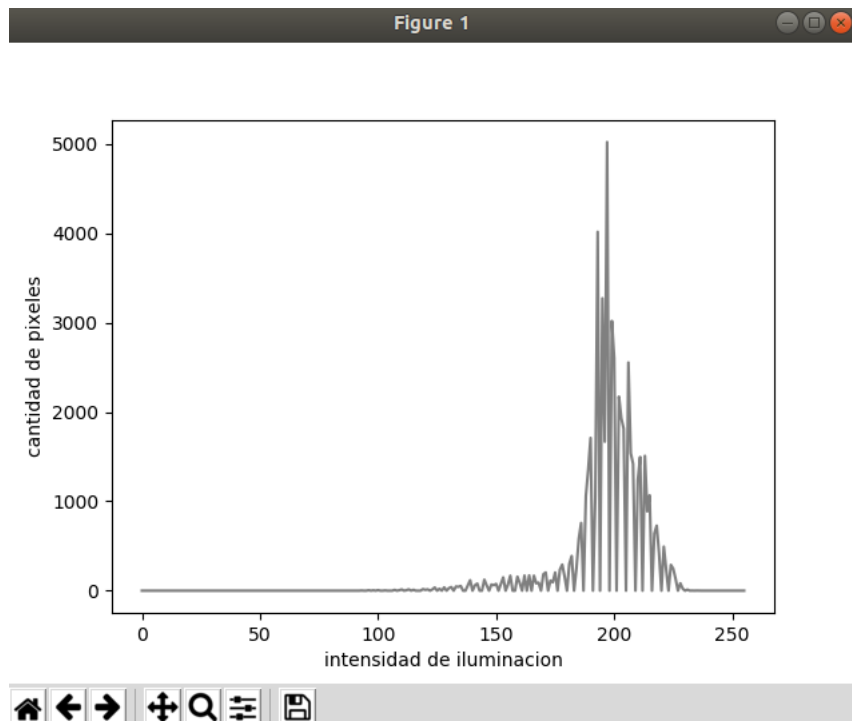


Figura 5: Histograma

- 1.3. Desarrolle un programa que segmente las cosechas de trigo (campos amarillos) en la imagen satelital (ver Figura 6 ).



Figura 6: Imagen satelital

### 1.3.1. Código

Algoritmo 3: Código Respuesta

```
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 # Cargamos la imagen del disco duro
6 imagen = cv2.imread('thresh3.png')
7
8 #convertimos la imagen a escala de grises
9 imgCopia = cv2.imread('thresh3.png', cv2.IMREAD_GRAYSCALE)
10
11 resultado = cv2.imread('thresh3.png')
12
13
14 cv2.waitKey()
15
16 cv2.imshow('Imagen original', imagen)
17 cv2.imshow('Imagen a escala de grises', imgCopia)
18 cv2.imwrite('grises_thresh3.png',imgCopia)
```

```

19
20 histB = cv2.calcHist([imagen], [0], None, [256], [0, 256])
21 histG = cv2.calcHist([imagen], [1], None, [256], [0, 256])
22 histR = cv2.calcHist([imagen], [2], None, [256], [0, 256])
23 histN = cv2.calcHist([imgCopia], [0], None, [256], [0, 256])
24
25
26 height, width, chanel= imagen.shape
27 limi=200 #limite inicial
28 limf=230 #limite final
29
30 #convierte los colores de la cosecha lo verde en negro
31 for i in range(height):
32     for j in range(width):
33         if(imagen[i][j][0]>limi or imagen[i][j][1]>limf or imagen[i][j][2]<
34             limi):
35             resultado[i][j]=0
36
37 cv2.imshow('imagen final',resultado)
38 cv2.waitKey()
39 cv2.imwrite('salida_thresh3.png',resultado)
40
41
42 plt.plot(histN, color='black')
43 plt.xlabel('intensidad de iluminacion')
44 plt.ylabel('cantidad de pixeles')
45 plt.show()

```

### 1.3.2. Resultado

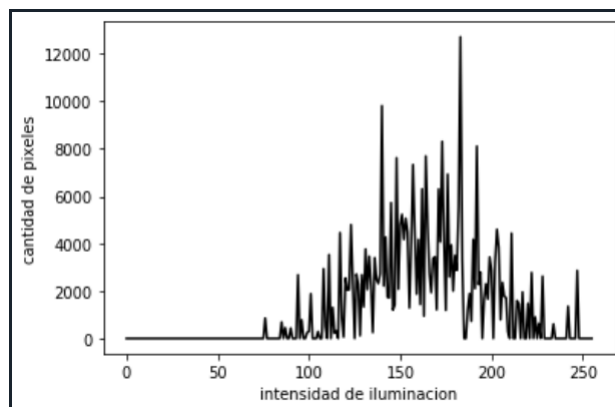


Figura 7: Histograma





Figura 8: Imagen a escala de grises

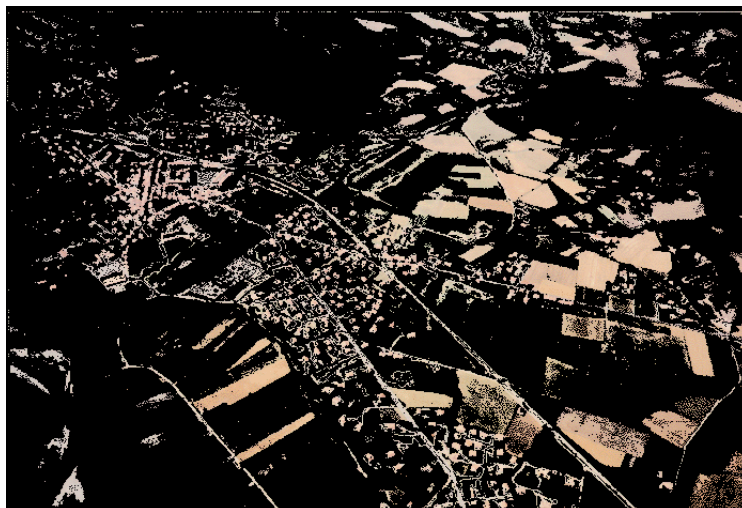


Figura 9: Resultado final

## 2. Conclusiones

- Al realizar el trabajo, en un inicio, nos fue difícil determinar el umbral para así poder diferenciar los píxeles que nos interesa del resto. Con ayuda de la función del histograma, para determinar las intensidades de los píxeles, determinamos el valor.

## 3. Referencias

- <https://www.lpi.tel.uva.es/nacho/docencia/ingond1/trabajos0304/umbralizacion.html>
- <https://sites.google.com/site/cg05procesamientodeimagenes/home/threshold-umbralizacion>

- *https : //www.colorspire.com/rgb – color – wheel/*
- **Enlace de git:** <https://github.com/UNSA-Computacion-Grafica-Grupo-09>