



Manual de Prácticas Procesamiento Digital de Imágenes

División de Ingeniería Eléctrica


Laboratorio de Cómputo para el
Procesamiento de Señales

Manejo Básico de Imágenes

No. de práctica: 0

Nombre completo de los alumnos		Firma
Gutiérrez Castillo Óscar Joya Venegas Jehosua Alan Nolasco Fuentes Juan Carlos		
No dé cuenta:	Fecha de elaboración: 07/10/2020	Grupo: 1

Elaborado por:	Revisado por:	Autorizado por:	Vigente desde:
Adán Martínez Sixto			

	<h1 style="text-align: center;">Manual de Prácticas Procesamiento Digital de Imágenes</h1>	
<p style="text-align: center;">División de Ingeniería Eléctrica</p>	<p style="text-align: center;">Laboratorio de Cómputo para el Procesamiento de Señales</p>	

Objetivo

Al término de esta práctica, el alumno conocerá las distintas formas de desplegar una imagen en distintos formatos, así como algunas manipulaciones básicas, para el procesamiento de imágenes.

Introducción

Matlab

MatLab[®] incluye un conjunto de funciones (algoritmos) y herramientas gráficas enfocadas al procesamiento, análisis y visualización de imágenes, este conjunto es denominado toolbox de imágenes. Muchas de las funciones del toolbox de imágenes se encuentran escritas en el lenguaje de MatLab[®], lo que permite inspeccionar los algoritmos usados. En esta práctica se tocarán los puntos de entrada, salida y visualización de imágenes en diferentes formatos usando las funciones del toolbox.

Representación de los Tipos de imágenes en MatLab[®]

MatLab[®] soporta varios tipos de imágenes, las cuales se enumeran a continuación:

- imágenes Indexadas:

Consisten de una matriz de datos y una paleta de colores. La paleta de colores es un arreglo de m filas por 3 columnas la cual contiene valores de punto flotante en el rango de $[0,1]$. Cada fila de la paleta especifica las componentes roja, verde y azul para un color único. El color de cada pixel es determinado usando el correspondiente valor en la matriz de datos como un índice dentro de la paleta asignada. El valor 1 apunta a la primer fila de la paleta, el valor 2 a la segunda fila, etc.

- Imágenes de intensidad:

Consisten de una matriz de datos, cuyos valores representan intensidades dentro de algún rango, cada elemento en la matriz corresponde a un pixel en la imagen representando niveles de gris, donde la intensidad 0 usualmente representa al negro y la intensidad 1, 255 o 65536 usualmente representa total intensidad, o blanco.

- Imágenes RGB o a color:

Una imagen RGB, a veces es referida como una imagen de color verdadero, en MatLab[®] es almacenada como una matriz de $m \times n \times 3$ (m filas por n columnas y una profundidad de 3) que define las componentes roja, verde y azul para cada pixel individual. Las imágenes RGB no usan paleta de colores, el color de cada pixel es determinado por la combinación de las intensidades roja, verde y azul almacenadas en cada plano de color en la ubicación del pixel. Los formatos gráficos almacenan las imágenes RGB como imágenes de 24 bits, donde las componentes roja, verde y azul son cada una de 8 bits, esto da un potencial de 16 millones de colores. Un pixel cuyas componentes de color

	<h1 style="text-align: center;">Manual de Prácticas Procesamiento Digital de Imágenes</h1>	
<p style="text-align: center;">División de Ingeniería Eléctrica</p>	<p style="text-align: center;">Laboratorio de Cómputo para el Procesamiento de Señales</p>	

son (0,0,0) es desplegado como negro y un pixel con las componentes (1,1,1) es desplegado como blanco.

La imagen

Una imagen es la representación en dos dimensiones, que contiene la información de color (intensidad) así como el diseño, esta información es guardada en los elementos más simples llamados pixeles o pel (del inglés picture element). En el campo de la medicina, una imagen es el resultado de la respuesta de un transductor del algún equipo de imagenología médica ya sea un equipo de Rayos X, de mastografía, de tomografía etc. En cuanto al diseño de la imagen contiene una disposición de matriz de m columnas por n filas (m, n), donde se encuentran ordenadamente los pixeles, el origen se ubica en la esquina superior izquierda donde se ubica el elemento (0,0). La imagen puede contener un solo canal, esto es con aquellas que son a escala de grises o blanco y negro, así mismo múltiples canales, tal es el caso de las imágenes a color RGB o YUV.

Resolución de una imagen

Una imagen puede ser juzgada por la calidad que posee al momento de ser obtenida. Para sistemas de imagenología médica, estas imágenes deben de ser diagnósticamente útiles para detectar una enfermedad. La resolución de la fuente de imágenes es definida por medio de tres cantidades:

- Resolución Espacial: Se refiere a la cantidad de pixeles en columnas (C) y renglones (R) para representar el espacio visual capturado en una imagen, se relaciona con el muestreo de la señal de una imagen. Generalmente se expresa de la forma (C x R), por ejemplo (600x 400, 1366 x 768, etc.)
- Resolución Temporal: Cuando se tiene imágenes continuas como en el caso del video, la resolución temporal determina la cantidad de cuadros por una cantidad de tiempo, en los archivos de video es común utilizar los cuadros por segundo o fps 1 .
- Resolución de bit o profundidad de bit: Define el número posible de valores de intensidad/color que un píxel puede tener y se relaciona con la cuantización de la información de una imagen. En el caso de las imágenes binarias, solo disponemos de 2 colores, blanco y negro. Para una imagen de escala de grises puede tener 2 8 niveles de escala de grises. Generalmente se representa con el número de bits requeridos para almacenar el nivel de cuantización dado.

Formato de imagen

Existen por lo menos 100 formatos de imagen. Algunos de estos formatos tales como jpeg o png, son utilizados para imágenes fotográficas. Por otro lado, los formatos DICOM o NiftI, son utilizados en imagenología médica. Otros formatos como el TIFF, ics, ims, entre otros son usados para imágenes de microscopio. Sin embargo, no nos debemos preocupar por el formato de imagen, así

	<h1 style="text-align: center;">Manual de Prácticas Procesamiento Digital de Imágenes</h1>	
<p style="text-align: center;">División de Ingeniería Eléctrica</p>	<p style="text-align: center;">Laboratorio de Cómputo para el Procesamiento de Señales</p>	

como de sus metadatos. Los módulos y paquetes de Python tienen la capacidad de leerlos y extraer los datos necesarios que le serán útiles al usuario (por ejemplo, tamaño de la imagen, tipo o modo, tipo de dato). A continuación, definiremos algunos formatos más comunes de imágenes digitales.

- **JPEG:** es el acrónimo inglés de Joint Photographic Experts Group, su formato es representado como .jpg o .jpeg. Es uno de los más populares debido a su habilidad para comprimir significativamente los datos con una pérdida visual mínima. Es un formato con pérdida, que comprime los datos usando la Transformada de Coseno Discreta (TCD). Este formato no es adecuado para almacenar imágenes que contienen estructuras finas como líneas, curvas etc.
- **TIFF:** es el acrónimo inglés de Tagged Image Format. Su extensión es .tif o .tiff, fue creado en la década de los 80's para almacenar y codificar documentos escaneados. Originalmente fue desarrollado para datos de un solo bit, pero hoy en día el estándar permite almacenar 16 bits incluso datos de punto flotante.
- **DICOM:** es el acrónimo inglés de Digital Imaging and Communication in Medicine. Es un formato estándar para codificar y transmitir Imágenes de Resonancia Magnética (MRI) e Imágenes de Tomografía Computarizada. Este formato almacena información junto con otros datos entre los que destacan; detalles del paciente, parámetros de adquisición, hora de creación, modelo del equipo, información del hospital etc. DICOM es usado en varias especialidades de medicina como Radiología, Neurología, Cirugía, Cardiología, Oncología entre otros.

Tipos de imagen


El tipo de imagen elegido para cierta aplicación involucra la información que se quiere obtener, así como su resolución de bit. Para esto se tienen diferentes tipos de imagen que se enlistan a continuación:

Las imágenes de un solo canal son aquellas en las que cada píxel es representado por un solo valor, los podemos dividir en dos tipos:

- **Imágenes binarias:** o también llamadas monocromáticas, el valor de un píxel 0 se representa color negro y el píxel de valor 1 representa el color blanco.
- **Imágenes de escala de grises:** cada píxel puede ser representado con 8 bits y tiene valores de dentro del rango de 0-255

Las imágenes multicanal son representados por una tupla de valores. Los podemos dividir en los siguientes tipos.

- **Imágenes RGB:** son imágenes en arreglos de 3-D en la que cada píxel es representado por una tupla de valores (r,g,b), que representan a los colores rojo, verde y azul respectivamente. Comúnmente, se almacenan como una secuencia de enteros en orden de canal sucesivo.

	<h1 style="text-align: center;">Manual de Prácticas Procesamiento Digital de Imágenes</h1>	
<p style="text-align: center;">División de Ingeniería Eléctrica</p>	<p style="text-align: center;">Laboratorio de Cómputo para el Procesamiento de Señales</p>	

- **Imágenes HSV:** son imágenes en las que cada píxel es representado por una tupla de valores (h,s,v) que representan al color, saturación y brillo respectivamente. El modelo HSV describe colores de manera similar a como el ojo humano tiende a percibir colores.
- **Imágenes de RGBA:** cada píxel es representado por una tupla de cuatro valores (r,g,b,a) el ultimo valor representa la transparencia.

Imágenes de punto flotante: Estas imágenes almacenan números de punto flotante que, dentro de un rango determinado definido por la precisión de punto flotante de la resolución de bits de la imagen, representa la intensidad. Las imágenes de punto flotante pueden ser almacenadas en formato TIFF o DICOM.

Manipulación de Imágenes Matlab

Abrir imágenes con MatLab. Una imagen digital es un mosaico de elementos llamados pixeles, dichos elementos se encuentran almacenados en archivos digitales en un determinado formato definido por el tipo de imagen. Se consideran las imágenes que estan almacenadas en un archivo con un formato determinado, por ejemplo: BMP, JPG, TIF, PNG, GIF, etc. MatLab[®] posee la función `imread` encargada de abrir imágenes de diferentes formatos dentro del workspace, su sintaxis básica es la siguiente:

```
A=imread('nombre_imagen', 'tipo_imagen')
```

donde:

A.- Es el nombre de la variable donde se guardan los valores de los pixeles de la imagen.


nombre_imagen.- Especifica el nombre completo del archivo que contiene a la imagen, puede incluir la ruta relativa o absoluta del fichero.

tipo_imagen.- Indica el formato de la imagen, por ejemplo: bmp, gif, jpg, png, tiff, etc. Se usa para forzar la lectura de un formato en particular sin importar la extensión del archivo.

Para el caso de las imágenes indexadas MatLab regresa dos valores, uno para la matriz de valores y otro para la paleta o mapa de colores asociado:

```
[A, mapa]=imread('nombre_imagen', 'tipo_imagen')
```

Visualizar imágenes con MatLab. Para desplegar en pantalla los datos contenidos en una matriz se utiliza el comando `image`, además para interpretar de manera correcta los datos para el caso de imágenes de intensidad e indexadas se usa el comando `colormap`, a continuación se muestra la sintaxis de ambas funciones:

	<h1 style="text-align: center;">Manual de Prácticas Procesamiento Digital de Imágenes</h1>	
<p style="text-align: center;">División de Ingeniería Eléctrica</p>	<p style="text-align: center;">Laboratorio de Cómputo para el Procesamiento de Señales</p>	

figure(1); image(A); colormap(mapa)

donde:

1- Indica el número de ventana donde se visualizará la imagen

mapa. Es el arreglo de m filas por 3 columnas que contiene la paleta de colores que interpretará los colores de la imagen.

Ejercicios

En el folder de imágenes se le proporcionan otro tipo de imágenes que no son médicas, puede usted utilizarlas para el desarrollo de la práctica. Desarrolla un script para leer y desplegar cada imagen. De todas las imágenes que lea imprimir el tipo de imagen, el tamaño y el tipo de dato

1. Abrir y visualizar la imagen RGB “magriclonRGB.jpg” y cada una de sus componentes de color por separado usando la paleta de colores roja, verde y azul.

Código:

```
rgbImage = imread('magriclonRGB.jpg','jpg' ); % Read image
% Extract color channels.
redChannel = rgbImage(:,:,1); % Red channel
greenChannel = rgbImage(:,:,2); % Green channel
blueChannel = rgbImage(:,:,3); % Blue channel

% Create an all black channel.
allBlack = zeros(size(rgbImage, 1), size(rgbImage, 2), 'uint8');

% Create color versions of the individual color channels.
just_red = cat(3, redChannel, allBlack, allBlack);
just_green = cat(3, allBlack, greenChannel, allBlack);
just_blue = cat(3, allBlack, allBlack, blueChannel);

% Display them all.
subplot(3, 3, 2);
imshow(rgbImage);
fontSize = 20;
title('Imagen Original', 'FontSize', fontSize)

subplot(3, 3, 4);
imshow(just_red);
title('Canal rojo', 'FontSize', fontSize)

subplot(3, 3, 5);
```



Manual de Prácticas Procesamiento Digital de Imágenes

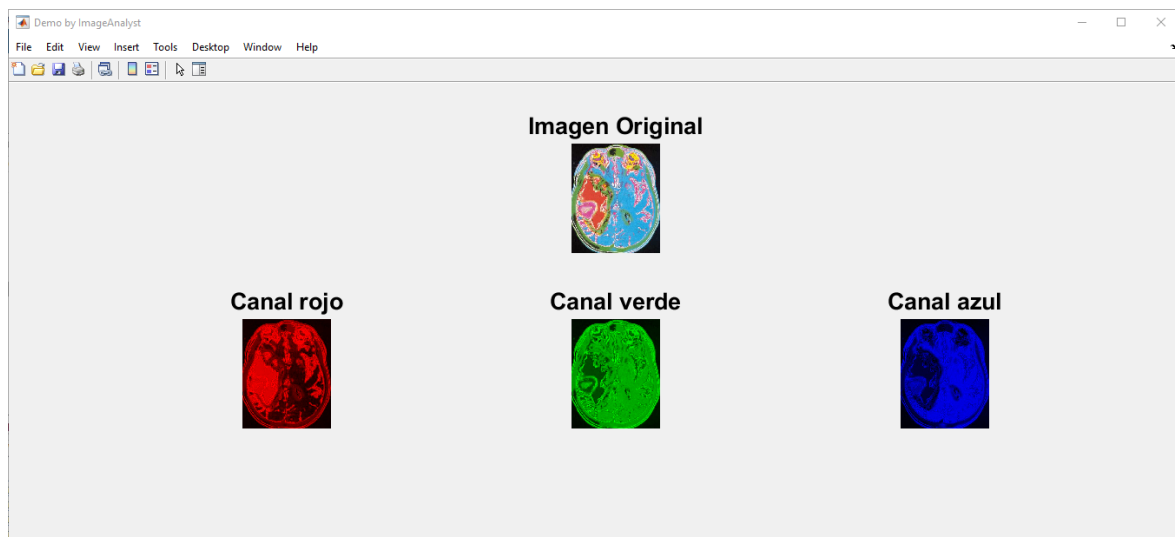
División de Ingeniería Eléctrica

Laboratorio de Cómputo para el
Procesamiento de Señales

```
imshow(just_green)
title('Canal verde', 'FontSize', fontSize)

subplot(3, 3, 6);
imshow(just_blue);
title('Canal azul', 'FontSize', fontSize)

% Set up figure properties:
% Enlarge figure to full screen.
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0, 1, 1]);
% Get rid of tool bar and pulldown menus that are along top of
figure.
% set(gcf, 'Toolbar', 'none', 'Menu', 'none');
% Give a name to the title bar.
set(gcf, 'Name', 'Demo by ImageAnalyst', 'NumberTitle', 'Off')
```



2. Abrir y desplegar la imagen RGB “retinaRGB.jpg” y convertir cambiando el espacio de color de:
 - 2.1. RGB a Escala de grises
 - 2.2. RGB a YUV
 - 2.3. RGB a HSV

Código

```
RGB = imread('retinaRGB.jpg', 'jpg');

escalaGrises = rgb2gray(RGB);

hsv = rgb2hsv(RGB);

subplot(2, 3, 2);
```




Manual de Prácticas Procesamiento Digital de Imágenes

División de Ingeniería Eléctrica

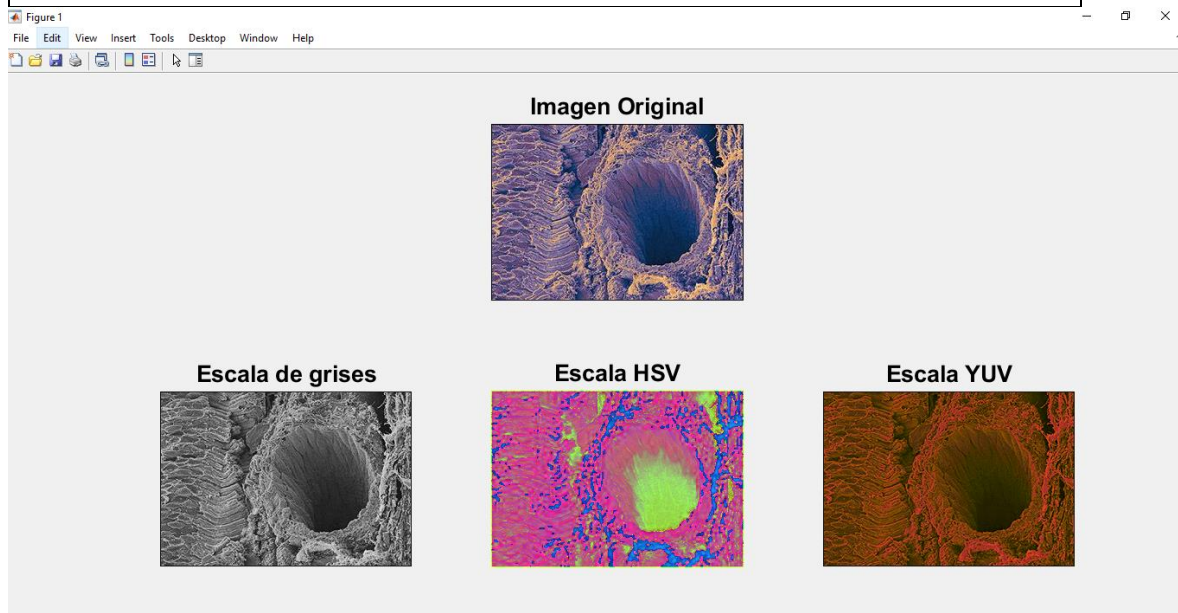
Laboratorio de Cómputo para el
Procesamiento de Señales

```
imshow( RGB );
fontSize=20;
title('Imagen Original','FontSize',fontSize)

subplot(2,3,4);
imshow(escalaGris);
title('Escala de grises','FontSize',fontSize)

subplot(2,3,5);
imshow(hsv);
title('Escala HSV','FontSize',fontSize)

R=RGB(:,:,1);
G=RGB(:,:,2);
B=RGB(:,:,3);
Y = 0.299 * R + 0.587 * G + 0.114 * B;
U = -0.14713 * R - 0.28886 * G + 0.436 * B;
V = 0.615 * R - 0.51499 * G - 0.10001 * B;
YUV = cat(3,Y,U,V);
subplot(2,3,6);
imshow(YUV);
title('Escala YUV','FontSize',fontSize)
```



3. Leer y desplegar la imagen del archivo “corte.bmp”, y obtener una imagen cortada a un tercio de su tamaño, realice el corte (crop) en la esquina inferior derecha. Realice otra subimagen donde usted desee.

Código:

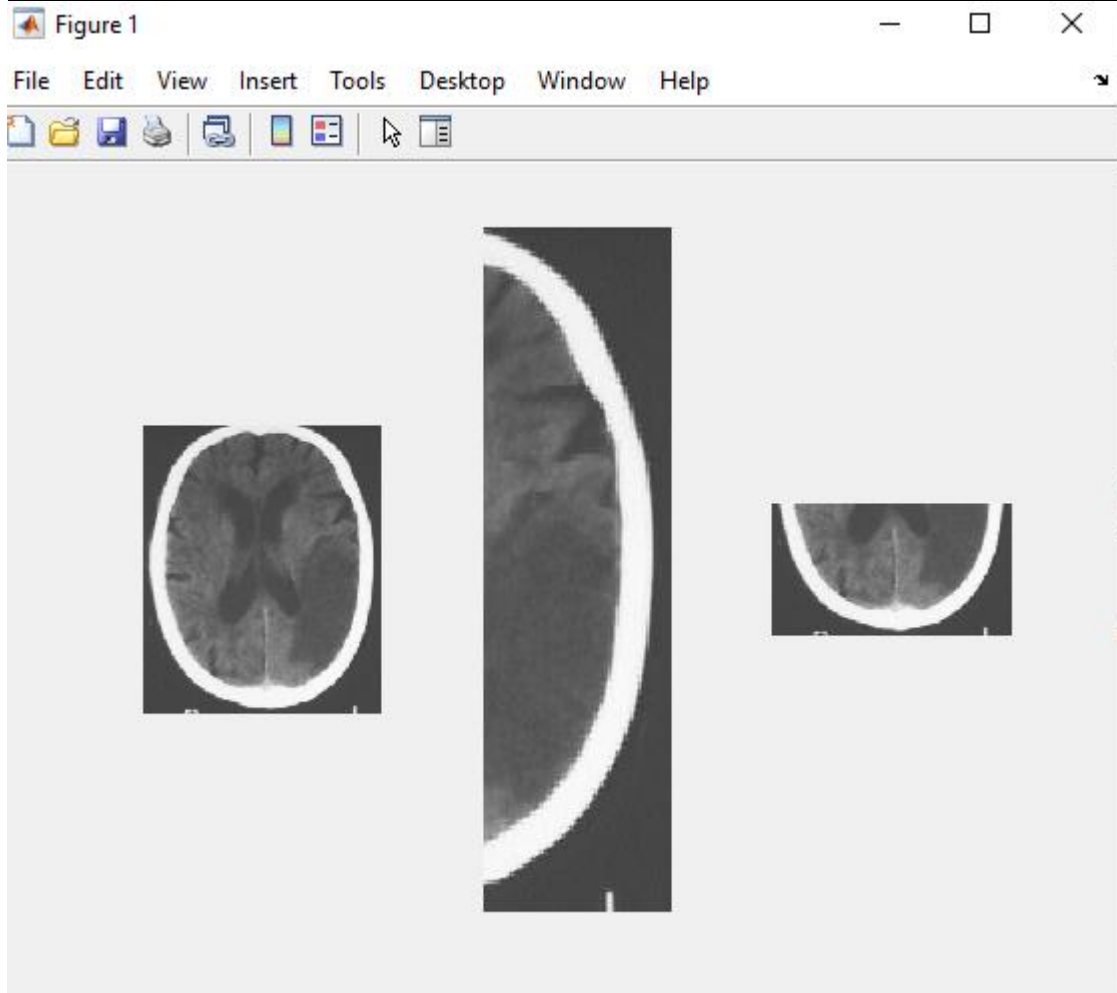


Manual de Prácticas Procesamiento Digital de Imágenes

División de Ingeniería Eléctrica

Laboratorio de Cómputo para el
Procesamiento de Señales

```
originalImage = imread("corte.bmp");  
[rows, columns, numberOfColorChannels] = size(originalImage);  
  
cropImage1 = imcrop(originalImage,[275,0,413,513]);  
cropImage2 = imcrop(originalImage,[0,275,513,413]);  
  
subplot(1,3,1);  
imshow(originalImage);  
  
subplot(1,3,2);  
imshow(cropImage1);  
  
subplot(1,3,3);  
imshow(cropImage2);
```





Manual de Prácticas Procesamiento Digital de Imágenes

División de Ingeniería Eléctrica

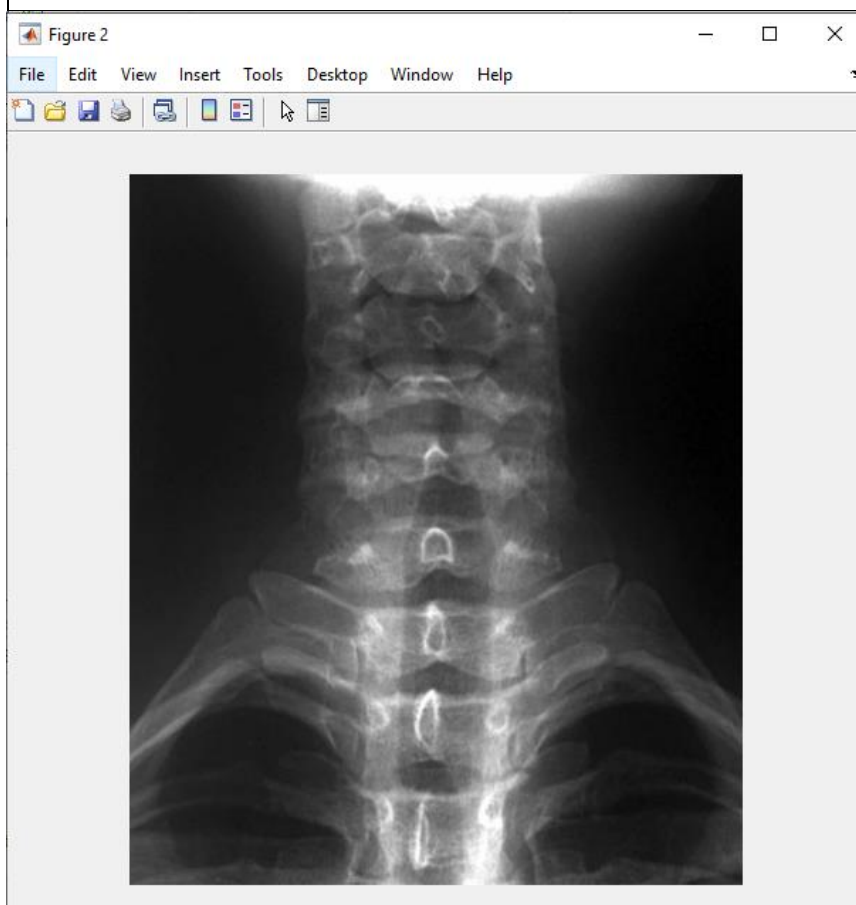
Laboratorio de Cómputo para el
Procesamiento de Señales

4. Leer la imagen del archivo “rx cerv.pcx” y desplegarla usando primero el comando image y posteriormente el comando imshow.

Código:

```
rx cerv = imread('rx cerv.pcx','pcx' ); % Read image
%Showing the image with image
figure(1);
image(rx cerv);

%Showing the image with imshow
figure(2);
imshow(rx cerv);
```



5. Leer y desplegar la imagen “abdomen.png” usando el comando image y usando posteriormente el comando imagesc.

Código:

```
subplot(1,2,1);
C = imread("abdomen.png");
```

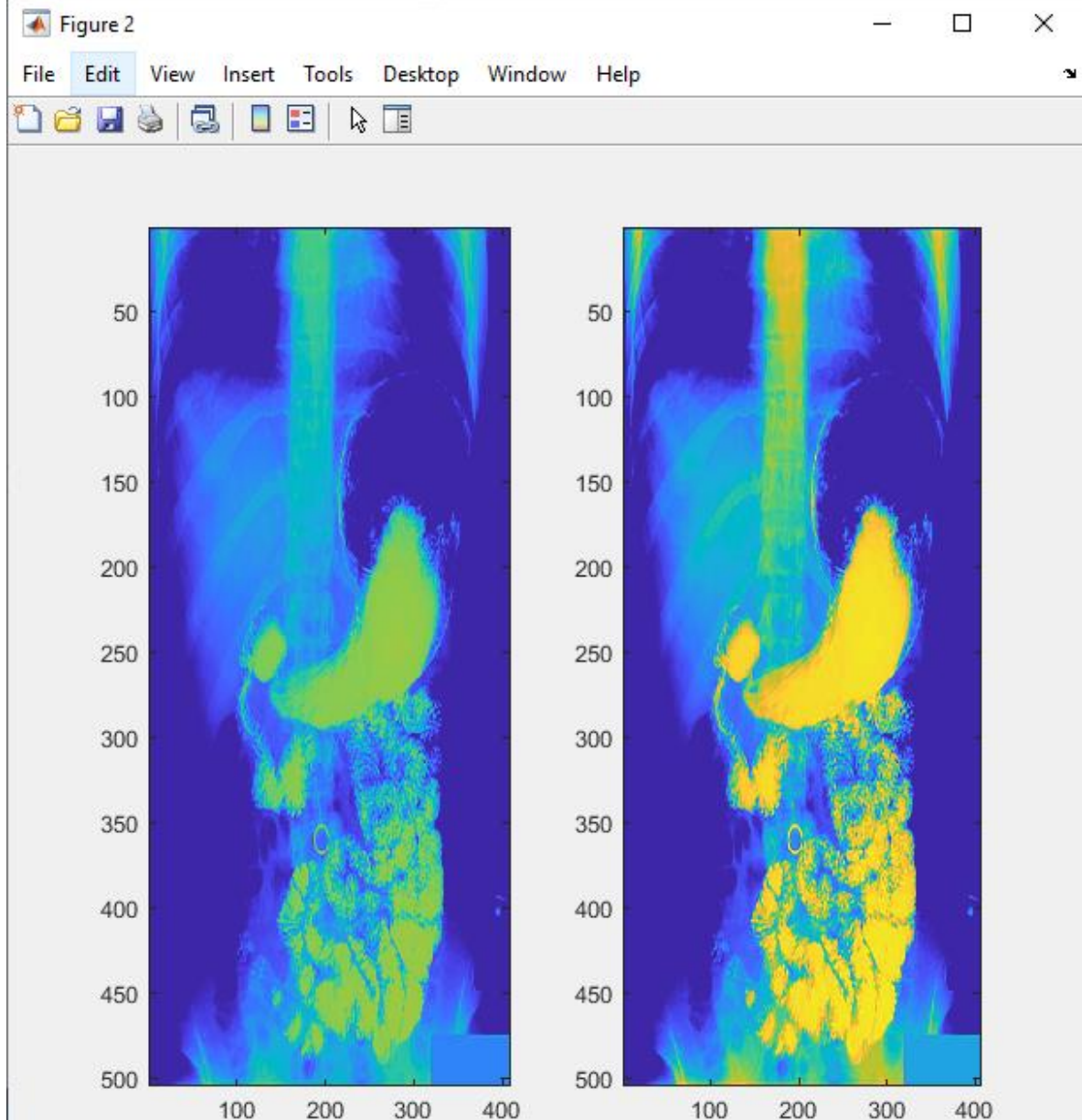


Manual de Prácticas Procesamiento Digital de Imágenes

División de Ingeniería Eléctrica

Laboratorio de Cómputo para el
Procesamiento de Señales

```
image(C);  
  
subplot(1,2,2);  
C = imread("abdomen.png");  
imagesc(C);
```



	<h1 style="text-align: center;">Manual de Prácticas Procesamiento Digital de Imágenes</h1>	
<p style="text-align: center;">División de Ingeniería Eléctrica</p>	<p style="text-align: center;">Laboratorio de Cómputo para el Procesamiento de Señales</p>	

6. Despliega la paleta de colores de RGB por separado

Código:

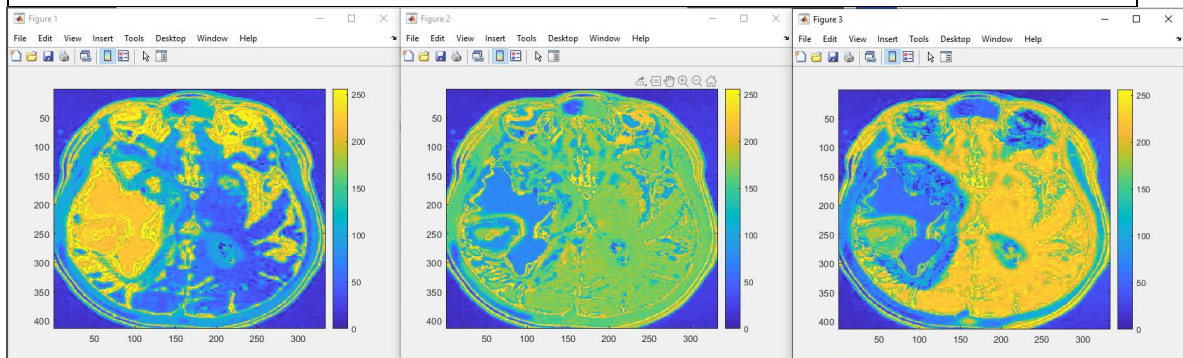
```
magriclonRGB = imread('magriclonRGB.jpg','jpg' ); % Read image

% Extract color channels.
redChannel = rgbImage(:,:,1); % Red channel
greenChannel = rgbImage(:,:,2); % Green channel
blueChannel = rgbImage(:,:,3); % Blue channel

%Showing red colorbar
figure(1)
image(redChannel)
colorbar

%Showing green colorbar
figure(2)
image(greenChannel)
colorbar

%Showing blue colorbar
figure(3)
image(blueChannel)
colorbar
```



7. Visualizar la imagen “intestinoRGB.jpg” con diferentes paletas de colores predefinidas de MatLab.

Código:

```
%Reading image
abdomen=imread("abdomen.png","png")

%Colormap jet
figure(1);
image(abdomen);
colormap("jet");
```



Manual de Prácticas Procesamiento Digital de Imágenes

División de Ingeniería Eléctrica

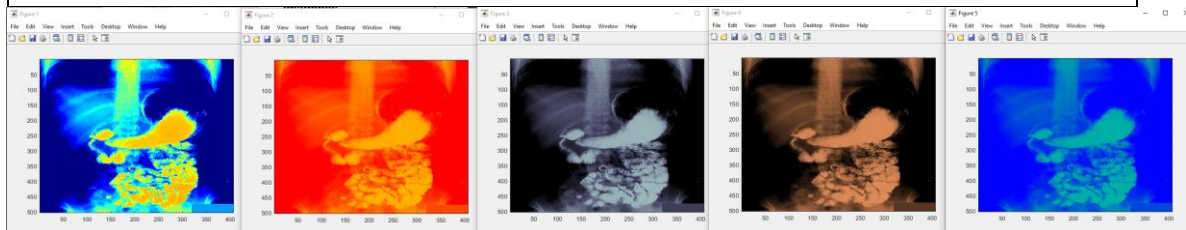
Laboratorio de Cómputo para el
Procesamiento de Señales

```
%Colormap autumn
figure(2);
image(abdomen);
colormap("autumn");

%Colormap bone
figure(3);
image(abdomen);
colormap("bone");

%Colormap copper
figure(4);
image(abdomen);
colormap("copper");

%Colormap winter
figure(5);
image(abdomen);
colormap("winter");
```



8. Visualizar la imagen indexada "cta_scan_index.bmp". Rota la imagen a 45, 90 y 180 grados
Código:

```
cta_image = imread("cta_scan_index.bmp") ;
rotate45 = imrotate(cta_image,45) ;
rotate90 = imrotate(cta_image,90);
rotate180 = imrotate(cta_image,180);

subplot(2,2,1);
imshow(cta_image);

subplot(2,2,2);
imshow(rotate45);

subplot(2,2,3);
imshow(rotate90);

subplot(2,2,4);
```

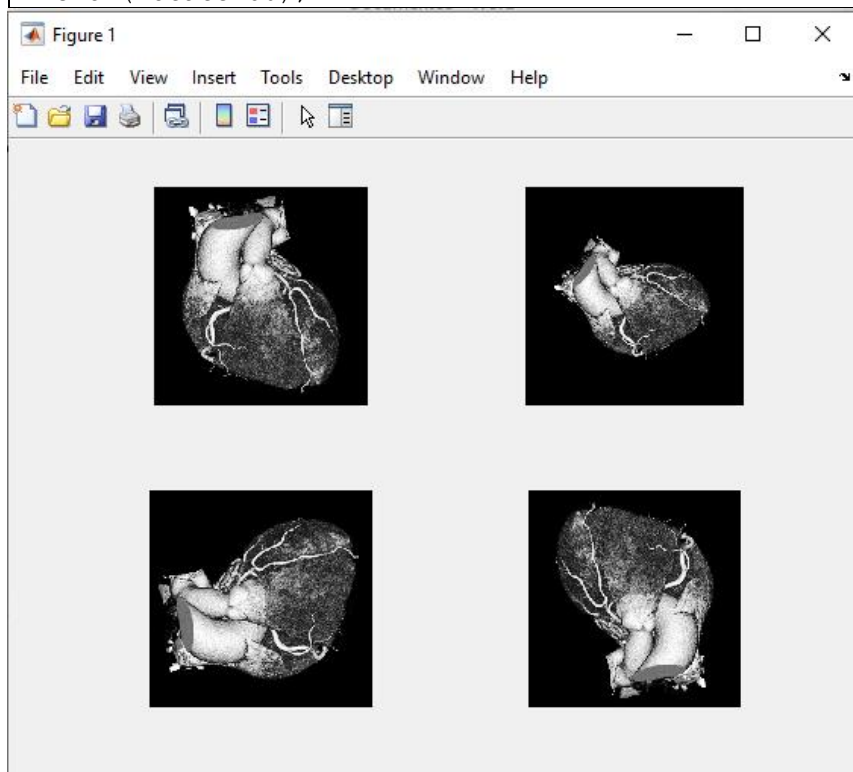


Manual de Prácticas Procesamiento Digital de Imágenes

División de Ingeniería Eléctrica

Laboratorio de Cómputo para el
Procesamiento de Señales

```
imshow(rotate180);
```



9. Convierte la imagen mri.jpg a escala de grises, y guárdala en formato .tiff
Código:

```
RGB = imread('mri.jpg','jpg');  
  
escalaGrises = rgb2gray(RGB);  
  
imwrite(escalaGrises, 'mri.tiff');
```

	mri		07/10/2020 09:33 p. m.	Archivo TIFF	83 KB
	mri		15/10/2019 05:39 p. m.	Archivo JPG	17 KB

10. Imagen DICOM ó Imagen satelital:

Si usted decide usar DICOM: Realiza un script que lea las imágenes de la carpeta DICOM y muestra en una misma ventana las 280 imágenes de tal manera que se pueda simular el movimiento del corazón. Utiliza ciclos for y el comando pause, para leer la ruta completa de la carpeta.

Si usted decide usar una imagen satelital:

- 10.1. Lea la imagen satelital. Y enumere los componentes que tiene la imagen, así como la resolución que manejan. Despliegue dos de los componentes.



Manual de Prácticas Procesamiento Digital de Imágenes

División de Ingeniería Eléctrica

Laboratorio de Cómputo para el
Procesamiento de Señales

- 10.2. Reajuste el tamaño de la imagen disminuyendo 3 veces el tamaño original.
Despliegue
Código:

```
for i = 1:280
    if (i < 10)
        imagen = dicomread("IM-0001-000"+i+".dcm");
    elseif (i > 9 && i < 99)
        imagen = dicomread("IM-0001-00"+i+".dcm");
    else
        imagen = dicomread("IM-0001-0"+i+".dcm");
    end
    imshow(imagen, 'DisplayRange', [])
    pause(0.01)
end
```

