MATLAB Assignment 1

Quetion 1

The portable graymap format (PGM) grayscale images used in this assignment were retrieved from http://sun.aei.polsl.pl/~rstaros/mednat/index.html. This is a dataset of medical grayscale images for evaluating image compression algorithms. The set was prepared as a part of the research project Nr 4 T11C 032 24, which was fully supported by the Grant of the Polish National Research Committee (KBN).

Read 4 Computed Tomography (CT) image file and store as array of structs

```
close all; clear; clc;

% *Load 1st image filename, width, and height into struct myIMG*
img1 = imfinfo('ct_3030.pgm');
f = img1.Filename;
w = img1.Width;
h = img1.Height;
myImg(1).imagename = f;
myImg(1).width = w;
myImg(1).height = h;

% *Load 2nd image filename, width, and height into struct myIMG*
img2 = imfinfo('ct_3071.pgm');
f = img2.Filename;
w = img2.Width;
h = img2.Height;
myImg(2).imagename = f;
myImg(2).width = w;
myImg(2).height = h;

% *Load 3rd image filename, width, and height into struct myIMG*
img3 = imfinfo('ct_4006.pgm');
f = img3.Filename;
w = img3.Width;
h = img3.Height;
myImg(3).imagename = f;
myImg(3).width = w;
myImg(3).height = h;

% *Load 4th image filename, width, and height into struct myIMG
img4 = imfinfo('ct_4087.pgm');
f = img4.Filename;
w = img4.Width;
h = img4.Height;
myImg(4).imagename = f;
myImg(4).width = w;
myImg(4).height = h;
```

Save array of structs as .mat file

```
save imagesFileInfo.mat myImg;     % imagesFileInfo.mat
```

Published with MATLAB® R2015b

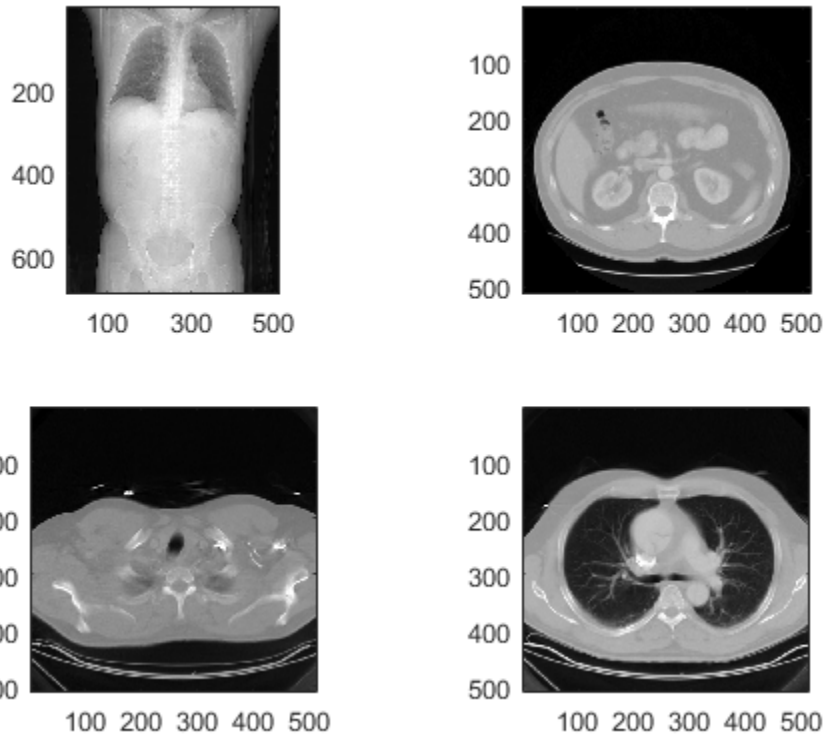
Quetion 2

Load the .MAT file created in Question 1

```
close all; clear; clc;
load('imagesFileInfo.mat')
A = myImg(1).imagename;
B = myImg(2).imagename;
C = myImg(3).imagename;
D = myImg(4).imagename;
img1 = imread(A);
img2 = imread(B);
img3 = imread(C);
img4 = imread(D);
```

Display the four images in a 2x2 arrangement

```
subplot(2,2,1), subimage(img1)
subplot(2,2,2), subimage(img2)
subplot(2,2,3), subimage(img3)
subplot(2,2,4), subimage(img4)
```



Published with MATLAB® R2015b

Question 3

Read, open, and display an image file

```
close all;clear; clc;
filename = uigetfile({'*.jpg;*.tif;*.png;*.gif;*.pgm','All Image
 Files';...
          '*.*','All Files' },'Pick an Image File');
imshow(filename)
```

Report ColorType of image file

```
info = imfinfo(filename);
type = info.ColorType;
stringType = sprintf('The type is %s. \n', type);
disp(stringType)
```

```
The type is grayscale.
```

Convert image to grayscale if needed.

```
grayImage = imread(filename);
```

Get the dimensions of the image. numberOfColorBands should be = 1.

```
[rows, columns, numberOfColorChannels] = size(grayImage);
if numberOfColorChannels > 1
```

Image is not grayscale Convert it to grayscale by taking only the green channel.

```
 filename = grayImage(:, :, 2); % Take green channel.
    disp('Converting to Grayscale')

end
```

Display maximum gray level values

```
maxGrayLevel = max(filename(:));
intMax = sprintf('The maximum gray level is %d. \n', maxGrayLevel);
disp(intMax)
```

```
The maximum gray level is 116.
```

Display minimum gray level values

```
minGrayLevel = min(filename(:));
intMin = sprintf('The minimum gray level is %d. \n', minGrayLevel);
disp(intMin)
```

```
The minimum gray level is 46.
```

Display average gray level values

```
avgGrayLevel = mean(filename(:));
floatAvg = sprintf('The average gray level is %f. \n', avgGrayLevel);
disp(floatAvg)
```

```
The average gray level is 80.272727.
```

Published with MATLAB® R2015b

Question 4

Open an image file

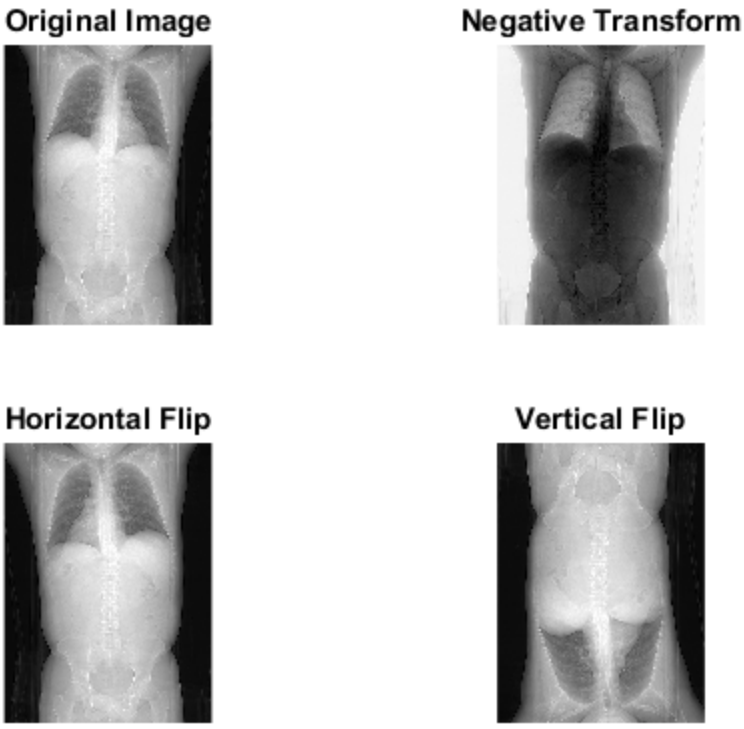
```
close all; clear; clc;
filename = uigetfile({'*.jpg;*.tif;*.png;*.gif;*.pgm','All Image
 Files';...
          '*.*','All Files' },'Pick an Image File');
```

Flip the original image upside-down, left side-right, and compute the negative

```
I = imread(filename);
I2 = imcomplement(I);        % negative transform
I3 = fliplr(I);            % horizontal flip
I4 = flip(I);            % vertical flip
```

Display the four images in a 2x2 arrangement

```
subplot(2,2,1), imshow(I); title('Original Image');
subplot(2,2,2), imshow(I2); title('Negative Transform');
subplot(2,2,3), imshow(I3); title('Horizontal Flip');
subplot(2,2,4), imshow(I4); title('Vertical Flip');
```

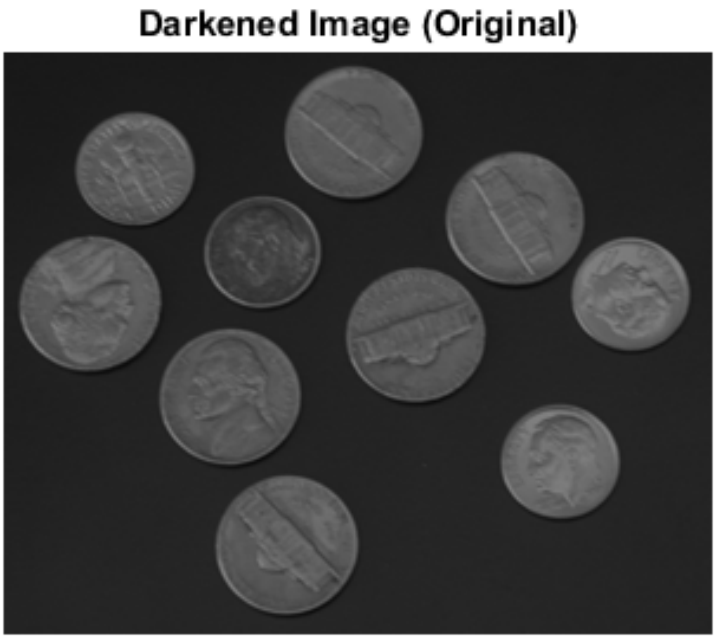


Published with MATLAB® R2015b

Question 5

A beginner MATLAB programmer wrote the code below to darken a grayscale image by reducing the intensity of each pixel by half. Please retype and run their code and answer the following questions:

```
close all; clear; clc;
I = imread('coins.png');
[height, width]= size(I);
J = uint8(zeros(size(I)));
for i =1:height
    for j =1:width
        J(i,j) = 0.5 * I(i,j);
    end
end
imshow(I); title('Original Image');
figure, imshow(J); title('Darkened Image (Original)');
```



a. Does the script work as expected?

Yes, the intesity of each pixel is reduced by half, making the image darker.

b. Will it still work if you remove the uint8 typecasting in line 3? Why (not)?

No, the zeros function creates a matrix of zeros, making the whole image black

c. What is the main problem with this code? Be specific!

The code is inefficient because it attempts to convert each pixels, one at a time, rather than use a function to darken the whole image at once.

d. Rewrite the code to improve its efficiency while keeping it readable.

```
clear; clc;
I = imread('coins.png');
J = imadjust(I,stretchlim(I),[0 .5]);
figure, imshow(J); title('Darkened Image (New)');
```

e. Compare the performance of the original code against the modified version you wrote, using tic and toc.

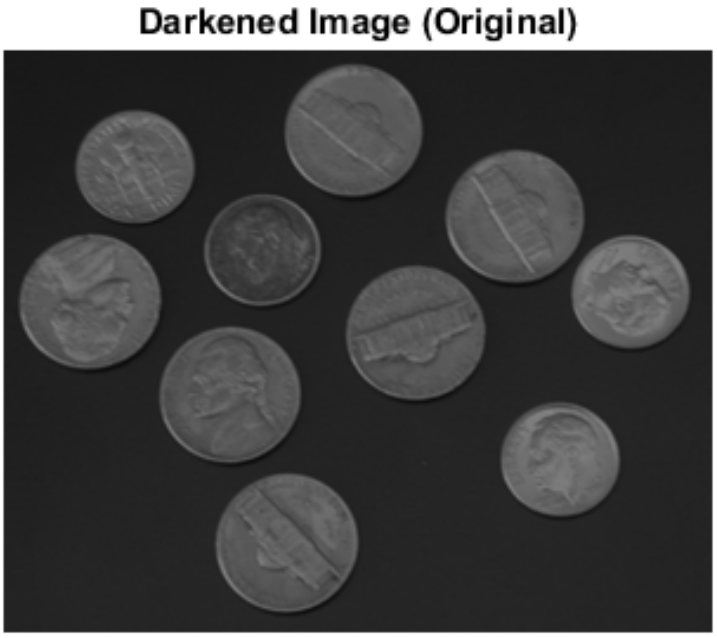
```
clear;
tic
I = imread('coins.png');
[height, width]= size(I);
J = uint8(zeros(size(I)));
for i =1:height
    for j =1:width
        J(i,j) = 0.5 * I(i,j);
    end
end
imshow(I); title('Original Image');
figure, imshow(J); title('Darkened Image (Original)');
t = toc;
elapsed = sprintf('The elapsed time of the original code is');
disp(elapsed)
disp(t)

%
clear;
tic
I = imread('coins.png');
J = imadjust(I,stretchlim(I),[0 .5]);
figure, imshow(J); title('Darkened Image (New)');
t = toc;
elapsed = sprintf('The elapsed time of the new code is');
disp(elapsed)
```

```
disp(t)
```

```
The elapsed time of the original code is
    0.8567

The elapsed time of the new code is
    0.6098
```



Published with MATLAB® R2015b