

# IEE239 - Procesamiento de Señales e Imágenes Digitales

## Laboratorio 4 - Ejercicios Propuestos

### Segundo Semestre 2017

1. Dado un conjunto de imágenes (**flash.png**, **towers.png**, **marvel.png** y **fondo.png**)<sup>1</sup> se evaluarán diferentes técnicas de procesamiento como filtrado, umbralización y transformación espacial.

#### Observaciones:

- Tener en cuenta que para la correcta visualización de la imágenes usando **imshow()** se deben de respetar los rangos de intensidades permitidas para cada tipo de dato. Por ejemplo: Para los datos tipo uint8, las intensidades deben estar definidas entre 0 a 255 y para los datos tipo double las intensidades deben estar normalizadas entre 0 y 1.
  - Realizar un cast significa convertir un tipo de dato a otro. Por ejemplo: para convertir un dato a tipo double se utiliza **double()**.
- a. Leer las imágenes (flash.png, towers.png, marvel.png y fondo.png), y convertir sólo las dos primeras a escala de grises usando el comando **rgb2gray()**. Finalmente sobre todas las imágenes hacer un cast a double y normalizar.

```
1 % Lectura de imagen
2 Iml_color = imread('flash.png');
3 % Convertimos la imagen de RGB a escala de grises
4 Iml_gray = rgb2gray(Iml_color);
5 % Hacemos un cast de un dato tipo uint8 a double y normalizamos
6 Im1 = double(Iml_gray)/255;
7
8 Im2 = double(rgb2gray(imread('towers.png')))/255;
9 Im3 = double(imread('marvel.png'))/255;
10 Im4 = double(imread('fondo.png'))/255;
```

- b. En este inciso se evaluará la eliminación de ruido (denoising) gaussiano y cómo el orden del filtro afecta sobre la imagen resultante. Los comandos a utilizar son **imnoise()**, **fspecial()** y **conv2()**.
  - i. Sobre la primera imagen, agregar ruido gaussiano blanco con media cero y desviación estandar igual a 0.05 y 0.2 con el fin de obtener dos imágenes corrompidas.
  - ii. Generar las máscaras de los filtros promedio para los tamaños  $5 \times 5$ ,  $10 \times 10$  y  $15 \times 15$ .
  - iii. Realizar la convolución entre los filtros y las imágenes corrompidas anteriormente.

```
1 % Se agrega ruido gaussiano blanco con media cero y de varianza igual a ...
   sigma cuadrado sobre la imagen
```

---

<sup>1</sup>La imagen está almacenada en la carpeta 'laboratorio/lab04/'

```

2  Im_cor1 = imnoise(Im1,'gaussian', 0.05^2);
3  Im_cor2 = imnoise(Im1,'gaussian', 0.2^2);
4
5  % Se define los filtros promedio
6  h1 = fspecial('average',5);
7  h2 = fspecial('average',10);
8  h3 = fspecial('average',15);
9
10 % Filtrado del ruido con sigma = 0.05
11 Im_filt1 = conv2(Im_cor1,h1,'same');
12 Im_filt2 = conv2(Im_cor1,h2,'same');
13 Im_filt3 = conv2(Im_cor1,h3,'same');
14 % Similar para la otra imagen corrompida con sigma = 0.2 ...

```

La relación señal a ruido (SNR por su siglas en inglés) se define como la proporción existente entre la potencia de la señal de referencia y la potencia del ruido que la corrompe. Esta es una métrica de calidad muy utilizada para las tareas de eliminación de ruido, ya que se quiere saber qué tan buena es la técnica de filtrado realizada. Por otro lado, en la Figura 1 se puede observar que a mayor varianza, el SNR disminuye (hay mayor nivel de ruido).

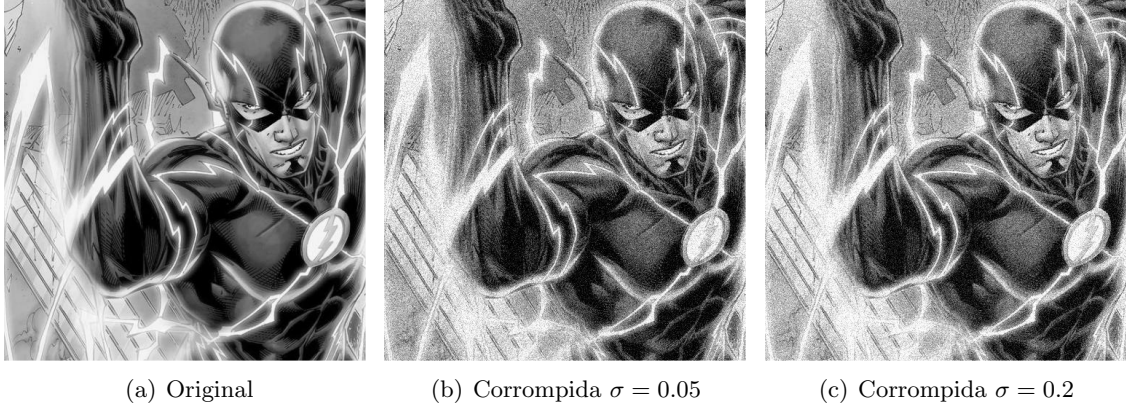


Figure 1: Imágenes corrompidas con ruido gaussiano blanco. (a) SNR = 9.45 dB y (b) SNR = 8.83 dB

El filtro promedio definido como

$$w(x, y) = \frac{1}{M \times N} \begin{bmatrix} \boxed{1} & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

cuyo tamaño es  $M \times N$ , es efectivo ante el ruido gaussiano  $\eta \sim \mathcal{N}(\mu, \sigma^2)$ , puesto al promediar una vecindad de  $M \times N$  el promedio del ruido existente en esa vecindad sería cercano a cero ( $E[\mu] = 0$ ). Si bien a mayor tamaño de filtro hay mayor probabilidad de que la media del ruido en la vecindad sea igual a cero, el problema de hacer un promedio en una ventana cada vez más grandes es que aparece una difuminación mayor de los bordes y se pierden detalles de contraste. Como se puede observar en la Figura 2 a mayor tamaño de filtro la imagen se ve mas difuminada.



(a) Denoising - Filtro  $5 \times 5$

(b) Denoising - Filtro  $10 \times 10$

(c) Denoising - Filtro  $15 \times 15$

Figure 2: Imágenes resultantes luego de aplicar los filtros promedios de tamaño (a)  $5 \times 5$  (b)  $10 \times 10$  y (c)  $15 \times 15$

- c. Agregar un 10 % de ruido sal y pimienta sobre la imagen original 1 y evaluar si el filtro medio o mediano de orden 5 tiene un mejor efecto para la eliminación de este tipo de ruido. Los comandos a utilizar son `conv2()` y `medfilt2()`.

```
1 % Se corrompe la imagen 0.1 de ruido sal y pimienta
2 Im_cor = imnoise(Iml, 'salt & pepper', 0.10);
3
4 % Denoising con un filtro promedio
5 Im_filt_mean = conv2(Im_cor, h1, 'same');
6 % Denoising con un filtro mediano
7 Im_filt_median = medfilt2(Im_cor, [5 5]);
```

En la figura 3, se puede observar que la técnica más efectiva para eliminar el ruido sal y pimienta es con un filtro mediano ya que tanto los pixeles sal y pimienta son intensidades extremas que quedarían excluidas como valor esperado en la vecindad a evaluar. Por otro lado, en el filtrado con un filtro promedio estos valores serían predominantes.



(a) Imagen corrompida

(b) Filtro promedio

(c) Filtro mediano

Figure 3: Imágenes resultantes luego de aplicar el filtro promedio (a)  $\text{SNR} = 10.39 \text{ dB}$  y el filtro mediano (b)  $\text{SNR} = 13.34 \text{ dB}$

- d. Haciendo uso de la segunda imagen (towers.png), se aplicaran las transformaciones espaciales de escalamiento y rotación con el fin de analizar cómo los métodos de interpolación

involucrados en estas transformaciones afectan en la imagen resultante.

- i. Realizar un escalamiento de 0.2 veces el tamaño original, utilizando el comando **imresize()**, con los métodos de interpolación vecino más cercano y bilineal
- ii. Realizar un rotación de 45 grados en sentido antihorario, utilizando el comando **imrotate()**, con los mismos métodos de interpolación que el inciso anterior.

```
1 % Se escala la imagen en 2 con diferentes tipos de interpolación
2 Im_near = imresize(Im2, 0.2, 'nearest');
3 Im_bil = imresize(Im2, 0.2, 'bilinear');
4 Im_bic = imresize(Im2, 0.2, 'bicubic');
5 % Rotación disponible en el .m adjunto
```



(a) Original



(b) Escalada-Vecino más cercano

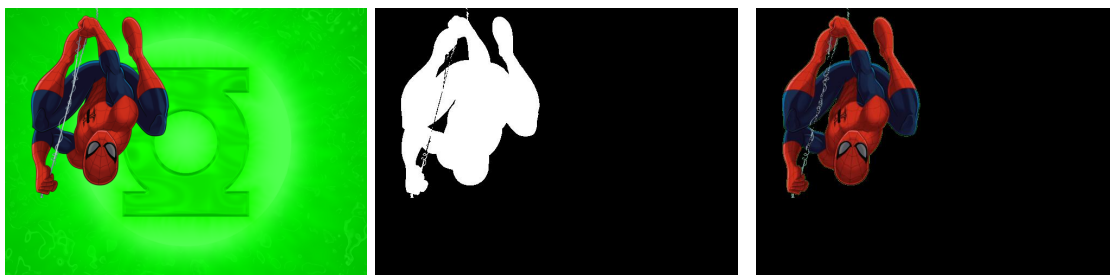


(c) Escalada-Bilineal

Figure 4: Escalamiento de la imagen original towers.png por medio de los métodos de interpolación (a) vecino más cercano y (b) bilineal.

- e. Sobre la capa verde de la tercera imagen (marvel.png), separar la imagen del hombre araña del fondo. Para ello, graficar la capa verde y encontrar un umbral que permita separar el fondo del objetivo. Luego, copiar el valor de los píxeles del personaje en la parte superior izquierda de la imagen fondo.png.

```
1 % Haciendo uso del cursor sobre grafica de la capa verde se halló el ...
   umbral igual 0.6863,
2 % el cual permite separar la imagen objetivo (spider man) del fondo
3 mask = Im3_g < 0.6863;
4
5 % Segmentación
6 seg = zeros(size(Im3));
7 seg(:, :, 1) = mask.*Im3_r;
8 seg(:, :, 2) = mask.*Im3_g;
9 seg(:, :, 3) = mask.*Im3_b;
10
11 % Superponiendo la imagen segmentada en la parte superior izquierda del ...
   nuevo fondo
12 Im4(1:494, 1:668, 1) = not(mask).*Im4(1:494, 1:668, 1) + seg(:, :, 1);
13 Im4(1:494, 1:668, 2) = not(mask).*Im4(1:494, 1:668, 2) + seg(:, :, 2);
14 Im4(1:494, 1:668, 3) = not(mask).*Im4(1:494, 1:668, 3) + seg(:, :, 3);
```



(a) Imagen marvel.png

(b) Mascara

(c) Imagen segmentada



(d) Imagen fondo.png

(e) Imagen reposicionada

2. Dada dos imágenes en escala de grises de bajo contraste **church.jpg** y **landscape.jpg**<sup>2</sup>. Realizar los siguientes pasos:
  - a. Leer las imágenes, realizar el cast a un dato tipo double y normalizar sus intensidades.
  - b. Calcular el histograma y obtener la moda y desviación estándar. ¿Encuentra alguna relación entre la distribución del histograma de la imagen y los datos hallados con el poco contraste que tiene? Los comandos a utilizar son **imhist()**, **std()** y **mode()**.
  - c. Basado en el análisis del histograma de cada imagen hecho en b, se le pide aplicar una de las 2 transformaciones siguientes a las imágenes originales: Gamma con parámetro 0.8 o transformación de intensidad logarítmica. Graficar y responder : ¿Se notan mejoras en el contraste con alguna de las transformaciones? ¿Cómo ha cambiado el histograma después de aplicar las transformaciones?
  - d. Aplicar sobre la imagen una ecualización de histograma. Mostrar la nueva imagen y su histograma. ¿Cuál ha sido el cambio sobre el contraste de la imagen? ¿Por qué la ecualización de histograma funciona o no funciona adecuadamente en este caso? Los comandos a utilizar son **histeq()** e **imhist()**.

#### Notas:

- La transformada logarítmica es definida como  $g(x, y) = c \cdot \log_{10}(1 + f(x, y))$ .
  - El comando **imhist(I, N)** permite graficar el histograma de una imagen *I* a N bins (número de intensidades). Si *I* es de tipo “double”, sus intensidades deben estar normalizadas y si *I* es de tipo “uint8” las intensidades deben estar entre 0 a 255.
3. El pre-procesamiento de imágenes comprende un conjunto de técnicas que son utilizadas en imágenes digitales para mejorar la calidad de la misma o simplificar la obtención de alguna característica particular. En esta etapa se realiza la corrección de imperfecciones (inadecuada luminosidad), resaltar aspectos deseados (incrementar contraste), entre otras tareas. Una

<sup>2</sup>La imagen está almacenada en la carpeta 'laboratorio/lab04/'



herramienta ampliamente usada para esto es la aplicación de transformaciones de intensidad sobre las imágenes.

- a. Leer la imagen **'parrot.jpg'**<sup>3</sup>. Convertir la imagen a escala de grises usando **rgb2gray()**. Hallar el histograma normalizado usando **imhist()**.

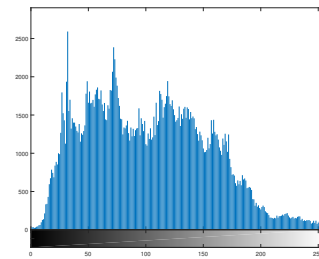
```
1 % leemos la imagen
2 I=imread('parrot.jpg');
3 % transformar a escala de grises. Se usa rgb2gray(), que asigna pesos ...
  diferentes a cada canal, buscando retener solo informacion de ...
  luminosidad de la imagen.
4 I2=rgb2gray(I);
5 % calcular el histograma
6 imhist(I2(:))
```



(a)



(b)



(c)

Figure 5: (a) Imagen original RGB, (b) Imagen representada en escala de grises y (c) histograma de la imagen en escala de grises.

- b. Aplicar dos transformaciones de intensidad sobre la imagen: transformación logarítmica y ecualización del histograma, la última usando **histeq()**. Mostrar el histograma resultante después de la aplicación de las transformaciones indicadas.

La transformación logarítmica está dada por:

$$G(x, y) = c \cdot \log(1 + f(x, y)), \quad (1)$$

donde  $c$  es una constante (en este caso se escoge  $c = 1$ ) y  $f$  la imagen normalizada entre 0 - 255.

Por otro lado, el ecualización del histograma consiste en que, dada una imagen de tamaño  $M \times N$ , con  $n_k$  pixeles para cada nivel de intensidad  $r_k$ , se realiza una transformación sobre los niveles de intensidad de la imagen, dada por

$$s_k = \frac{L-1}{MN} \sum_{j=0}^k n_j \quad (2)$$

```
1 %% TRANSFORMACION LOGARITMICA
2 % Realizamos un cast a double, para que los valores se encuentren ...
  entre 0 y 255.
```

<sup>3</sup>La imagen está almacenada en la carpeta 'laboratorio/lab04/'

```

3  f=double(I2);
4  C=1;
5  %Aplicamos la ecuacion (1)
6  G=C*log(1+f);
7  %Para mostrar la imagen entre [0 255], hallamos un factor de ...
   normalizacion.
8  factor=255/max(G(:));
9  %Multiplicamos la imagen resultante por dicho factor (comprobar que ...
   en efecto,B tiene valores entre 0 y 255).
10 B=factor*G;
11 %Se grafica la imagen resultante, y se coloca el titulo adecuado.
12 figure,imshow(B,[]);title('Transformacion Log -> G = c*log(1+f)');
13 %Realizamos un cast a uint8 para proceder a hallar el histograma. ...
   Notar que si no se realizaba el cast, imhist() solo iba a ...
   considerar valores entre 0 y 1.
14 B2=uint8(B);
15 imhist(B2)
16 %% ECUALIZACION DEL HISTOGRAMA
17 %realizamos la ecualizacion directamente a la imagen en escala de grises.
18 ee=histeq(I2);
19 figure,imshow(ee);title('Ecualizacion del histograma');

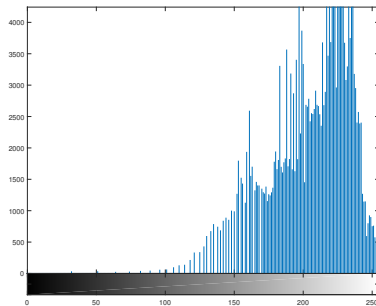
```



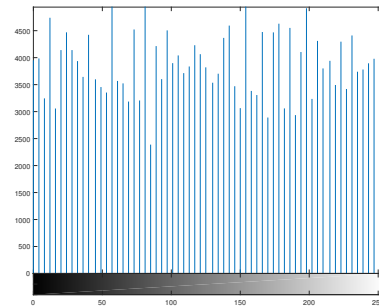
(a)



(b)



(c)



(d)

Figure 6: (a) Imagen con transformación logarítmica, (b) Imagen con ecualización de histograma, (c) Histograma de la imagen con transformación logarítmica y (d) Histograma de la imagen ecualizada.

- c. Ahora, tomar como imagen de entrada las imágenes resultantes de b) e intentar obtener nuevamente '**parrot.jpg**' (aplicar la transformación inversa). ¿En ambos casos es posible reconstruir la imagen original?

Primero, se requiere definir si es posible encontrar una función inversa de la transformación

logarítmica. Se puede probar que la función logaritmo es inyectiva, pues  $\forall x, y \in \mathbb{R}$

$$\begin{aligned} f(x) &= f(y) \\ \log(x) &= \log(y) \\ x &= y \end{aligned}$$

Es decir, a cada valor de  $x$  le corresponde un único valor de  $y$ , es una proyección 1 a 1. Entonces, se procede a hallar la transformación inversa de la transformación logarítmica:

$$\begin{aligned} G &= c \cdot \log(1 + f) \\ e^{(G/c)} &= 1 + f \\ f &= e^{(G/c)} - 1 \end{aligned} \quad (3)$$

Por otro lado, se requiere definir si es posible obtener una transformación inversa para la ecualización del histograma. De la ecuación (2), se puede verificar que el procedimiento requiere calcular el histograma acumulado de la imagen, y en base a este modificar los valores de cada nivel de intensidad. Es decir, el nivel  $j$  tiene influencia sobre el nivel  $j + 1, j + 2, \dots, j + n$ . Por ello, no es posible obtener una transformación inversa, ya que no se trata de una proyección 1 a 1.

```
1  % transformacion logaritmica inversa (segun ecuacion 3)
2  f_inv=exp(G./C)-1;
3  figure,imshow(f_inv,[]);title('Transformacion Log Inversa-> f_{inv} = ...
   exp(G/C)-1');
4
5  % Para el caso de la ecualizacion, no es posible obtener nuevamente ...
   la imagen original, por el motivo mencionado anteriormente.
```



Figure 7: Imagen de la aplicación de la transformación logarítmica inversa.

- d. Aplicar bit-plane slicing a la imagen **'parrot.jpg'** en escala de grises y graficar los resultados. Analizar en qué bits se tiene mayor información.

Bit-plane-slicing es una técnica usualmente usada para comprimir imágenes. Así, involucra realizar una cuantización de la imagen a tratar. La estrategia para resolver la pregunta será en primer lugar expresar la imagen en escala de grises y, dependiendo de el número de bits a considerar, cuantizarla. Para ello, se divide entre  $2^n$  cuando se desea trabajar con  $n$  bits, y se cuantiza usando **ceil()**. Esto ocasionará que el rango original  $[0 - 255]$  sea modificado al final. Para restaurarlo, se vuelve a multiplicar por  $2^n$ .



```

1  % Se hace un cast a tipo double a la imagen en escala de grises.
2  I22=double(I2);
3  [r,c]= size(I2);
4  % Se trabaja para los 7 bits de intensidad. La imagen es dividida por ...
   2^n, donde n es el numero de bits a mostrar. Luego, para ...
   efectivamente emplear solo n bits, se emplea ceil(), cuya funcion ...
   es cuantizar la imagen. Luego, para recuperar el rango de ...
   valores original, se vuelve a multiplicar por 2^n.
5  for m = 1:r
6      for n = 1:c
7          b(m,n,1)=ceil(I22(m,n)/power(2,7))*power(2,7);
8          b(m,n,2)=ceil(I22(m,n)/power(2,6))*power(2,6);
9          b(m,n,3)=ceil(I22(m,n)/power(2,5))*power(2,5);
10         b(m,n,4)=ceil(I22(m,n)/power(2,4))*power(2,4);
11         b(m,n,5)=ceil(I22(m,n)/power(2,3))*power(2,3);
12         b(m,n,6)=ceil(I22(m,n)/power(2,2))*power(2,2);
13         b(m,n,7)=ceil(I22(m,n)/power(2,1))*power(2,1);
14     end
15 end
16 % Se grafica las 8 imagenes en una misma ventana. Se hace un cast a ...
   uint8 para su correcta visualizacion.
17 for i=7:-1:1
18     subplot(2,4,i+1),imshow(uint8(b(:,:,i)));
19     plot_header = sprintf('bit - %d',i);
20     title(plot_header);
21 end
22 subplot(2,4,1),imshow(I2); plot_header = sprintf('Imagen original');
23 title(plot_header);

```

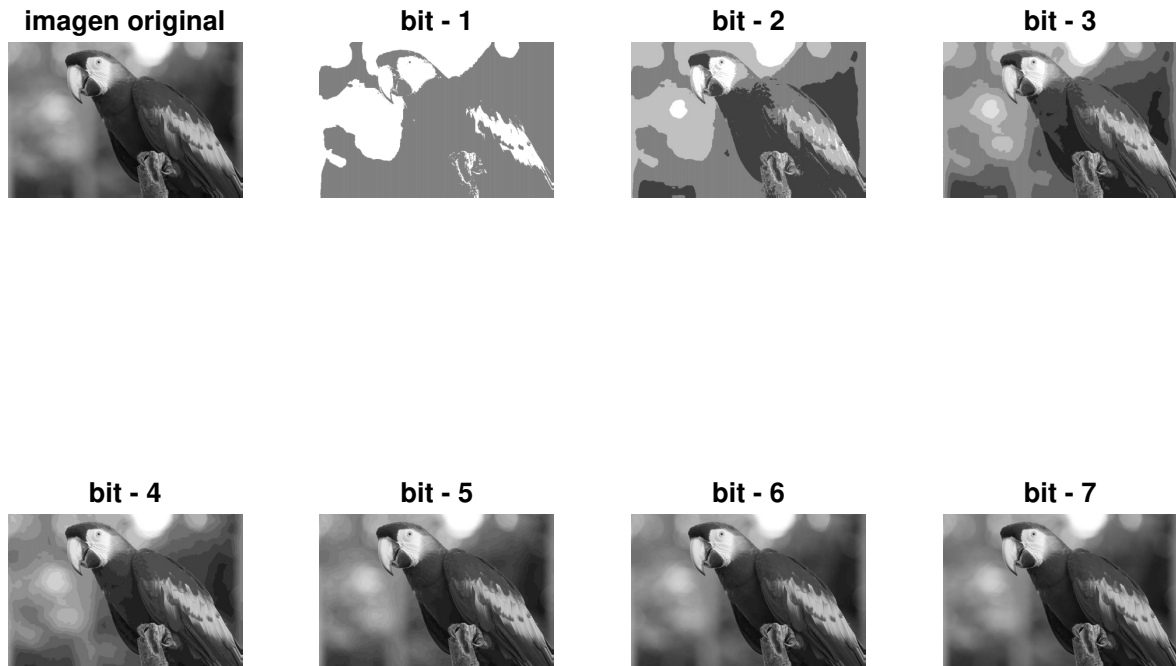


Figure 8: Imagen bit-plane.

4. Dado el archivo **'flowers.mat'**<sup>4</sup>, se pide:
  - a. Cargar el archivo con **load()**. En una misma ventana, graficar el archivo y cada uno de sus canales. Al visualizarla, notará que no es una imagen en RGB, pues está en el espacio de color HSV. Convertirla a RGB usando **hsv2rgb()**. Luego, hacer un cast a uint8 usando **cast()**. Comparar visualmente la imagen en el espacio de color HSV y la imagen en RGB. ¿Qué característica de la imagen fue atenuada al realizar la conversión?
  - b. Notar que el contraste en la imagen es bajo. Analizar el histograma en cada capa e indicar a qué se debe. Usar **imhist()**. Posteriormente, para corregir esto, aplicar una transformación gamma a cada capa, la cual está definida por

$$I_{out} = (I_{in})^\gamma \frac{1}{I_{max}^{\gamma-1}}$$

para los valores de  $\gamma=0.1$ , 1 y 4.  $I_{in}$  hace referencia cada capa de la imagen RGB de entrada, e  $I_{max}$  al máximo valor de dicha capa. Luego, para cada valor de  $\gamma$ , graficar en una misma ventana la imagen antes y después de aplicar la transformación en el espacio de color RGB.

---

<sup>4</sup>La imagen está almacenada en la carpeta 'laboratorio/lab04/'

**Nota:** Si se obtiene  $I_{out}R$  (transformación de la capa roja),  $I_{out}G$  (transformación de la capa verde) Y  $I_{out}B$  (transformación de la capa azul), asignar cada una a cada capa de  $I_{enhanced}$ .

- c. Calcular el histograma para los canales R,G y B por cada valor de gamma usado. ¿Cómo ha variado el histograma con respecto al hallado en el ítem anterior? ¿Se percibe alguna tendencia en el histograma a medida que gamma aumenta de valor?
- d. En la imagen original se quiere transformar a escala de grises aquellas regiones que no sean de color rojo. Para ello, analizar la intensidad en cada canal y usar la siguiente fórmula para transformar a escala de grises:

$$I = 0.3 \cdot I_R + 0.59 \cdot I_G + 0.11 \cdot I_B$$

$$I_{RGB}(:, :, i) = I, \quad i = 1, 2, 3$$

donde  $I_R$ ,  $I_G$ ,  $I_B$  corresponden a cada canal e I corresponde a la imagen en escala de grises con un color RGB resaltado.

**Nota:** Considerar los umbrales 100 para el canal B y 80 para los canales RG. Por ejemplo, si se quiere resaltar el color azul, la condición debe considerar que  $B < 100$  y  $RG > 80$ ; y en caso uno de ellos se cumpla, debe ejecutarse la ecuación.