Image Processing and Computer Vision
UE19CS333
Lab 1

Team Name
019_536_571_589

Team Members

1. Abhishek Aditya BS - PES1UG19CS019
2. T Vijay Prashant - PES1UG19CS536
3. Vishal R - PES1UG19CS571
4. Yashas KS - PES1UG19CS589

Question 1

Read an image (cm=imread('cameraman.tif');) and display the image (figure; imshow(cm1);) and understand coordinate conventions

```
>> % Reading cameraman.tif image and displaying it
>>
>> cm = imread('cameraman.tif');
>> figure(1); imshow(cm); hold on; title("cameraman.tif Image");
```

---

* Equal Contribution - Questions were randomly assigned to each team member by a computer generated program.

1. What is the origin of the image?

   Origin is the Top Left Corner of the image i.e. the coordinates (0,0) or (1,1) in Matlab

```
>> % Origin of the cameraman Image
>>
>> origin = cm(1,1);
>> origin

origin =

  uint8

   156
```

2. What is the intensity value at the centre of the image ?

```
>> % Intensity Value at the center of the Image
>>
>> [x,y] = size(cm);
>> center = cm(x/2, y/2);
>> center

center =

  uint8

   160
```

3. Accessing one row (eg. Row 150)

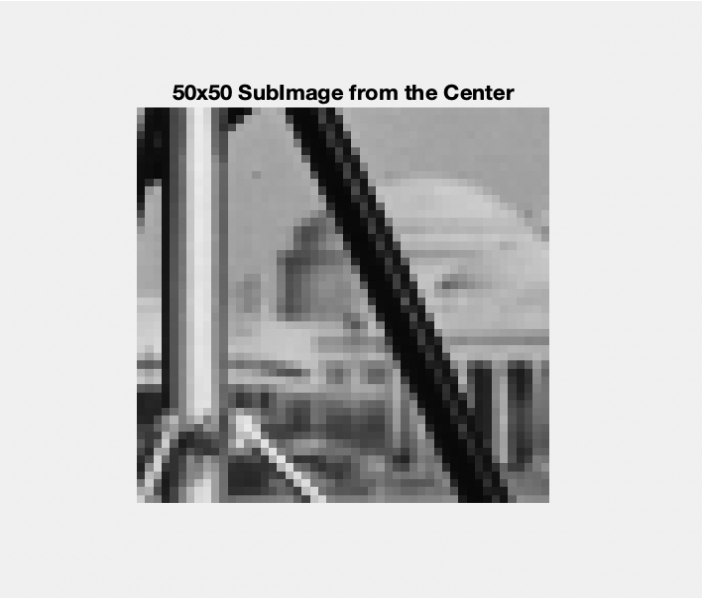
```
>> % Accessing Row 150
>> Row150 = cm(150,:);
>> Row150

Row150 =

  1×256 uint8 row vector
```

4. Accessing one column (eg. Column 170)

```
>> % Accessing Column 170
>> Col170 = cm(:,170);
>> Col170

Col170 =

  256×1 uint8 column vector
```

5. Accessing a subset of rows and columns (50x50 subimage from the centre)

```
>> % Accessing a subset of rows and columns (50x50 subimage from the center)
>> sub50x50 = cm(((end/2)):((end/2)+49), ((end/2)):((end/2)+49));
>> sub50x50

sub50x50 =

  50×50 uint8 matrix
```



Question 2

What happens (a) when we multiply cameraman by 2 and (b) divide cameraman by 2?

```
>> figure(1);imshow(i1);title('Original');
>> figure(2);imshow(i1 .* 2);title('Multiplied by 2');
>> figure(3);imshow(i1 ./ 2);title('Divided by 2');
```

(a) When cameraman is multiplied by 2 every pixel in the image is multiplied by 2 resulting in an overall lighter shade if the resultant pixel exceeds 255 it is clipped to 255. It can be seen that many of the buildings which were on the lighter scale disappear when multiplied by 2.

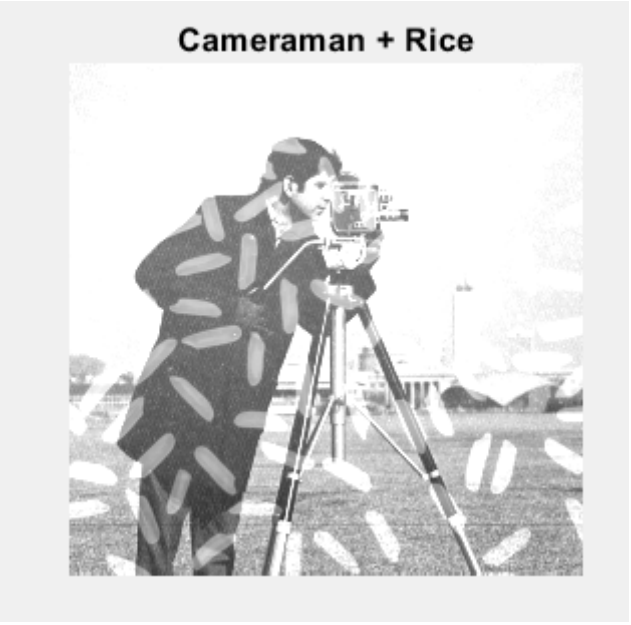
(b) When the cameraman is divided by 2 every pixel in the image is divided by 2 resulting in an overall darker shade all the fractional values are rounded up.

It can also be noted that scaling affects pixels having higher value more than pictures having low value.
Multiplication by a factor is one of the simple methods to scale an image. Eg: If images are taken in low lighting, scaling by factor > 1 results in the features of the image being brought out.

Question 3

To the original cameraman image, what happens if we add rice.tif?

```
>> figure(1);imshow(i1 + imread('rice.png'));title('Cameraman + Rice');
```



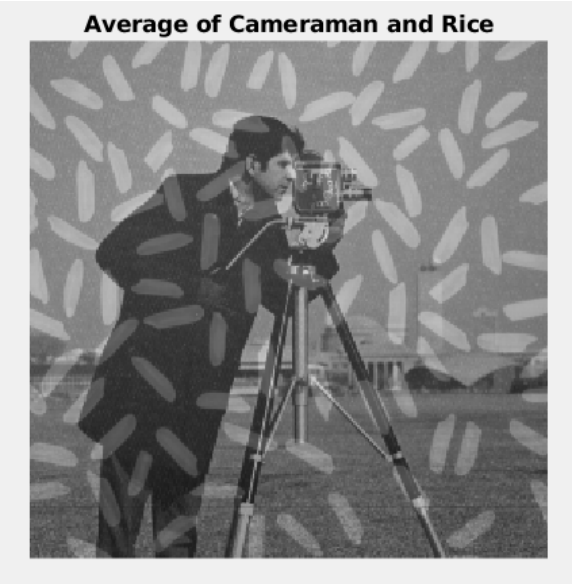
Every pixel of both the images are added and the resultant image is shown.
Resultant pixel values greater than 255 are clipped to 255.
Rice patterns can be seen in the picture wherever the cameraman has a darker background.

Question 4

What is the result of averaging cameraman and rice?

```
>> cameraman = imread('cameraman.tif');
>> rice = imread('rice.png');
>> avg = cameraman .* 0.5 + rice .* 0.5;
>>
```

Pixels values in both the images are reduced by half and then added resulting in an image that contains both cameraman and rice grains but with reduced pixel intensities.

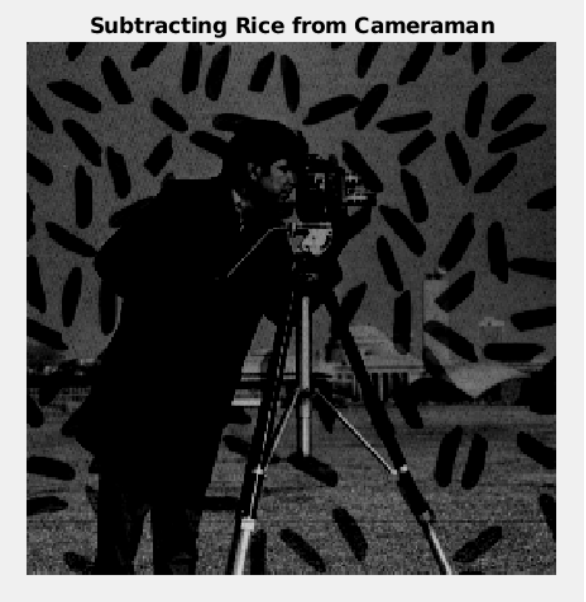


Question 5

What is the result of subtracting rice from the cameraman?

```
>> cm_sub_rice = cameraman - rice;
```

Every pixel value in the rice image is subtracted from the cameraman image. This results in an image where, wherever there was a rice grain, those pixels get subtracted from the cameraman leading to dark spots in places where there exists a rice grain.



Question 6

What are the minimum and maximum intensity values in the resulting image above?

```
>> min(cm_sub_rice(:))

ans =

  uint8

   0
```

```
>> max(cm_sub_rice(:))

ans =

  uint8

   209
```

Minimum intensity value for the above resulting image is 0 whereas the maximum intensity value for the image is 209.

a. Where are they found?

```
>> [minval,minpos] = min(cm_sub_rice(:))

minval =

  uint8

   0

minpos =

    23

>> [row,col] = ind2sub(256,minpos)

row =

    23

col =

     1
```

```
>> [maxval,maxpos] = max(cm_sub_rice(:))

maxval =

  uint8

   209

maxpos =

       26367

>> [row,col] = ind2sub(256,maxpos)

row =

   255

col =

   103
```

It can be seen that the Minimum intensity value can be found at the location (23,1) and the Maximum intensity value can be found at the location (255,103).

b. Is the min(min(cameraman)) the same as the minimum value of intensity found in cameraman?

Yes, the two return the minimum intensity value in the cameraman image as min(min(cameraman)) takes the min of the minimum values of each row which eventually gives us the overall minimum intensity value of the cameraman image.

c. How do you find the position of the minimum value in 2D array terms (row, column)?

There are three methods to find the minimum value in a 2d array:

1. Use the min function to find the position of the minimum value and since the position returned is in the form of a 1D array we need to convert it into a 2D array indexing using the ind2sub function by specifying the size of the image which gives the row and column value.
2. Use the min function and to find the position given by the min function we divide the 1D min position by arraysize to give column number and take modulo to give the row number.

```
>> row = maxpos - 256.*floor(maxpos./256)

row =

   255

>> col = maxpos/256

col =

  102.9961
```

d. How do you find every instance of the minimum intensity value in the image?

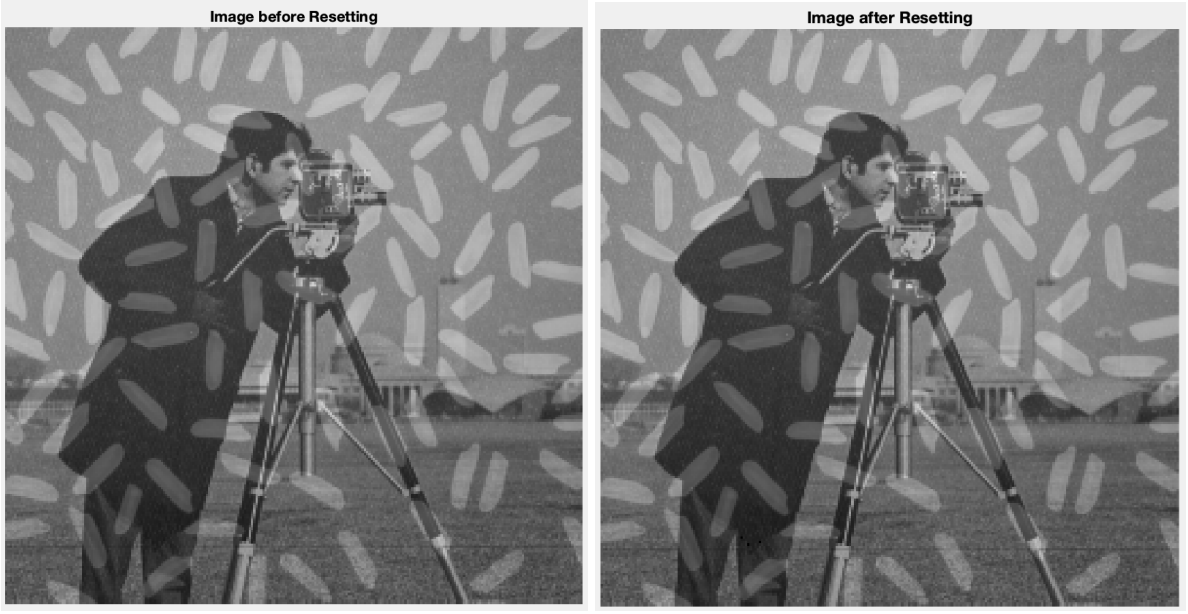
We can find every instance of the minimum intensity value in an image using the find function.

```
>> [row,col] = find(cm == min(cm(:)))

row =

   135
   134
   134
   135


col =

    61
    62
    63
    63
```

Question 7

Reset the minimum to 0 and maximum to 255 in the resulting image in Question 4. What do you see now?

```
>> reset_avg = avg;
>> reset_avg(reset_avg==min(reset_avg(:))) = 0;
>> reset_avg(reset_avg==max(reset_avg(:))) = 255;
```



There is a very minute change in the brightness level of the images before and after resetting the min and max values.The image after resetting the boundary values shows a higher contrast and brightness compared to the image before resetting.

Question 8

Downsample cameraman by 2 – at what stage does it start to appear pixelated?

The term pixelated is used to describe an image in which the individual pixels are visible to the naked eye. A pixelated image will appear blurry and blocky.

```
>> cm256x256 = imread('cameraman.tif');
>> figure(1); imshow(cm256x256); hold on; title("Original 256x256 Image");
>> cm128x128 = cm256x256(1:2:end, 1:2:end);
>> figure(2); imshow(cm128x128); hold on; title("DownSampled 128x128 Image");
>> cm64x64 = cm128x128(1:2:end, 1:2:end);
>> figure(3); imshow(cm64x64); hold on; title("DownSampled 64x64 Image");
>> cm32x32 = cm64x64(1:2:end, 1:2:end);
>> figure(4); imshow(cm32x32); hold on; title("DownSampled 32x32 Image");
```

As we can see from the downsampled images, at 128x128 we can see jagged edges, which is more prominent in downsampled 64x64, and in downsampled 32x32 the image appears pixelated with individual pixels visible to the naked eye and image appearing blurry and blocky.

Question 9

Is there a way to get rid of the 'jagged edges' in Cameraman?

We can get rid of the jagged edges using two methods:

1. One way is to blur the image with imfilter() or conv2().
2. The other way is to smooth the outline coordinates with a Savitzky-Golay filter, sgolayfilt() in the Signal Processing Toolbox.

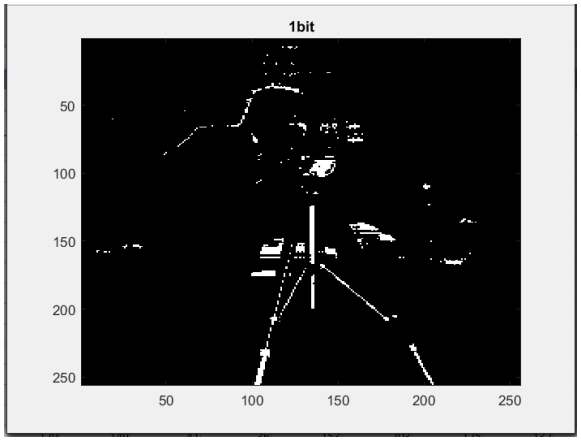
Question 10

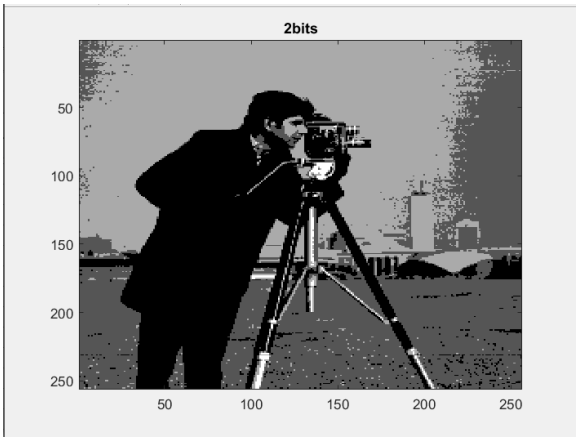
To the original cameraman image, apply uniform quantization (set every two gray levels to one gray level) all the way from 8 bits to 1 bit.

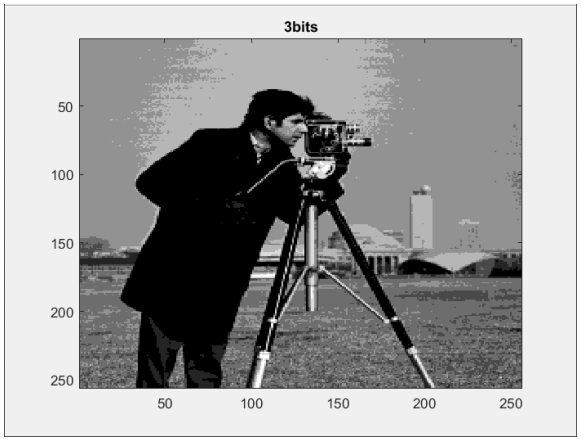
```
>> figure();imagesc(i1);colormap(gray);title('8bits');
figure();imagesc((i1 - 1)./2);colormap(gray);title('7bits');
figure();imagesc((i1 - 3)./4);colormap(gray);title('6bits');
figure();imagesc((i1 - 7)./8);colormap(gray);title('5bits');
figure();imagesc((i1 - 15)./16);colormap(gray);title('4bits');
figure();imagesc((i1 - 31)./32);colormap(gray);title('3bits');
figure();imagesc((i1 - 63)./64);colormap(gray);title('2bits');
figure();imagesc((i1 - 127)./128);colormap(gray);title('1bit');
```

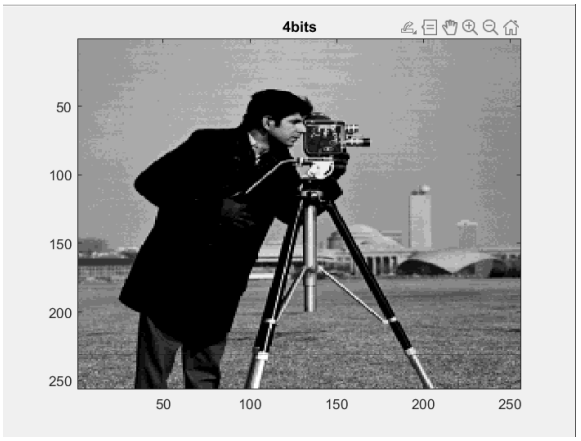
Note : Matlab rounds up the decimals. Subtracting (2 ^ (#bits) - 1) before dividing will result in correct mapping.

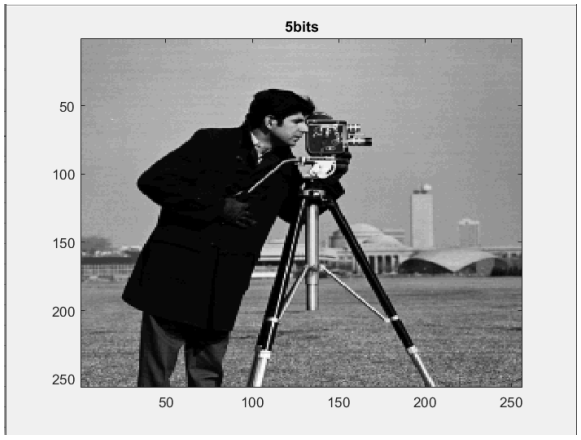
It can be seen that contouring starts at 4 bits

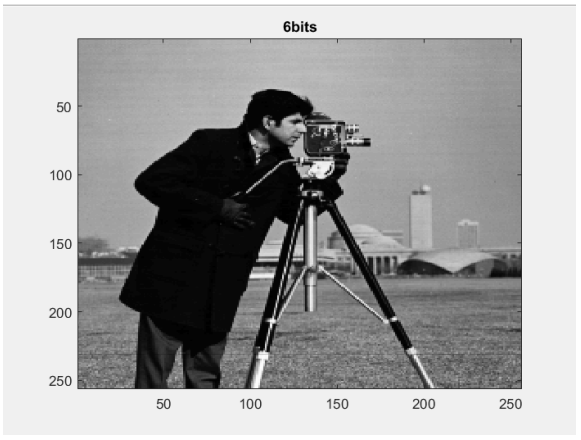

1bit
50
100
150
200
250
50
100
150
200
250

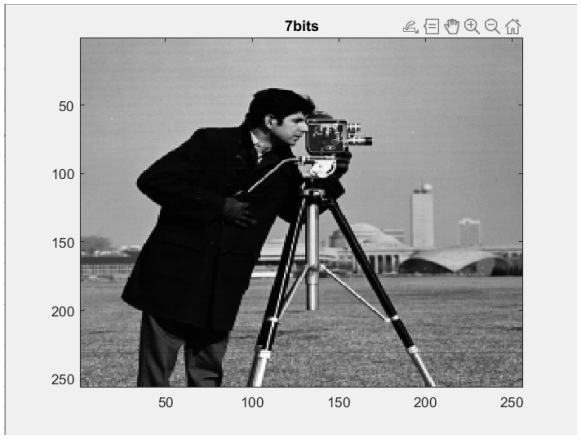

2bits
50
100
150
200
250
50
100
150
200
250

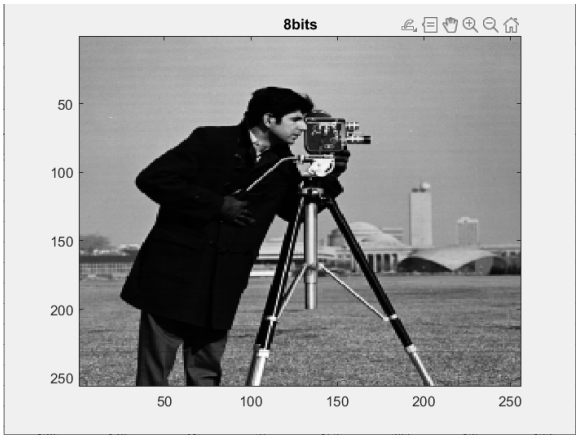

3bits
50
100
150
200
250
50
100
150
200
250


4bits
50
100
150
200
250
50
100
150
200
250


5bits
50
100
150
200
250
50
100
150
200
250


6bits
50
100
150
200
250
50
100
150
200
250


7bits
50
100
150
200
250
50
100
150
200
250


8bits
50
100
150
200
250
50
100
150
200
250

Question 11

What is the average intensity of the cameraman image? Apply a threshold to compute a binary image. Compare this image with the 1 bit representation in 10. (That is, find the 'difference image' and compute the sum of the difference.)

Average Intensity of Cameraman Image

```
>> avg_intensity = mean(cameraman(:))

avg_intensity =

   118.7245
```

Computing the binary image by applying a threshold of 119

```
>> threshold_image = imread('cameraman.tif');
>> threshold = 119;
>> filter = threshold_image >= threshold;
>> figure(1); imshow(filter); hold on; title('Binarized Image of Cameraman (threshold = 119)');
```

Computing the difference image and summing the difference image

```
>> one_bit_image = (cameraman - 127)./128;
>> difference_image = threshold_image - one_bit_image;
>> sum(difference_image(:))

ans =

     7779480
```

Question 12

Use the binary version of the cameraman to select intensity values from the original image. What is the average of the gray level intensity values in this masked image?

```
>> cm = imread('cameraman.tif');
>> avg_intensity = mean(cm(:));
>> avg_intensity

avg_intensity =

  118.7245

>> avg_intensity = round(avg_intensity);
>> cmBinary = (cm > avg_intensity);
>> cmMasked = cmBinary .* double(cm);
>> figure(10); imshow(uint8(cmMasked)); title("Original Image Masked with Binary Image");
>> mean(cmMasked(cmMasked~=0))

ans =

  160.0131

>> % The average intensity of the Masked Image is 160
```