```
A=imread('cameraman.tif'); % Read in image
subplot(1,2,1), imshow(A); % Display image
func = @(x) max(x(:)); % set filter to apply
B = nlfilter(A,[3 3],func); % apply over 3 x 3 neighbourhood
subplot(1,2,2), imshow(B); % Display result image B
```

****

```
A=imread('peppers.png'); % Read in image
subplot(1,2,1), imshow(A); % Display image
k = fspecial('motion', 50, 54); % create a 5x5 convolution kernel
B = imfilter(A, k, 'symmetric'); % apply using symmetric mirroring at edges
subplot(1,2,2), imshow(B); % Display result image B
```

****

```
I=imread('eight.tif'); % Read in image
subplot(1,3,1), imshow(I); % Display image
Isp = imnoise(I,'salt & pepper',0.03); % add 3% (0.03) salt and pepper noise
subplot(1,3,2), imshow(Isp); % Display result image Isp
Ig = imnoise(I,'gaussian',0.02); % add Gaussian noise (with 0.02 variance)
subplot(1,3,3), imshow(Ig); % Display result image Ig
```

//////////////////////////////////

```
k = ones(3,3) / 9;    % define mean filter
```

```
I_m = imfilter(I,k); % apply to original image
Isp_m = imfilter(Isp,k); % apply to salt and pepper image
Ig_m = imfilter(Ig,k); % apply tp gaussian image
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image
```

//

```
I_m = medfilt2(I,[3 3]); % apply to original image
Isp_m = medfilt2(Isp,[3 3]); % apply to salt and pepper image
Ig_m =medfilt2(Ig,[3 3]); % apply tp gaussian image
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image
```

//////////////////////////////////////

```
I_m = ordfilt2(I,25,ones(5,5)); % apply to original image
Isp_m = ordfilt2(Isp,25,ones(5,5)); % apply to salt and pepper image
```

```
Ig_m =ordfilt2(Ig,25,ones(5,5)); % apply tp gaussian image
subplot(1,3,1), imshow(I_m); % Display result image
subplot(1,3,2), imshow(Isp_m); % Display result image
subplot(1,3,3), imshow(Ig_m); % Display result image
```

//

```
k = fspecial('gaussian', [5 5], 2); % define Gaussian filter
I_g = imfilter(I,k); % apply to original image
Isp_g = imfilter(Isp,k); % apply to salt and pepper image
Ig_g = imfilter(Ig,k); % apply tp gaussian image
subplot(1,3,1), imshow(I_g); % Display result image
subplot(1,3,2), imshow(Isp_g); % Display result image
subplot(1,3,3), imshow(Ig_g); % Display result image
```

//////////////////////////

```
I=imread('circuit.tif'); % Read in image
IEr = edge(I,'roberts'); % roberts edges
IEp = edge(I,'prewitt'); % prewitt edges
IEs = edge(I,'sobel');   % sobel edges

subplot(2,2,1), imshow(I); % Display image
subplot(2,2,2), imshow(IEr); % Display image
subplot(2,2,3), imshow(IEp); % Display image
subplot(2,2,4), imshow(IEs); % Display image
```

//

```
I=rgb2gray(imread('peppers.png')); % Read in
image (in greyscale)
k = fspecial('laplacian'); % create laplacian
filter
IEl = imfilter(double(I),k,'symmetric'); %
laplacian edges

subplot(1,2,1), imagesc(I); % Display image
subplot(1,2,2), imagesc(IEl); % Display image
colormap('gray');
```

///

```
I=rgb2gray(imread('peppers.png')); % Read in
image (in greyscale)
k = fspecial('log'); % create laplacian
filter
IEl = imfilter(double(I),k,'symmetric'); %
laplacian edges

subplot(1,2,1), imagesc(I); % Display image
subplot(1,2,2), imagesc(IEl); % Display image
colormap('gray');
```

////////////////////////////////////

```
I=rgb2gray(imread('peppers.png')); % Read in
image (in greyscale)
k = fspecial('log', [10 10], 3.0); % create
laplacian filter
IEzc = edge(I, 'zerocross', [], k); % zero
crossing edges (auto thresholded)
```

```
subplot(1,2,1), imshow(I); % Display image
subplot(1,2,2), imshow(IEzc); % Display image
```

//////////////////////////////////////

```
A=imread('cameraman.tif'); %Read in image
h=fspecial('laplacian'); %Generate 3x3
Laplacian filter
B=imfilter(A,h); %Filter image with Laplacian
kernel
C=imsubtract(A,B); %Subtract Laplacian from
original.
subplot(1,3,1), imshow(A);
subplot(1,3,2), imagesc(B); axis image; axis
off %Display original, Laplacian and
subplot(1,3,3), imshow(C); %Enhanced image
```

////////////////////////////////

```
A=double(imread('cameraman.tif')); %Read in
image
h=fspecial('laplacian'); %Generate 3x3
Laplacian filter
B=imfilter(double(A),h); %Filter image with
Laplacian kernel
C=imsubtract(A,B); %Subtract Laplacian from
original.
subplot(1,3,1), imagesc(A); colormap('gray');
%Display original, Laplacian and
subplot(1,3,2), imagesc(B);
colormap('gray'); %Enhanced image
axis image; axis off
subplot(1,3,3), imagesc(C);
```

//

```
A=imread('cameraman.tif'); %Read in image
```

```
h=fspecial('log', [10 10], 3.0); %Generate
3x3 LoG filter
B=imfilter(A,h); %Filter image with Laplacian
kernel
C=imsubtract(A,B); %Subtract Laplacian from
original.
subplot(1,3,1), imshow(A);
subplot(1,3,2), imagesc(B); axis image; axis
off %Display original, Laplacian and
subplot(1,3,3), imshow(C); %Enhanced image
////////////////////////////////////////

Iorig=imread('cameraman.tif');  %Read in
image
g=fspecial('gaussian',[5 5],1.5); %Generate
gaussian kernel
subplot(2,3,1), imshow(Iorig);  %Display
original image
Is=imfilter(Iorig,g);       %Create smoothed
image by filtering
Ie=(Iorig - Is);         %Get difference
image
subplot(2,3,2), imshow(Ie); %Display unsharp
difference

Iout=Iorig+(0.3).*Ie; %Add k * difference
image to original
subplot(2,3,3),  imshow(Iout);
Iout=Iorig+(0.5).*Ie; %Add k * difference
image to original
subplot(2,3,4),  imshow(Iout);
Iout=Iorig+(0.7).*Ie; %Add k * difference
image to original
subplot(2,3,5),  imshow(Iout);
```

```
Iout=Iorig+(2.0).*Ie; %Add k * difference
image to original
subplot(2,3,6),  imshow(Iout);
```