OHC_T1_Umbralizacion

February 6, 2018

0.1 Temas Selectos de Imágenes Biomédicas - Tarea 1 (07 Feb 2018)

0.1.1 Oscar Hernández Constantino

1 Algoritmos de Umbralización

1.1 Algoritmo

```
In [1]: import skimage
        import numpy as np

        # import matplotlib.pyplot as plt
        %pylab inline

        from skimage import data
        from skimage.color import rgb2gray
        from skimage import img_as_ubyte
        from skimage import img_as_float

        from skimage.color.adapt_rgb import adapt_rgb, each_channel
        from math import sqrt
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
In [2]: # @adapt_rgb(each_channel)
        def histograma(g):
            h = np.zeros(256, dtype='uint32')

            M, N = g.shape

            for m in xrange(M):
                for n in xrange(N):
                    h[ g[m,n] ] += 1

            return h

        def generaUmbralizada(g):
            t = metodo_momentos(g)
```

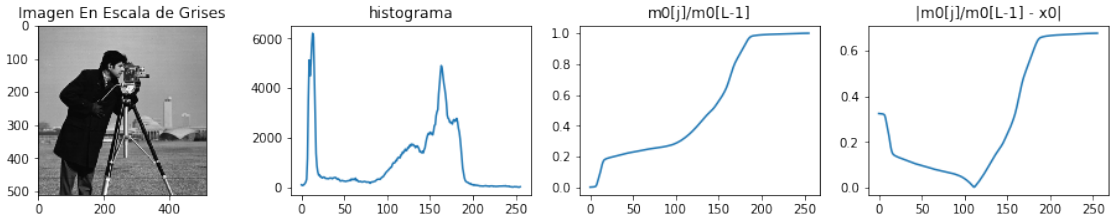
```
    M, N = g.shape
    f = np.zeros((M, N), dtype='uint8')
    for m in xrange(M):
        for n in xrange(N):
            f[m, n] = 255
            if g[m, n] > t:
                f[m, n] = 0
    return f

def metodo_momentos(g):
    h = histograma(g)

    #plot(h)
    M, N = g.shape
    L=256

    m0 = np.zeros(L, dtype='double')
    m1 = np.zeros(L, dtype='double')
    m2 = np.zeros(L, dtype='double')
    m3 = np.zeros(L, dtype='double')

    totalPixeles = 1

    m0[0] = h[0]/totalPixeles

    for j in xrange(1, L):
        m0[j] = m0[j-1] + h[j]
        m1[j] = m1[j-1] + j*h[j]
        m2[j] = m2[j-1] + (j*j)*h[j]
        m3[j] = m3[j-1] + (j*j*j)*h[j]

    aux1 = m0[L-1]*m2[L-1] - m1[L-1]*m1[L-1]

    x1 = m1[L-1]*m3[L-1]/aux1 - m2[L-1]*m2[L-1]/aux1
    x2 = m1[L-1]*m2[L-1]/aux1 - m0[L-1]*m3[L-1]/aux1

    aux2 = sqrt( x2*x2 - 4*x1)
    x0 = -((m1[L-1]/(m0[L-1]*aux2) + (x2/(2*aux2)))) + (0.5)
    print "X0....."
    print x0
#    x0 = 0.324

    tmp = np.zeros(256, dtype='float')
    tmp2 = np.zeros(256, dtype='float')

    for j in xrange(0, L):
        tmp[j] = m0[j]/m0[L-1]
        tmp2[j] = abs(tmp[j] - x0)
```

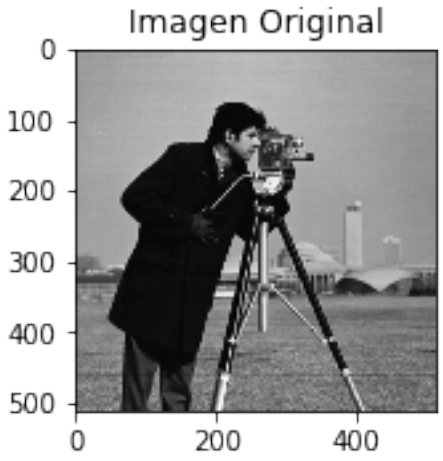
```
    #plot(tmp)
    #plot(tmp2)

    difMin = abs(m0[0]/m0[L-1] - x0)
    indMin = 0
    j = 1;

    fig, (ax0, ax1, ax2, ax3) = plt.subplots(nrows=1,
    ncols=4,
    figsize=(16, 2.5)
                                )
    ax0.imshow(g, cmap=cm.gray)
    ax0.set_title('Imagen En Escala de Grises')

    ax1.plot(h)
    ax1.set_title('histograma')

    ax2.plot(tmp)
    ax2.set_title('m0[j]/m0[L-1]')

    ax3.plot(tmp2)
    ax3.set_title('|m0[j]/m0[L-1] - x0|')

    while j < L:
        aux = abs(m0[j]/m0[L-1] - x0)
        if aux < difMin:
            difMin = aux
            indMin = j
        j += 1
    print("Umbral:", indMin)
    return indMin
```

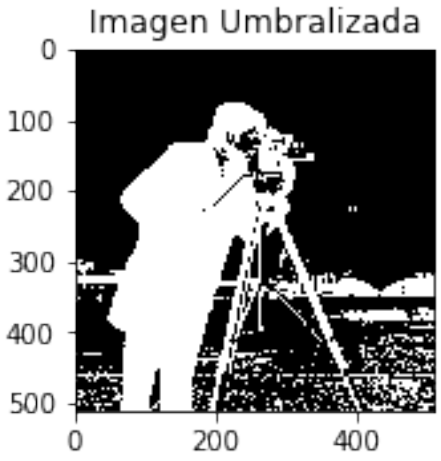
In [3]:
```
## EJEMPLOS:
f = data.camera()
f = img_as_ubyte(f)

g = generaUmbralizada(f)

fig, (ax0, ax1) = plt.subplots(nrows=1,
    ncols=2,
    figsize=(8, 2.5)
                                )
ax0.imshow(f, cmap=cm.gray)
ax0.set_title('Imagen Original')

ax1.imshow(g, cmap=cm.gray)
ax1.set_title('Imagen Umbralizada')
```

```
X0...
0.32431470882
('Umbral:', 111)
```

Out[3]: Text(0.5,1,u'Imagen Umbralizada')







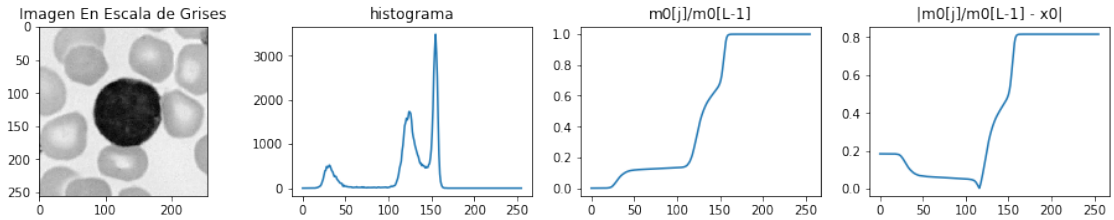
In [4]:
```
from skimage import io
f1 = io.imread("1.png")
f1 = img_as_ubyte(rgb2gray(f1))

f2 = io.imread("2.png")
f2 = img_as_ubyte(rgb2gray(f2))

f3 = io.imread("3.png")
f3 = img_as_ubyte(rgb2gray(f3))

f4 = io.imread("4.png")
f4 = img_as_ubyte(rgb2gray(f4))

#print t
```

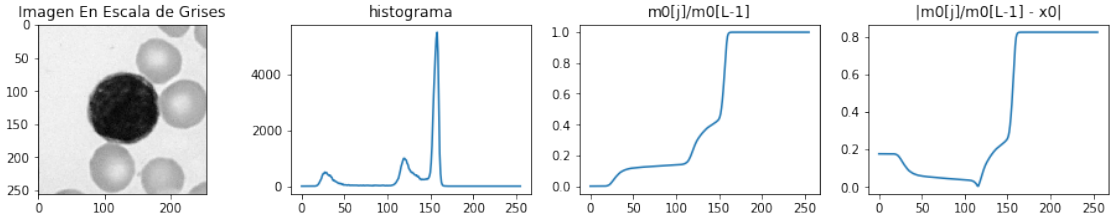
```
        g1 = generaUmbralizada(f1)
        g2 = generaUmbralizada(f2)
        g3 = generaUmbralizada(f3)
        g4 = generaUmbralizada(f4)

        fig, (ax0, ax1, ax2, ax3) = plt.subplots(nrows=1,
            ncols=4,
            figsize=(16, 2.5)
                                        )
        ax0.imshow(g1, cmap=cm.gray)
        ax0.set_title('Imagen Umbralizada')

        ax1.imshow(g2, cmap=cm.gray)
        ax1.set_title('Imagen Umbralizada')

        ax2.imshow(g3, cmap=cm.gray)
        ax2.set_title('Imagen Umbralizada')

        ax3.imshow(g4, cmap=cm.gray)
        ax3.set_title('Imagen Umbralizada')
```

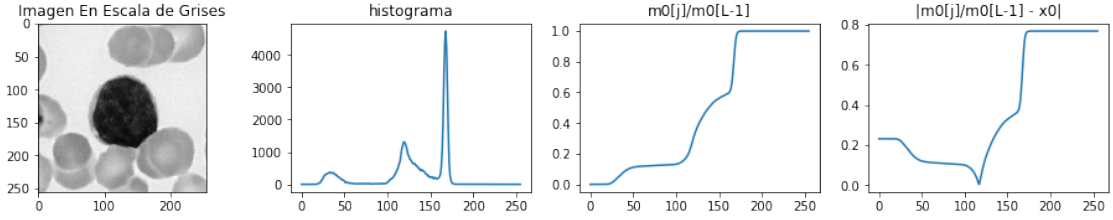
```
/home/oscahern/miniconda2/envs/prose/lib/python2.7/site-packages/skimage/util/dtype.py:122: User
  .format(dtypeobj_in, dtypeobj_out))
```

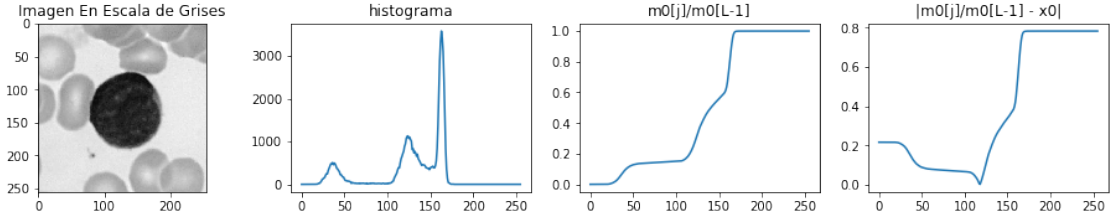
```
X0...
0.184175047247
('Umbral:', 116)
X0...
0.176869624253
('Umbral:', 116)
X0...
0.232153826728
('Umbral:', 117)
X0...
0.217425160801
('Umbral:', 118)
```

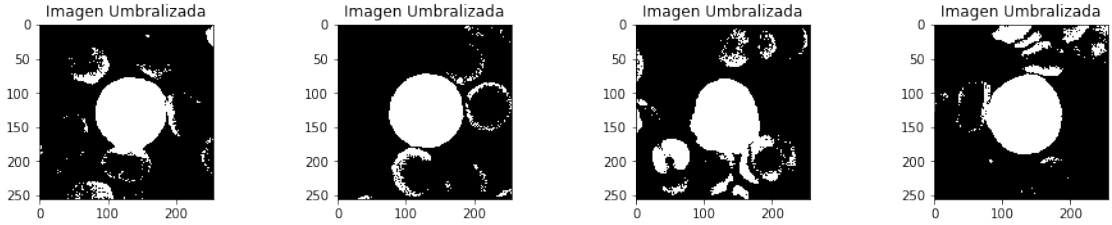
Out[4]: Text(0.5,1,u'Imagen Umbralizada')












```
In [5]: f1 = io.imread("1.jpg")
        f1 = img_as_ubyte(rgb2gray(f1))

        f2 = io.imread("2.jpg")
        f2 = img_as_ubyte(rgb2gray(f2))
```

```
        f3 = io.imread("3.jpg")
        f3 = img_as_ubyte(rgb2gray(f3))

        f4 = io.imread("4.jpg")
        f4 = img_as_ubyte(rgb2gray(f4))

        g1 = generaUmbralizada(f1)
        g2 = generaUmbralizada(f2)
        g3 = generaUmbralizada(f3)
        g4 = generaUmbralizada(f4)

        fig, (ax0, ax1, ax2, ax3) = plt.subplots(nrows=1,
            ncols=4,
            figsize=(16, 2.5)
                                        )
        ax0.imshow(g1, cmap=cm.gray)
        ax0.set_title('Imagen Umbralizada')

        ax1.imshow(g2, cmap=cm.gray)
        ax1.set_title('Imagen Umbralizada')

        ax2.imshow(g3, cmap=cm.gray)
        ax2.set_title('Imagen Umbralizada')

        ax3.imshow(g4, cmap=cm.gray)
        ax3.set_title('Imagen Umbralizada')
```

```
X0...
0.979194661013
('Umbral:', 117)
X0...
0.971232693901
('Umbral:', 119)
X0...
0.968871187882
('Umbral:', 118)
X0...
0.966366431843
('Umbral:', 118)
```

```
Out[5]: Text(0.5,1,u'Imagen Umbralizada')
```

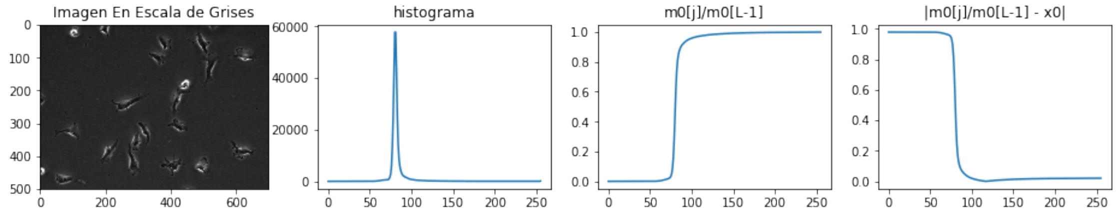


Imagen En Escala de Grises
histograma
m0[j]/m0[L-1]
|m0[j]/m0[L-1] - x0|

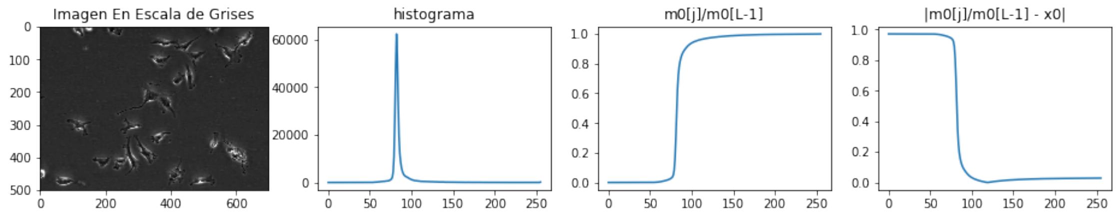


Imagen En Escala de Grises
histograma
m0[j]/m0[L-1]
|m0[j]/m0[L-1] - x0|

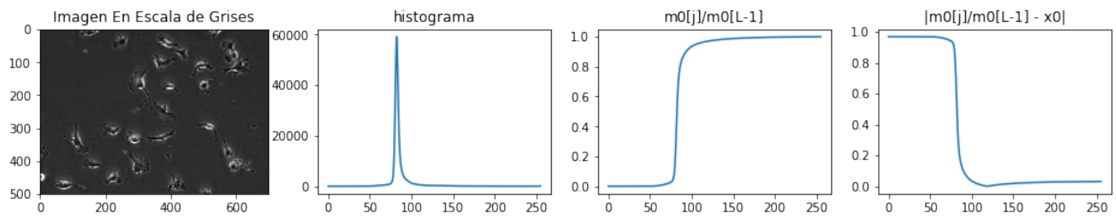


Imagen En Escala de Grises
histograma
m0[j]/m0[L-1]
|m0[j]/m0[L-1] - x0|

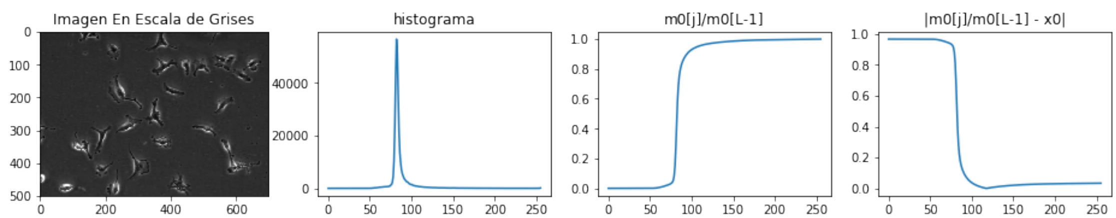


Imagen En Escala de Grises
histograma
m0[j]/m0[L-1]
|m0[j]/m0[L-1] - x0|

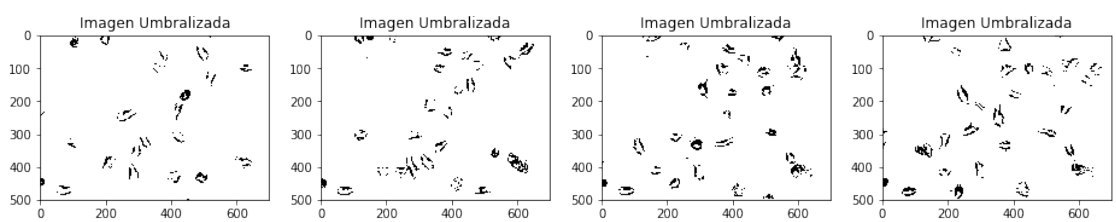
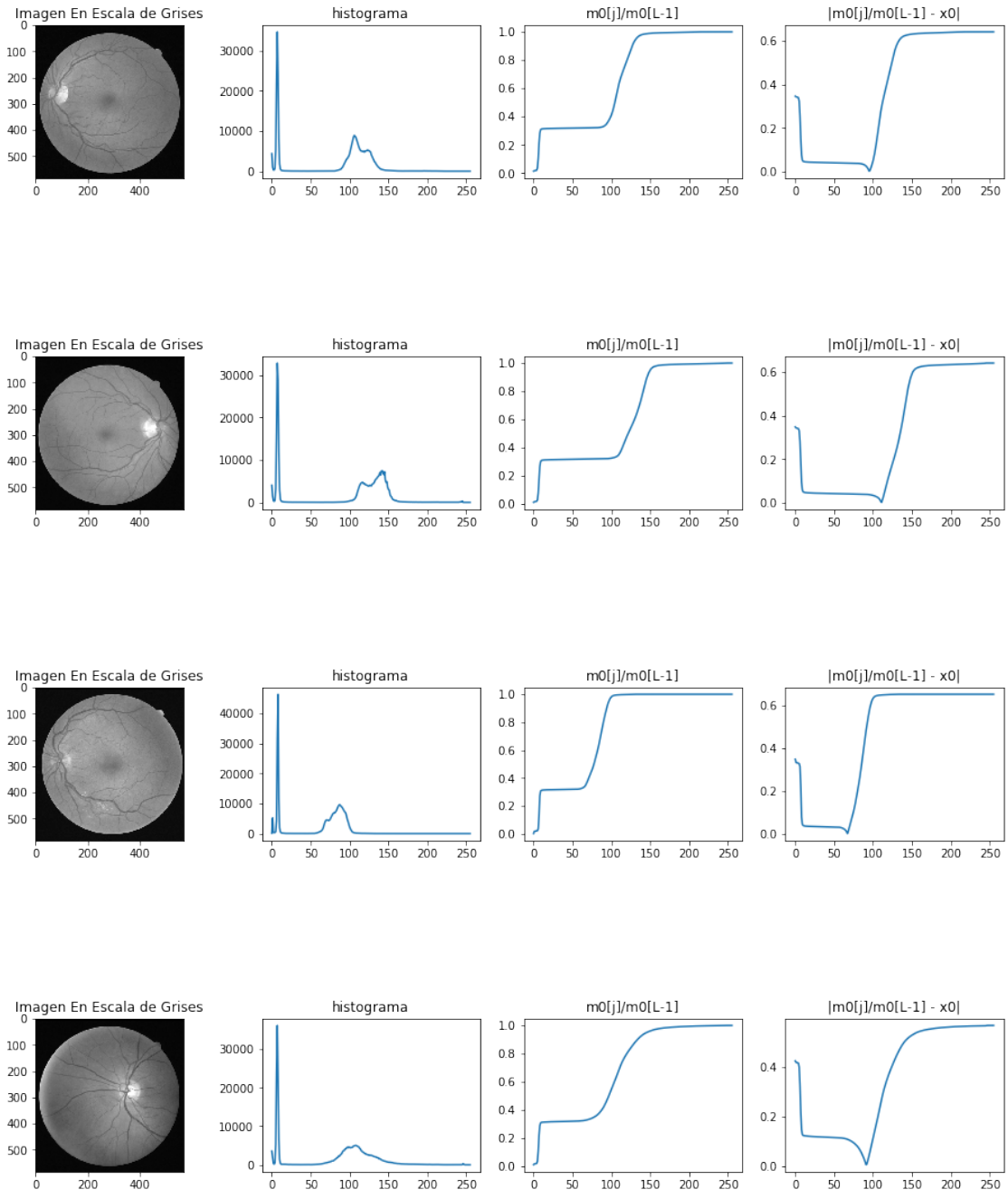


Imagen Umbralizada
Imagen Umbralizada
Imagen Umbralizada
Imagen Umbralizada

```
In [6]: f1 = io.imread("01 test.tif")
        f1 = img_as_ubyte(rgb2gray(f1))

        f2 = io.imread("02 test.tif")
        f2 = img_as_ubyte(rgb2gray(f2))

        f3 = io.imread("03 test.tif")
        f3 = img_as_ubyte(rgb2gray(f3))

        f4 = io.imread("04 test.tif")
        f4 = img_as_ubyte(rgb2gray(f4))


        #print t
        g1 = generaUmbralizada(f1)
        g2 = generaUmbralizada(f2)
        g3 = generaUmbralizada(f3)
        g4 = generaUmbralizada(f4)

        fig, (ax0, ax1, ax2, ax3) = plt.subplots(nrows=1,
            ncols=4,
            figsize=(16, 2.5)
                                        )
        ax0.imshow(g1, cmap=cm.gray)
        ax0.set_title('Imagen Umbralizada')

        ax1.imshow(g2, cmap=cm.gray)
        ax1.set_title('Imagen Umbralizada')

        ax2.imshow(g3, cmap=cm.gray)
        ax2.set_title('Imagen Umbralizada')

        ax3.imshow(g4, cmap=cm.gray)
        ax3.set_title('Imagen Umbralizada')

/home/oscahern/miniconda2/envs/prose/lib/python2.7/site-packages/skimage/external/tifffile/tifff
  strip = decompress(strip)


X0...
0.359505513541
('Umbral:', 95)
X0...
0.359984152025
('Umbral:', 111)
X0...
0.348413971648
('Umbral:', 67)
```

```
X0...
0.433986430541
('Umbral:', 91)
```

Out[6]: Text(0.5,1,u'Imagen Umbralizada')



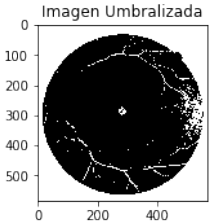


Imagen Umbralizada
0
100
200
300
400
500
0
200
400

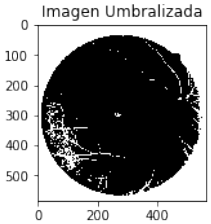



Imagen Umbralizada
0
100
200
300
400
500
0
200
400

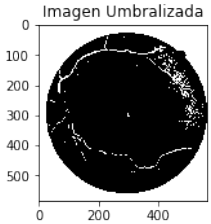



Imagen Umbralizada
0
100
200
300
400
500
0
200
400

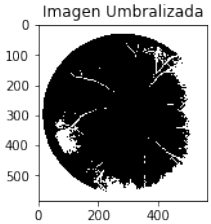



Imagen Umbralizada
0
100
200
300
400
500
0
200
400