IIT Vadodara
Autumn 2018-19
TE3 Computer Vision
Lab-2: Preliminary operations on digital images

Shivvrat Arya (201551059)

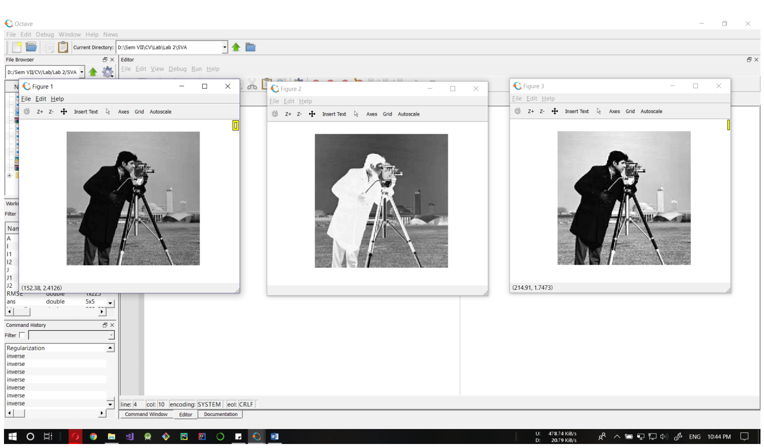
August 28, 2018

Problem 1. Compare imadjust() and imcomplement() for obtaining negative of an image.

Solution In matlab, imadjust() is used to adjust the intensity values or the colormap of the image, while incompliment() is used to complement the image as a whole. So it just inverts the image.

CODE

```
pkg load image;
I = imread ("cam.jpg");
figure(1)
imshow(I)
J = imcomplement(I);
figure(2)
imshow(J)
J2 = imadjust(I);
figure(3)
imshow(J2)
```

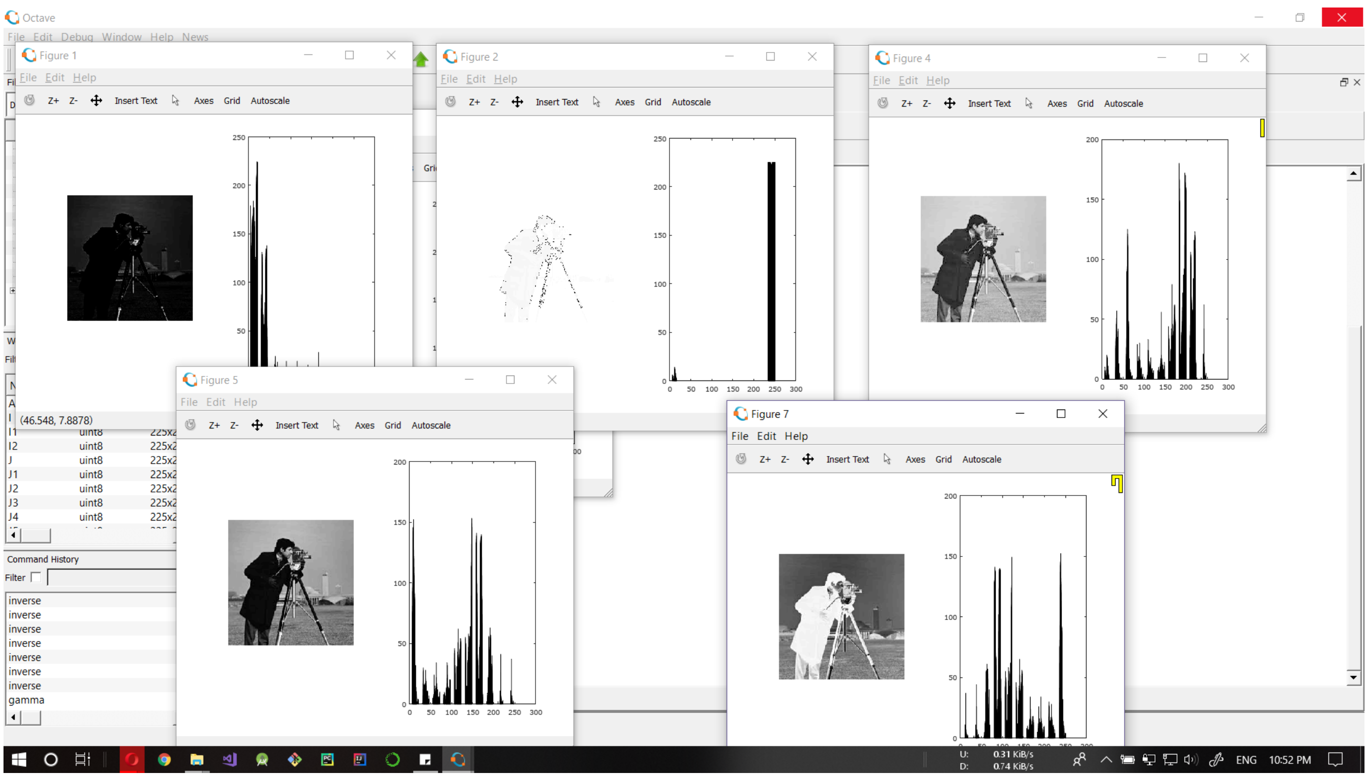


In this question we tried to compare two different functions of octave, imadjust and incompliment. They are used to change the properties of the image by changing the values on the pixles.

Problem 2. Observe the effects of changing Gamma on an image. Show their effects on band of interest using histogram of the input image and corresponding processed image.

Solution
```
pkg load image;
I = imread("cam.jpg");
imshow(I)
J1= imadjust(I,[],[],0.005);
figure(1);
subplot(1,2,1);
imshow(J1)
subplot(1,2,2);
hist(J1)
J2= imadjust(I,[],[],0.05);
figure(1);
subplot(1,2,1);
imshow(J2)
subplot(1,2,2);
hist(J2)
J3= imadjust(I,[],[],0.01);
figure(2);
subplot(1,2,1);
imshow(J3)
subplot(1,2,2);
hist(J3)
J4= imadjust(I,[],[],0.1);
figure(3);
subplot(1,2,1);
imshow(J4)
subplot(1,2,2);
hist(J4)
J5= imadjust(I,[],[],0.5);
figure(4);
subplot(1,2,1);
imshow(J5)
subplot(1,2,2);
hist(J5)
J6= imadjust(I,[],[],1.0);
figure(5);
subplot(1,2,1);
imshow(J6)
subplot(1,2,2);
```

```
hist(J6)
J7= imadjust(I,[],[],5.0);
figure(7);
subplot(1,2,1);
imshow(J)
subplot(1,2,2);
hist(J)
J= imadjust(I,[],[],10.0);
figure(1);
subplot(1,2,1);
imshow(J7)
subplot(1,2,2);
hist(J7)
```



In this problem we tried to see the effect of chaning gamma on the image. We used the imadjust function to change the gamma and also plotted the histogram for every image to see the characteristics of the image when we changed the gamma.

Problem 3. Demonstrate following linear filtering operations using imfilter(): (i) low-pass filter, (ii) high-pass filter, and (iii) high-boost filter.

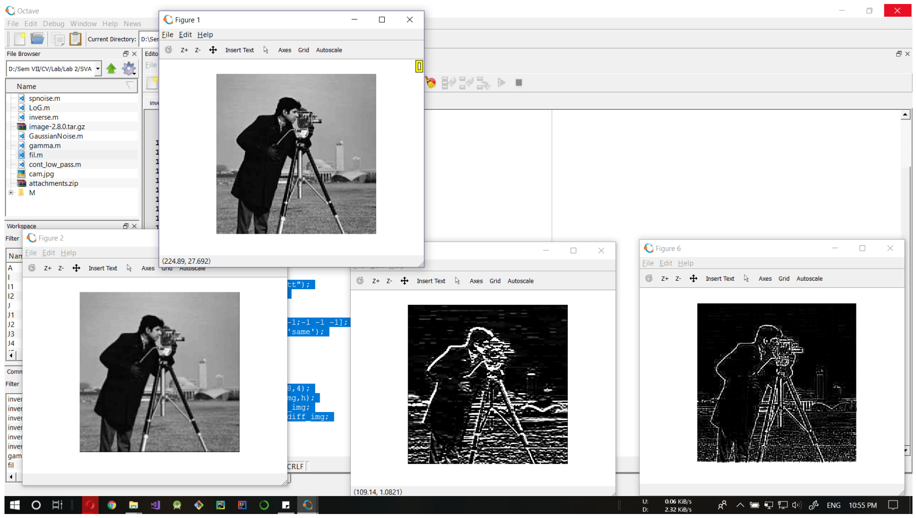
Solution
```
pkg load image;
I = imread("cam.jpg");
figure(1);
```

```
imshow(I)
% Low pass filter
filter = fspecial("average", 2);
J = imfilter(I, filter);
figure(2);
imshow(J)
filter = fspecial("average", 8);
J = imfilter(I, filter);
figure(3);
imshow(J)
filter=(1/9)*ones(3,3);
J = imfilter(I, filter);
figure(3);
imshow(J)


% High pass filter
filter = fspecial("kirsch");
J = imfilter(I,filter);
figure(5);
imshow(J)
filter = fspecial("prewitt");
J = imfilter(I, filter);
figure(6);
imshow(J)
filter = [-1 -1 -1;-1 8 -1;-1 -1 -1];
J = imfilter(I, filter, 'same');
figure(6);
imshow(J)


% High Boost Filter
h = fspecial('gaussian',8,4);
blurred_img = imfilter(img,h);
diff_img = img - blurred_img;
highboost_img = img + 5*diff_img;
```

In this problem we tried to see how different types of filters work. We implemented low pass filter by average filter, high pass filter by kirsch and prewitt funcions in fspecial and high boost filter by the gaussian filter.

Problem 4. Consider an image. Apply Laplacian of Gaussian (LoG) filtering opera- tion to detect the edges of the image. Experiment with different sizes of

masks and windows. Compare their outputs along with corresponding histograms. Use fspecial() and imfilter() commands.



Solution
```
pkg load image;
I = imread("cam.jpg");
figure(1);
imshow(I)
filter = fspecial('log', [3 3] , 0.2 )
J = imfilter(I, filter);
figure(2);
subplot(1,3,1);
imshow(J)
subplot(1,3,2);
hist(J)
subplot(1,3,3);
hist(I);
filter = fspecial('log', [3 3] , 0.4 )
J = imfilter(I, filter);
figure(3);
subplot(1,3,1);
imshow(J)
subplot(1,3,2);
hist(J)
subplot(1,3,3);
hist(I);
filter = fspecial('log', [3 3] , 0.6 )
J = imfilter(I, filter);
figure(4);
subplot(1,3,1);
imshow(J)
subplot(1,3,2);
hist(J)
subplot(1,3,3);
```

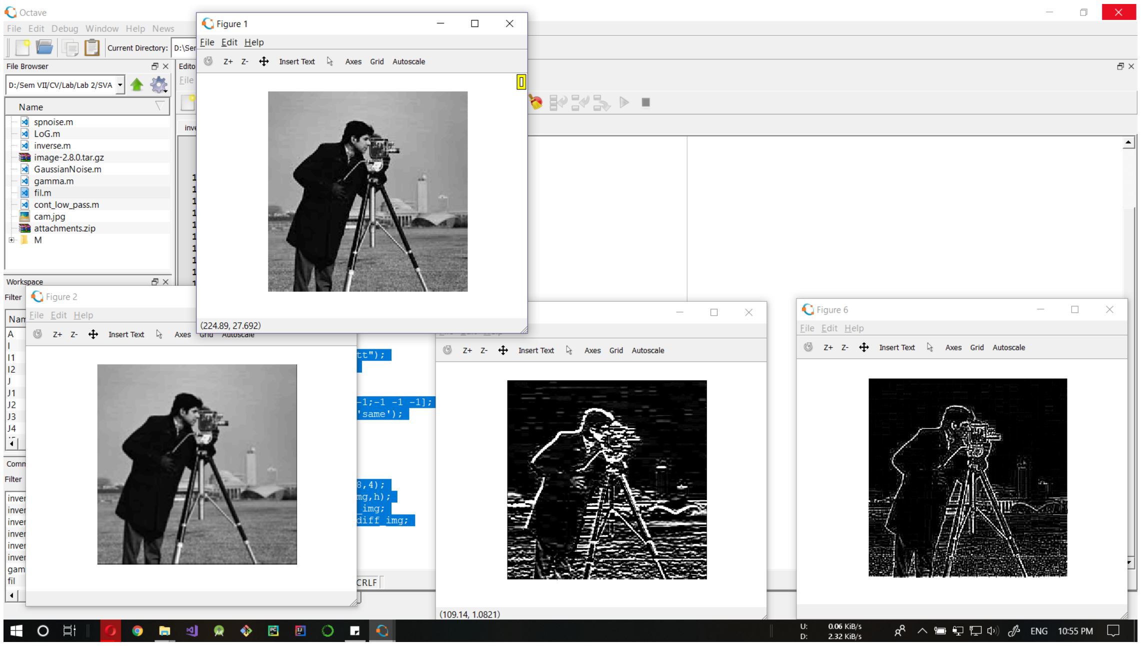
```
hist(I);
filter = fspecial('log', [5 5] , 0.3 )
J = imfilter(I, filter);
figure(5);
subplot(1,3,1);
imshow(J)
subplot(1,3,2);
hist(J)
subplot(1,3,3);
hist(I);
filter = fspecial('log', [5 5] , 0.5 )
J = imfilter(I, filter);
figure(5);
subplot(1,3,1);
imshow(J)
subplot(1,3,2);
hist(J)
subplot(1,3,3);
hist(I);
filter = fspecial('log', [5 5] , 0.7 )
J = imfilter(I, filter);
figure(5);
subplot(1,3,1);
imshow(J)
subplot(1,3,2);
hist(J)
subplot(1,3,3);
hist(I);
```

Here we applied the LoG function on the image by using the 'log' as a input to the function.

Problem 5. Compare and contrast the functioning of LoG and the high-pass filtering operations for detecting edges from an image.

Solution aplacian filters are derivative filters used to find areas of rapid change (edges) in images. Since derivative filters are very sensitive to noise, it is common to smooth the image (e.g., using a Gaussian filter) before applying the Laplacian. This two-step process is call the Laplacian of Gaussian (LoG) operation. There are different ways to find an approximate discrete convolution kernal that approximates the effect of the Laplacian. While in high pass filter is the one, that passes signals with a frequency higher than a certain cutoff frequency and attenuates signals with frequencies lower than the cutoff frequency. The amount of attenuation for each frequency depends on the filter design. A high-pass filter is usually modeled as a linear time-invariant system.

Question 6. Consider an image. Add different amounts of salt pepper noise in it. Now try



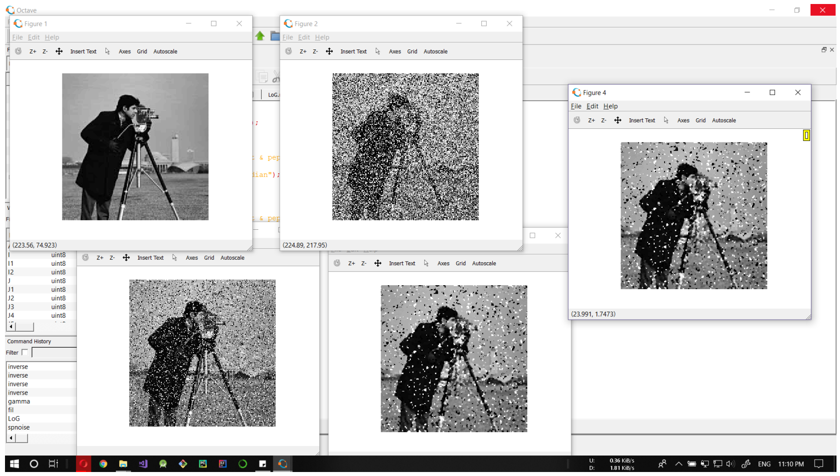
removing the noise using appropriate filtering in the spatial domain of the image. Display all the outputs.

Solution pkg **load image**;

```
figure(1);
I = imread ("cam.jpg");
imshow(I)
figure(2)
density = 0.5;
I1 = imnoise (I, "salt & pepper", density);
imshow(I1)
J1 = imsmooth(I1, "Median");
figure(5)
imshow(J1)
figure(3)
density = 0.2;
I2 = imnoise (I, "salt & pepper", density);
imshow(I2)
figure(4)
J2 = imsmooth(I1, "Median");
imshow(J2)
```

In this solution we added salt and pepper noise to the image tried to see how it affects the image characteristics

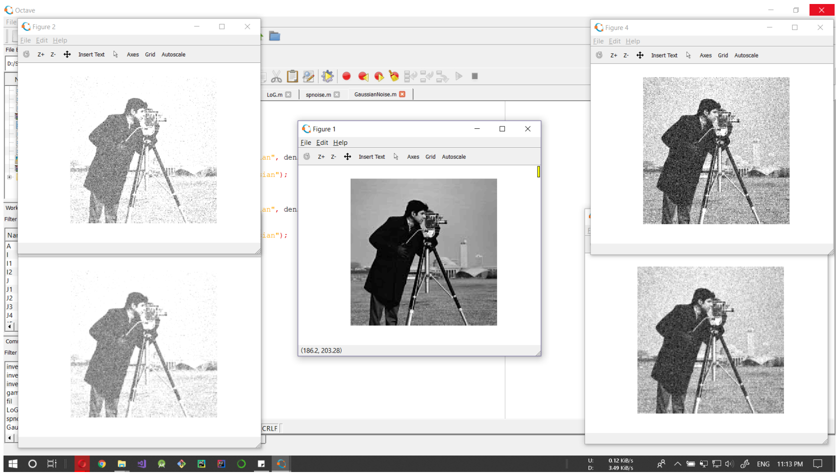
Question 7. Repeat the Q.6 for Gaussian noise.



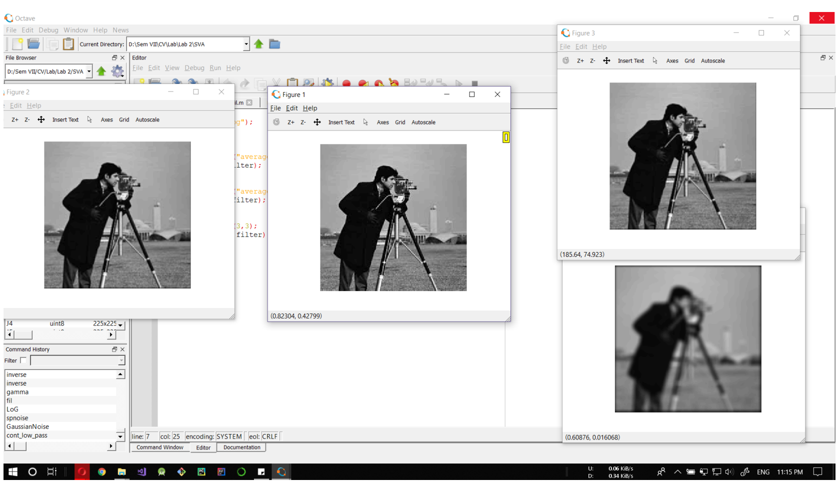
Solution
```
pkg load image;
figure(1);
I = imread ("cam.jpg");
imshow(I)
figure(2)
density = 0.5;
I1 = imnoise (I, "gaussian", density);
imshow(I1)
J1 = imsmooth(I1, "Gaussian");
figure(3)
imshow(J1)
density = 0.2;
I2 = imnoise (I, "gaussian", density);
figure(4)
imshow(I2)
J2 = imsmooth(I2, "Gaussian");
figure(5)
imshow(J2)
```

Question 8. Demonstrate the effects of repeatedly applying low-pass filter operation on an image.

Solution
```
pkg load image;
I = imread("cam.jpg");
figure(1);
imshow(I)
% Low pass filter
filter = fspecial("average", 2);
J = imfilter(I, filter);
figure(2);
imshow(J)
```



```
filter = fspecial("average", 8);
J1 = imfilter(J, filter);
figure(3);
imshow(J)
filter=(1/9)*ones(3,3);
J2 = imfilter(J1, filter);
figure(4);
imshow(J2)
```



Here we replied the low pass filter multiple times on the image and tried to find how it affects the image.