Image Convolution Kernels

Step-by-Step Solution for Exercise Proper

1. Encode the following convolution matrices in MATLAB

$H_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$H_2 = \left(\frac{1}{16}\right) * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$H_3 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
$H_4 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$H_5 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$H_6 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
$H_7 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	$H_8 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$	

```matlab
H1 = [0 0 0;...
        0 1 0;...
        0 0 0];
H2 = (1/16)*...
        [1 2 1;...
        2 4 2;...
        1 2 1];
H3 = [0 -1 0;...
        -1 5 -1;...
        0 -1 0];
H4 = [1 0 -1;...
           2 0 -2;...
           1 0 -1];
H5 = [0 1 0;...
        1 -4 1;...
        0 1 0];
H6 = [1 2 1;...
           0 0 0;...
           -1 -2 -1];
H7 = [-1 -1 -1;...
        -1 8 -1;...
        -1 -1 -1];
H8 = [-2 -1 0;...
        -1 1 1;...
        0 1 2];
```

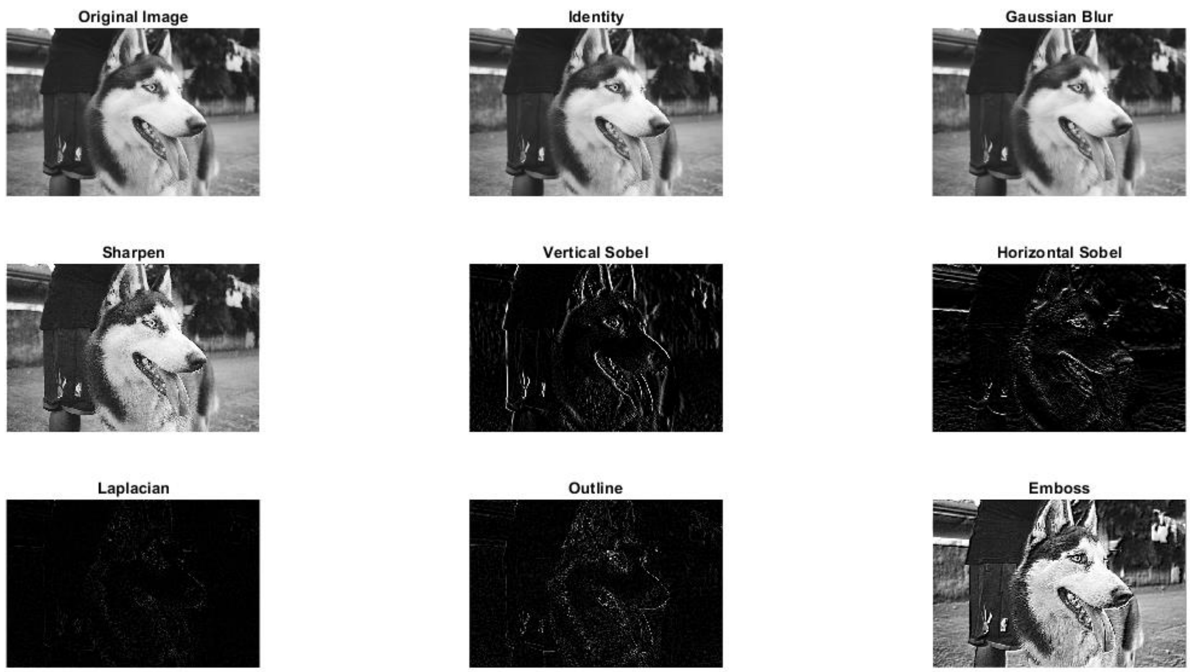
2. Using the grayscale version of the image dog.png, apply all the convolution kernels in #1. Show the original image and all the filtered images in one figure (use subplot). Ensure that the resolution of the image is 512x512. From the given descriptions in the theoretical background, determine the name of each convolution kernel. Put this as the title of the subplot.

```
%Acquire image and place on your current directory
%read the image using imread() and store it into a variable
I = imread('dog', 'png');
%Convert image to greyscale
Igs = im2gray(I);


%Filter image with all convolution kernels.
%Use function imfilter()
%Assure 512x512 image using uint8()
Idenfil = uint8(imfilter(Igs, H1));
Igbfil = uint8(imfilter(Igs, H2));
Ishfil = uint8(imfilter(Igs, H3));
Ivsfil = uint8(imfilter(Igs, H4));
Ihsfil = uint8(imfilter(Igs, H6));
Ilapfil = uint8(imfilter(Igs, H5));
Ioutlfil = uint8(imfilter(Igs, H7));
Iembofil = uint8(imfilter(Igs, H8));


%Show all the filtered images in one figure
figure(1);
subplot(331); imshow(Igs); title('Original Image')
subplot(332); imshow(Idenfil); title('Identity')
subplot(333); imshow(Igbfil); title('Gaussian Blur')
subplot(334); imshow(Ishfil); title('Sharpen')
subplot(335); imshow(Ivsfil); title('Vertical Sobel')
subplot(336); imshow(Ihsfil); title('Horizontal Sobel')
subplot(337); imshow(Ilapfil); title('Laplacian')
subplot(338); imshow(Ioutlfil); title('Outline')
subplot(339); imshow(Iembofil); title('Emboss')
```

- H1 = Identity
- H2 = Gaussian Blur
- H3 = Sharpen
- H4 = Sobel Operator for Vertical Edge Detection
- H5 = Laplacian Operator
- H6 = Sobel Operator for Horizontal Edge Detection
- H7 = Outline Kernel

- H8 = Emboss Kernel

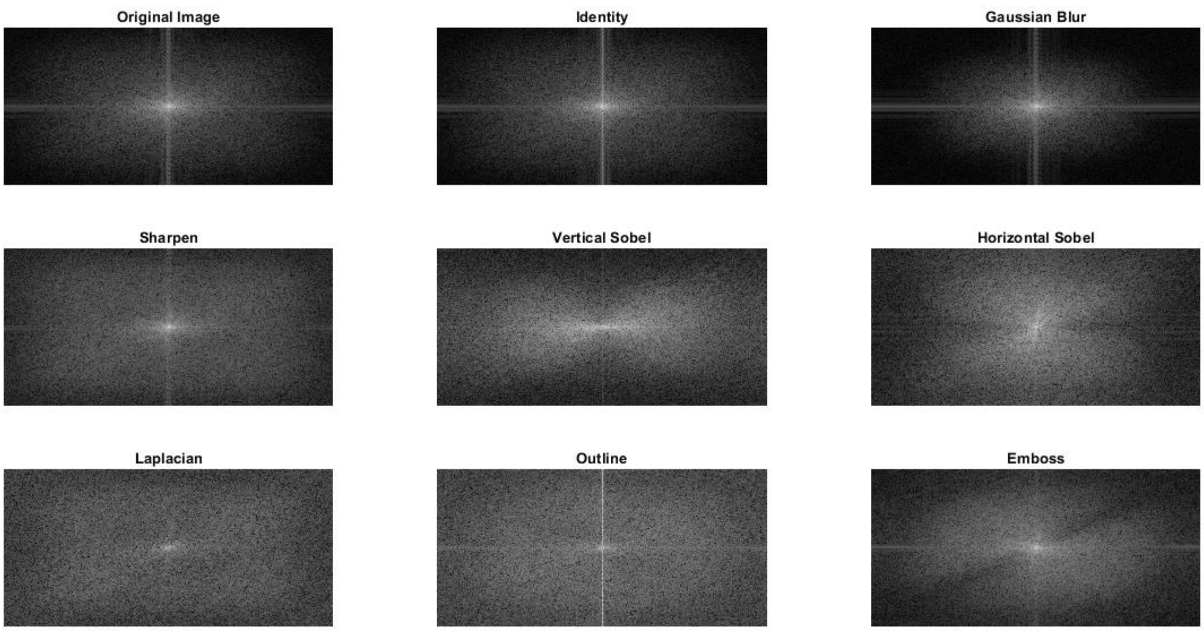
Expected Output:

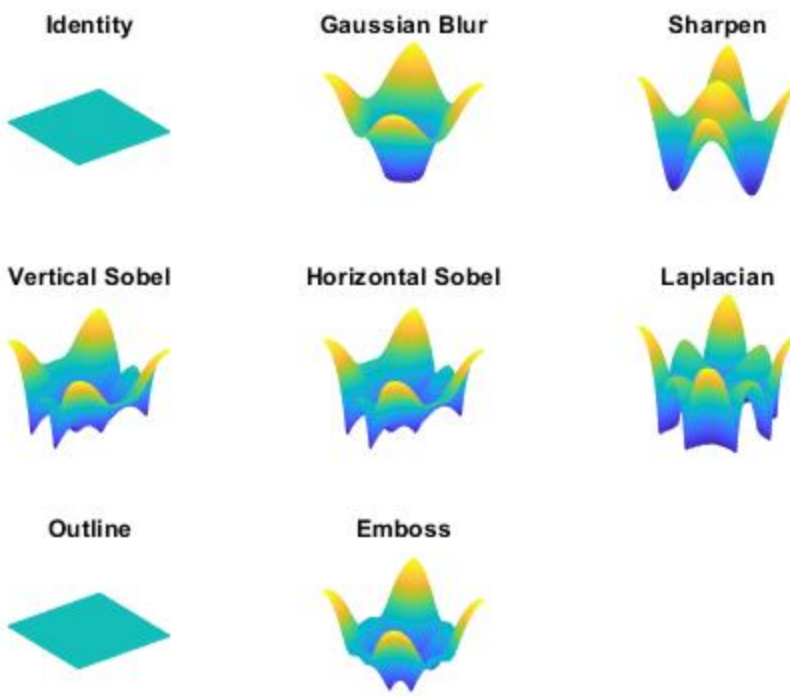


3. Plot the 2D and 3D Magnitude Spectrum of all the filters in #1.

```
%Plot the 2D Magnitude Spectrum of All Filters
%Initialize a function to calculate and plot the 2D magnitude spectrum
figure(2);
subplot(331); imgmagspec2D(Iden); title('Identity')
subplot(332); imgmagspec2D(Gblur); title('Gaussian Blur')
subplot(333); imgmagspec2D(Sharp); title('Sharpen')
subplot(334); imgmagspec2D(VertSob); title('Vertical Sobel')
subplot(335); imgmagspec2D(HorSob); title('Horizontal Sobel')
subplot(336); imgmagspec2D(Lap); title('Laplacian')
subplot(337);imgmagspec2D(Outl); title('Outline')
subplot(338); imgmagspec2D(Embo); title('Emboss')

%Plot the 3D Magnitude Spectrum of All Filters
%Initialize a function to calculate and plot the 3D magnitude spectrum
figure(3);
subplot(331); imgmagspec3D(Iden); title('Identity')
subplot(332); imgmagspec3D(Gblur); title('Gaussian Blur')
subplot(333); imgmagspec3D(Sharp); title('Sharpen')
subplot(334); imgmagspec3D(VertSob); title('Vertical Sobel')
subplot(335); imgmagspec3D(HorSob); title('Horizontal Sobel')
subplot(336); imgmagspec3D(Lap); title('Laplacian')
subplot(337);imgmagspec3D(Outl); title('Outline')
subplot(338); imgmagspec3D(Embo); title('Emboss')
```

Expected Output:

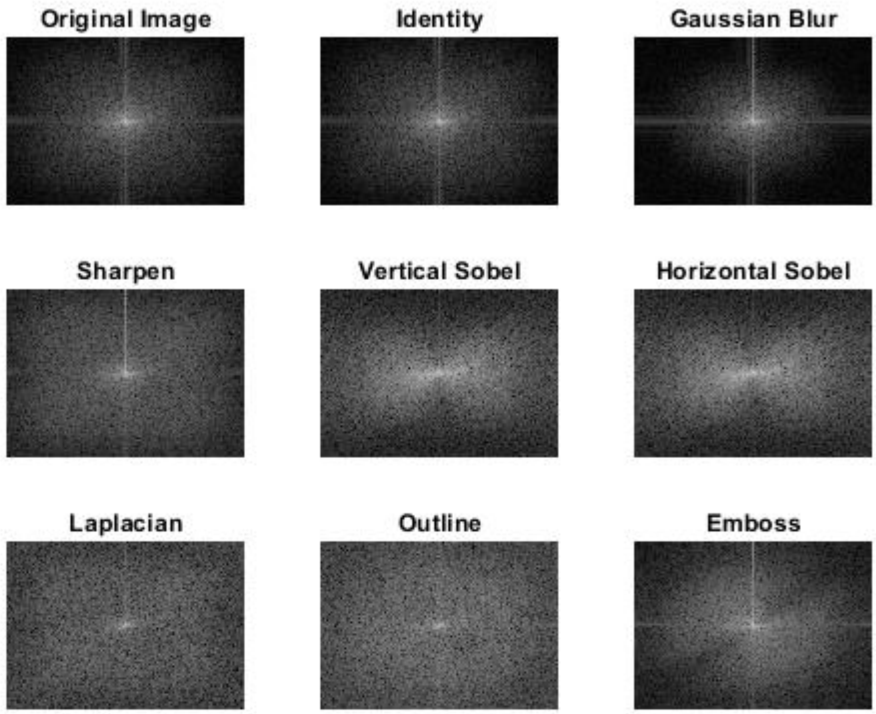


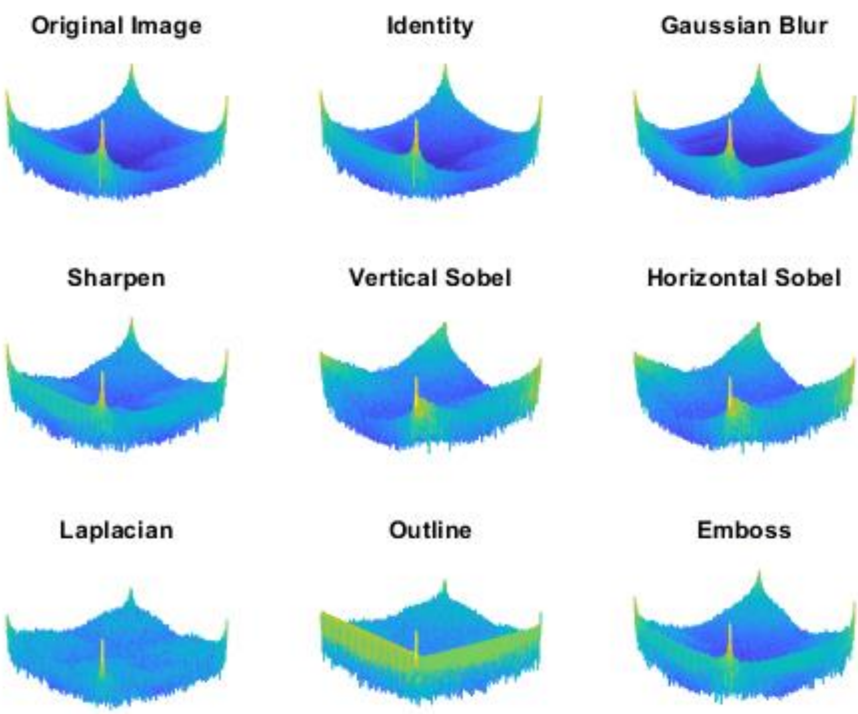


4. Plot the 2D and 3D Magnitude Spectrum of all the filtered images in #2.

```
%These are the functions used for the section. Note that this is placed at the
end of the file as prescribed by MATLAB’s convention in user-defined functions.

function imgmagspec2D(img) %image must be a convolution kernel or a grayscale
image
F = fftshift(fft2(double(mat2gray(img)), 512, 512));
Ilog = log(1+abs(F));
colormap(gray); imagesc(Ilog); axis off
end

function imgmagspec3D(img)
H=fftshift(fft2(double(mat2gray(img)), 512, 512));
Hlog=log(1+abs(H));
u=-256:255;
v=-256:255;
[u,v]=meshgrid(u,v);
mesh(u,v,[Hlog(257:512,257:512),Hlog(257:512,1:256);
Hlog(1:256,257:512) Hlog(1:256,1:256)]); axis off
end


%Plot the 2D Magnitude Spectrum of Filtered Images
figure(4);
subplot(331); imgmagspec2D(Igs); title('Original Image')
subplot(332); imgmagspec2D(Idenfil); title('Identity')
subplot(333); imgmagspec2D(Igbfil); title('Gaussian Blur')
subplot(334); imgmagspec2D(Ishfil); title('Sharpen')
subplot(335); imgmagspec2D(Ivsfil); title('Vertical Sobel')
subplot(336); imgmagspec2D(Ihsfil); title('Horizontal Sobel')
subplot(337); imgmagspec2D(Ilapfil); title('Laplacian')
subplot(338); imgmagspec2D(Ioutlfil); title('Outline')
subplot(339); imgmagspec2D(Iembofil); title('Emboss')

%Plot the 3D Magnitude Spectrum of Filtered Images
figure(5);
subplot(331); imgmagspec3D(Igs); title('Original Image')
subplot(332); imgmagspec3D(Idenfil); title('Identity')
subplot(333); imgmagspec3D(Igbfil); title('Gaussian Blur')
subplot(334); imgmagspec3D(Ishfil); title('Sharpen')
subplot(335); imgmagspec3D(Ivsfil); title('Vertical Sobel')
subplot(336); imgmagspec3D(Ihsfil); title('Horizontal Sobel')
subplot(337); imgmagspec3D(Ilapfil); title('Laplacian')
subplot(338); imgmagspec3D(Ioutlfil); title('Outline')
subplot(339); imgmagspec3D(Iembofil); title('Emboss')
```

Expected Output:





5. Using dog.png, filter the image using a 3x3 and 5x5 Gaussian Blur. Compare the results obtained when using approximate kernels and when using fspecial(). Show the filtered images in one figure and plot the 3D magnitude spectrum in another figure. Use subplot for both instances.

```
%Initialize Gaussian Kernels. Use the approximate kernels and
%kernels obtained from fspecial
G3x3 = (1/16)*...
        [1 2 1;...
        2 4 2;...
        1 2 1]


G5x5 = (1/273)*[1 4 7 4 1;...
                4 16 26 16 4;...
                7 26 41 26 7;...
                4 16 26 16 4;...
                1 4 7 4 1]
G3x3f = fspecial("gaussian", [3 3], 1)
G5x5f = fspecial("gaussian", [5 5], 1)
%Filter image using approximate 3x3 and 5x5 kernels,
%and 3x3 and 5x5 gaussian filter functions
Ig3x3app = uint8(imfilter(Igs, G3x3));
Ig5x5app = uint8(imfilter(Igs, G5x5));
Ig3x3f = uint8(imfilter(Igs, G3x3f));
Ig5x5f = uint8(imfilter(Igs, G5x5f));


%Show filtered images in one figure
figure(6);
subplot(221); imshow(Ig3x3app); title('Approximate 3x3 Gaussian Kernel')
subplot(222); imshow(Ig5x5app); title('Approximate 5x5 Gaussian Kernel')
subplot(223); imshow(Ig3x3f); title('Exact 3x3 Gaussian Kernel')
subplot(224); imshow(Ig5x5f); title('Exact 5x5 Gaussian Kernel')


%Plot the 3D magnitude spectrum of the images in one figure
figure(7);
subplot(221); imgmagspec3D(G3x3);  title('Approximate 3x3 Gaussian Kernel')
subplot(222); imgmagspec3D(G5x5); title('Approximate 5x5 Gaussian Kernel')
subplot(223); imgmagspec3D(G3x3f); title('Exact 3x3 Gaussian Kernel')
subplot(224); imgmagspec3D(G5x5f); title('Exact 5x5 Gaussian Kernel')
```

Expected Output:

```
G3x3 = 3×3
      0.0625    0.1250    0.0625
      0.1250    0.2500    0.1250
      0.0625    0.1250    0.0625

G5x5 = 5×5
      0.0037    0.0147    0.0256    0.0147    0.0037
      0.0147    0.0586    0.0952    0.0586    0.0147
      0.0256    0.0952    0.1502    0.0952    0.0256
      0.0147    0.0586    0.0952    0.0586    0.0147
      0.0037    0.0147    0.0256    0.0147    0.0037

G3x3f = 3×3
      0.0751    0.1238    0.0751
      0.1238    0.2042    0.1238
      0.0751    0.1238    0.0751

G5x5f = 5×5
      0.0030    0.0133    0.0219    0.0133    0.0030
      0.0133    0.0596    0.0983    0.0596    0.0133
      0.0219    0.0983    0.1621    0.0983    0.0219
      0.0133    0.0596    0.0983    0.0596    0.0133
      0.0030    0.0133    0.0219    0.0133    0.0030
```

Approximate 3x3 Gaussian Kernel

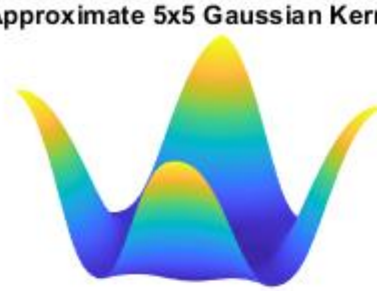
Approximate 5x5 Gaussian Kernel

Exact 3x3 Gaussian Kernel

Exact 5x5 Gaussian Kernel

Approximate 3x3 Gaussian Kernel

Approximate 5x5 Gaussian Kernel



Exact 3x3 Gaussian Kernel

Exact 5x5 Gaussian Kernel