

Apellidos, Nombre: Fernández Cabrita, Alejandro
Apellidos, Nombre: Evangelista Sarabia, Carlos
Apellidos, Nombre: De Almeida Pissarra, Inês

Ejercicio 1:

Volcad valores máximo/mínimo obtenidos, así como el valor medio (función mean).

Valor máximo: 255, valor mínimo: 0, valor medio: 125.0347

Adjuntad la imagen en este caso. ¿Por qué pasa esto?



Los valores de cada píxel al dividirse por 255 antes de hacer la conversión a double, se redondean a 0 (<0.5) o a 1 (≥ 0.5), pues se trata de una división entera. Esto hace que en la imagen solo se vean dos colores, el negro y el blanco.

Adjuntad imagen resultado.



Hacedlo y volcad el máximo y mínimo de la imagen resultante.

Máximo de la imagen con ruido es: 1.349329

Mínimo de la imagen con ruido es: -0.4072875

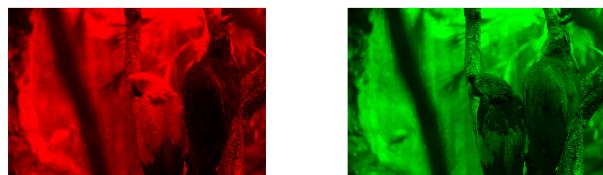
Indicad qué comandos usaríamos para hacer esto SIN BUCLES (trabajando con la imagen en su conjunto). Visualizar y adjuntad la imagen resultante.

```
image_noisy = image_double+0.1*randn(size(image_double));  
image_noisy(image_noisy>1)=1;  
image_noisy(image_noisy<0)=0;
```



Ejercicio 2:

Adjuntad la figura resultante.



Adjuntad la imagen obtenida con `imshow(im(:,:,3 1 2))`. ¿Qué estamos haciendo?



Intercambiamos los planos de color de la imagen. Los colores del canal azul se muestran en el canal rojo, los del canal rojo en el verde; y los del canal verde en el azul.

En cada caso dad el comando usado y adjuntad la imagen resultante.
`hsv1(:,:,3)=0.5;`



```
hsv2(:,:,2)=1;
```



```
hsv3(:,:,1)=0.33;
```



¿Por qué creéis que se usan esta fórmula y no simplemente la media $(R+G+B)/3$?
Porque el ojo humano es menos sensible a la luz azul, seguidamente a la luz roja y por último y con mayor sensibilidad, la luz verde.

Ejercicio 3:

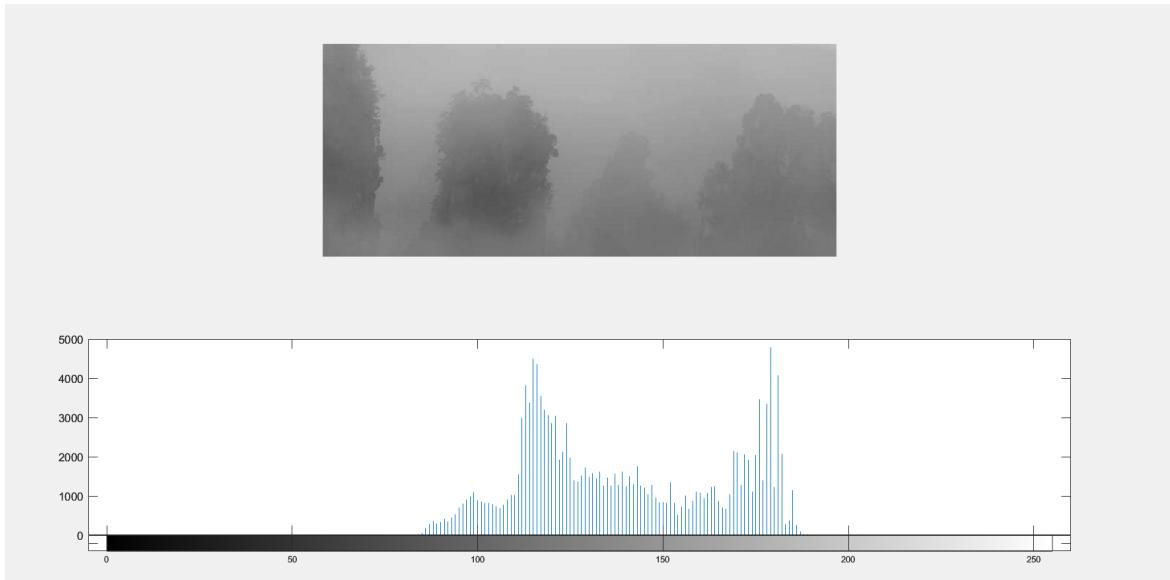
Cargad la imagen 'bw1.png' y obtener su media usando `mean2()` y comprobad que es igual a la calculada como `mean(im(:))`.

```
%Cargamos la imagen
im=imread("bw1.png");
%Para la media de los pixeles se puede usar mean2 sobre la matriz, o mean
%sobre la matriz trasladada a vector
media=mean2(im);
fprintf("Medio de la imagen usando mean2: %d\n", media);
media=mean(im(:));
fprintf("Medio de la imagen usando mean: %d\n", media);
```

```
>> Ej3
Medio de la imagen usando mean2: 1.388968e+02
Medio de la imagen usando mean: 1.388968e+02
```

Adjuntar código de vuestra función y la figura resultante (imagen + histograma) para la imagen cargada.

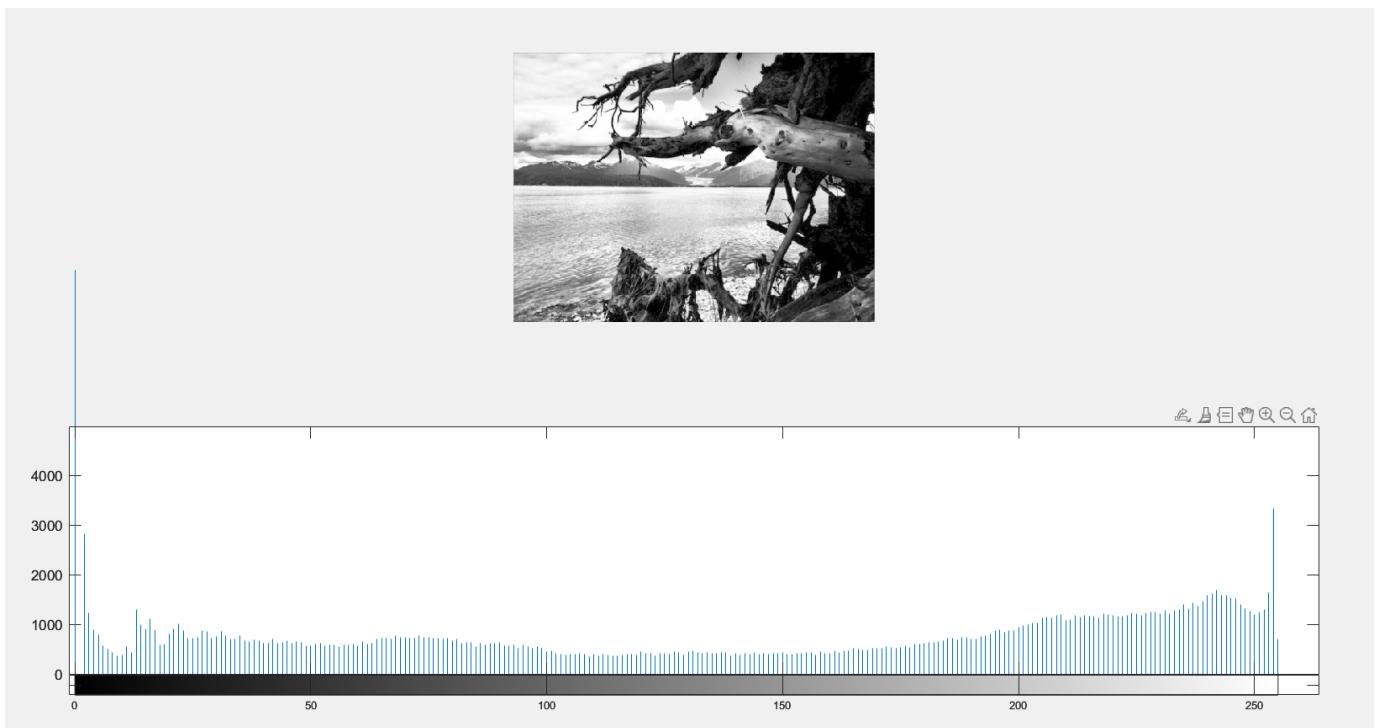
```
function show_hist(im_monocromo)
    figure();
    subplot(2,1,1);
    imshow(im_monocromo);
    subplot(2,1,2);
    imhist(im_monocromo);
    set(gca, 'Xlim', [-5 260]);
    set(gca, 'Ylim', [0 5000]);
end
```



Cargad ahora la imagen 'bw2.png' y obtener su media.

```
Medio de la imagen usando mean2: 1.385039e+02
```

Adjuntad la imagen + histograma obtenida con show_hist().



¿A qué creéis que es debido los picos a la izquierda y a la derecha del histograma?
 ¿A qué zonas de la imagen corresponden?

Creemos que se deben a que, al tratarse de una imagen de alto contraste, se abarca todo el rango completo de colores, y se pueden visualizar el color negro, que corresponde con el valor 0, y tiene la mayor frecuencia de aparición (8130 veces), seguido del color 254, muy cercano al blanco (3333 veces). Estos dos colores son los que más aparecen en la imagen, por lo que tiene sentido que formen picos.

Calculad las desviaciones standard σ de nuestras 2 imágenes (con la función std2).

```
-----
Vmax píxeles de bw1.png: 192
Vmin píxeles de bw1.png: 82
Media valor píxeles de bw1.png: 1.388968e+02
Desv. típica píxeles de bw1.png: 2.703342e+01
-----
Vmax píxeles de bw2.png: 255
Vmin píxeles de bw2.png: 0
Media valor píxeles de bw2.png: 1.385039e+02
Desv. típica píxeles de bw2.png: 8.597676e+01
```

Se aprecia que la imagen bw1, presenta un rango de valores de píxeles más reducido [82, 192], frente a la imagen bw2, que abarca todo el rango posible de colores [0, 255]. Se aprecia también que los valores de desviación típica en la primera imagen es significativamente menor que los de la segunda imagen, por lo que esta última presenta un mayor contraste.

Ejercicio 4:

Adjuntad vuestro código y la imagen obtenida.,. Volcad para cada plano los valores de partida (m_0, σ_0) y los nuevos (m, σ). Usad fprintf con 2 decimales (%.2f).

% apartado 4.1

```
im1=imread('faro.jpg');
im2=imread('playa.jpg');

ntsc1=rgb2ntsc(im1);
ntsc2=rgb2ntsc(im2);

for i = (1:3)
    fprintf("plano %.0f: media de partida: %.2f, desv. estandar de partida: %.2f\n", i,
mean2(ntsc2(:,:,i)), std2(ntsc2(:,:,i)));
    ntsc2(:,:,:,i) = (ntsc2(:,:,:,i)-mean2(ntsc2(:,:,:,i)))/std2(ntsc2(:,:,:,i));
    ntsc2(:,:,:,i) = (ntsc2(:,:,:,i)*std2(ntsc1(:,:,:,i))) + mean2(ntsc1(:,:,:,i));
    fprintf("plano %.0f: media transferida: %.2f, desv. estandar transferida: %.2f\n", i,
mean2(ntsc2(:,:,i)), std2(ntsc2(:,:,i)));
end

res = ntsc2rgb(ntsc2);
figure(1);
imshow(res);
```

Valores

plano 1: media de partida: 0.59, desv. estandar de partida: 0.24
 plano 1: media transferida: 0.30, desv. estandar transferida: 0.25

plano 2: media de partida: -0.07, desv. estandar de partida: 0.10
 plano 2: media transferida: 0.16, desv. estandar transferida: 0.16

plano 3: media de partida: 0.01, desv. estandar de partida: 0.02
 plano 3: media transferida: -0.00, desv. estandar transferida: 0.03

