

Importación de librerías

```
In [1]: #Librerías
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
```

Definición de la función de entropía

```
In [2]: #Calculo de entropía para un vector dado
def entropy(signal):
    #La función devuelve la entropía de el array de entrada de 1 sola dime
nsión
    lensig=signal.size
    symset=list(set(signal))
    numsym=len(symset)
    propab=[np.size(signal[signal==i])/(1.0*lensig) for i in symset]
    ent=np.sum([p*np.log2(1.0/p) for p in propab])
    return ent
```

Tratamiento de la imagen

```
In [3]: #Importación de imagen
colorIm=Image.open('tema5_actividad.png')
#Pasar a escala de grises
greyIm=colorIm.convert('L')
#Numpy array
colorIm=np.array(colorIm)
greyIm=np.array(greyIm)
```

Función máscara

```
In [4]: def entropmask(imagen,sizemask):  
        img=np.array(imagen)  
        N=(sizemask-1)//2  
        S=img.shape  
        for row in range(S[0]):  
            for col in range(S[1]):  
                Lx=np.max([0,col-N])  
                Ux=np.min([S[1],col+N])  
                Ly=np.max([0,row-N])  
                Uy=np.min([S[0],row+N])  
                region=imagen[Ly:Uy,Lx:Ux].flatten()  
                img[row,col]=entropy(region)  
  
        return img
```

Aplicar máscaras

```
In [5]: #Máscara 3x3  
        E3=entropmask(greyIm,3)  
        #Máscara 5x5  
        E5=entropmask(greyIm,5)  
        #Máscara 7x7  
        E7=entropmask(greyIm,7)
```

```

In [6]: #Imprimimos las imagenes para las diferentes máscaras
fig, (ax0, ax1, ax2, ax3) = plt.subplots(ncols=4, figsize=(12, 4), sharex=True,
sharey=True)

img0 = ax0.imshow(greyIm, cmap=plt.cm.gray)
ax0.set_title("Imagen")
ax0.set_xlabel("Imagen Original")
fig.colorbar(img0, ax=ax0)

img1 = ax1.imshow(E3, cmap='jet')
ax1.set_title("Entropy 3x3")
ax1.set_xlabel("Máscara 3")
fig.colorbar(img1, ax=ax1)

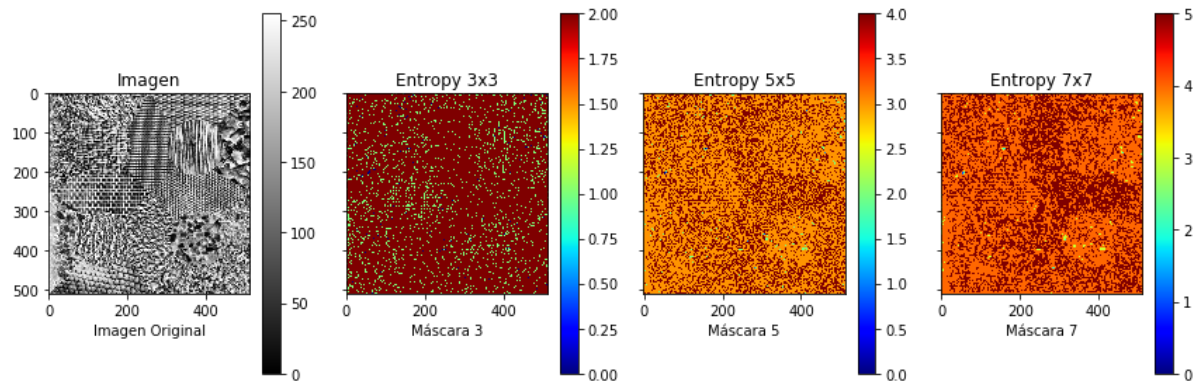
img2 = ax2.imshow(E5, cmap='jet')
ax2.set_title("Entropy 5x5")
ax2.set_xlabel("Máscara 5")
fig.colorbar(img2, ax=ax2)

img3 = ax3.imshow(E7, cmap='jet')
ax3.set_title("Entropy 7x7")
ax3.set_xlabel("Máscara 7")
fig.colorbar(img3, ax=ax3)

fig.tight_layout()

plt.show()

```



El cálculo de estos valores de entropía es más costoso cuanto mayor es la máscara a aplicar. A mayor máscara también podemos notar algo mejor la detección de texturas.

Los gráficos de entropía darían mejores resultado para detectar bordes y fondos de imágenes.