I M A G E

Image Processing

EE202

SIGNAL

&

SYSTEM

Digital Image Representation

An image may be defined as a two-dimensional function f (x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates is called the intensity of the image at that point.
The smallest unit of image is called pixel and each pixel correspond to co-ordinate value of image.

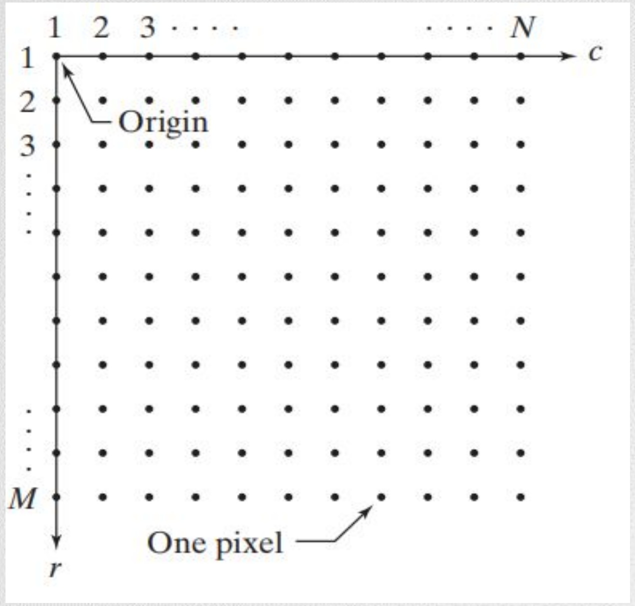
Image can be divided into three type:

- **Binary image:**
  Each pixel has value corresponding to either '0' or '1'. '0' indicate a black pixel and '1' indicate a white pixel. Combination of 0's and 1's form a complete image.

- **Grey Scale image**:
  A grayscale image is one in which the value of each pixel is a single sample representing only an amount of light (carries only intensity information).

- **R(red), G(green), B(blue) images:**
  An RGB image is basically a M*N*3 array of colour pixel, where each colour pixel is associated with three value which correspond to red, blue and green colour component of RGB image at a specified spatial location.

- **Indexed images:**
  An indexed image is basically a M*N array of number, where each number contain detail of further RGB value of corresponding pixel.



$$f = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix}$$

Images can be represented in matrix form and each cell of the matrix is representing image pixel, where M and N are number of rows and column respectively.

An image may be continuous with respect to the x- and y-coordinates, and also in amplitude. Converting such an image to digital form requires that the coordinates, as well as the amplitude, be digitized. Digitizing the coordinate values is called sampling; digitizing the amplitude values is called quantization. Thus, when x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image.

APPLICATION of IMAGE PROCESSING

Digital image processing has very wide applications and almost all of the technical fields are impacted by DIP.

- Adjustment of spatial resolution, increasing/decreasing brightness, enhancement of images are done by DIP.
- Our human eye can only see the visible portion, but a camera can see the other things that a naked eye is unable to see. The analysis of X rays, gamma rays, etc is done in digital image processing.
- Image sharpening and restoration, medical field, Remote sensing, machine vision, pattern recognition, color processing, microscopic imaging, transmission and encoding etc.

MATLAB TOOLBOX for IMAGE PROCESSING

- imread('filename')

  Images are read into the MATLAB environment using function imread. Here, filename is a string containing the complete name of the image file (including any applicable extension).

- imshow(f)

  Images are displayed on the MATLAB desktop using function imshow, where f is an image array.

- size(f) -- gives the row and column dimensions of an image.
- figure -- To display several images at same time.
- imfinfo('filename') -- Provide detail of corresponding file.
- impixelinfo('filename') -- provide detail of each pixel of image.

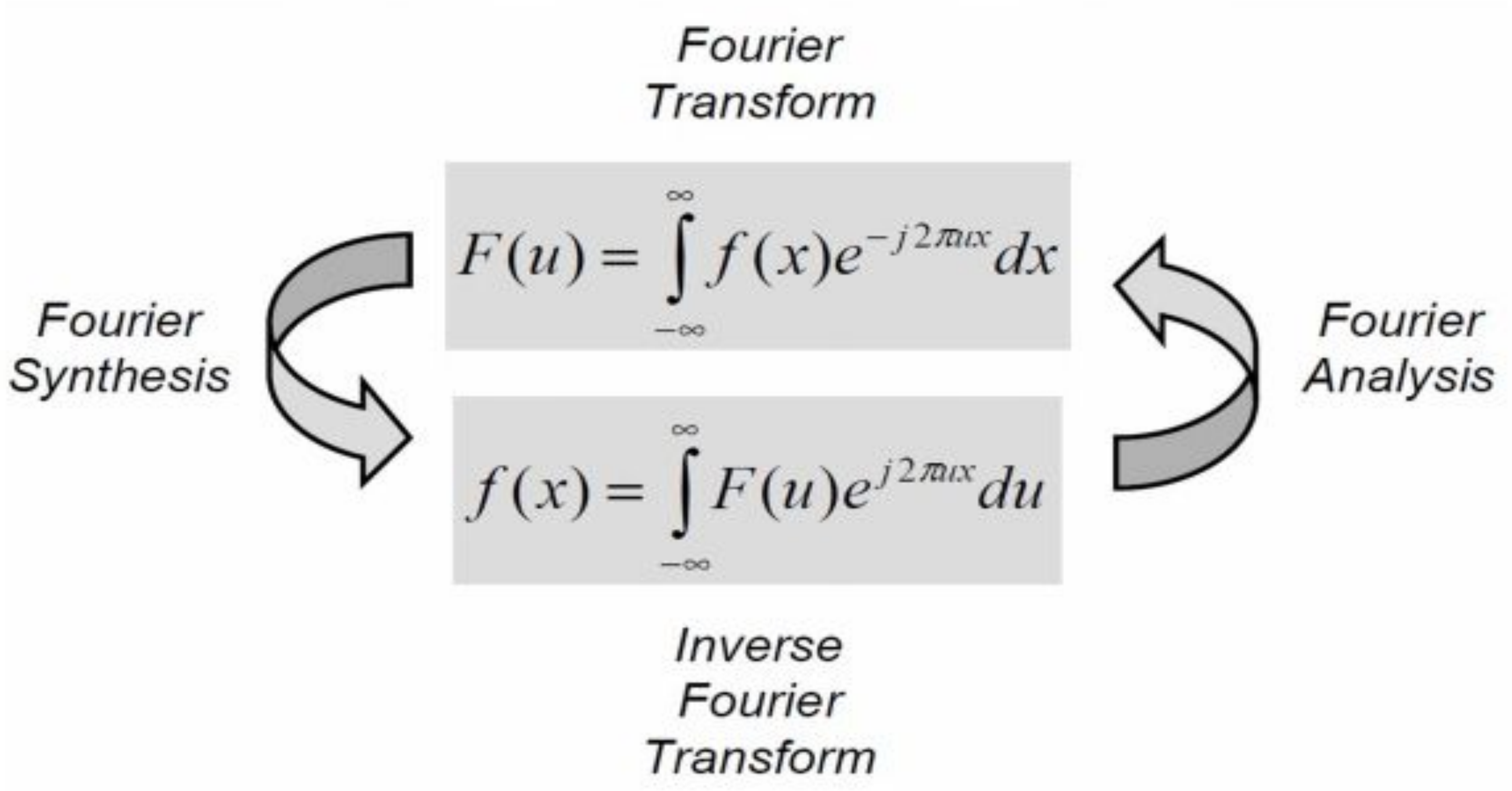
- gray2rgb(f) -- convert gray image to RGB image.
- fft(f) -- Fourier transfer of a vector.
- fft2(f) -- Fourier transfer of a matrices.
- ifft(f) -- Inverse Fourier transfer of a vector.
- ifft2(f) -- Inverse Fourier transfer of a matrices.
- fftshift(f) -- Rearranges a Fourier transform f by shifting the zero-frequency component to the center of the array.

Fourier Transform in Image Processing

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.

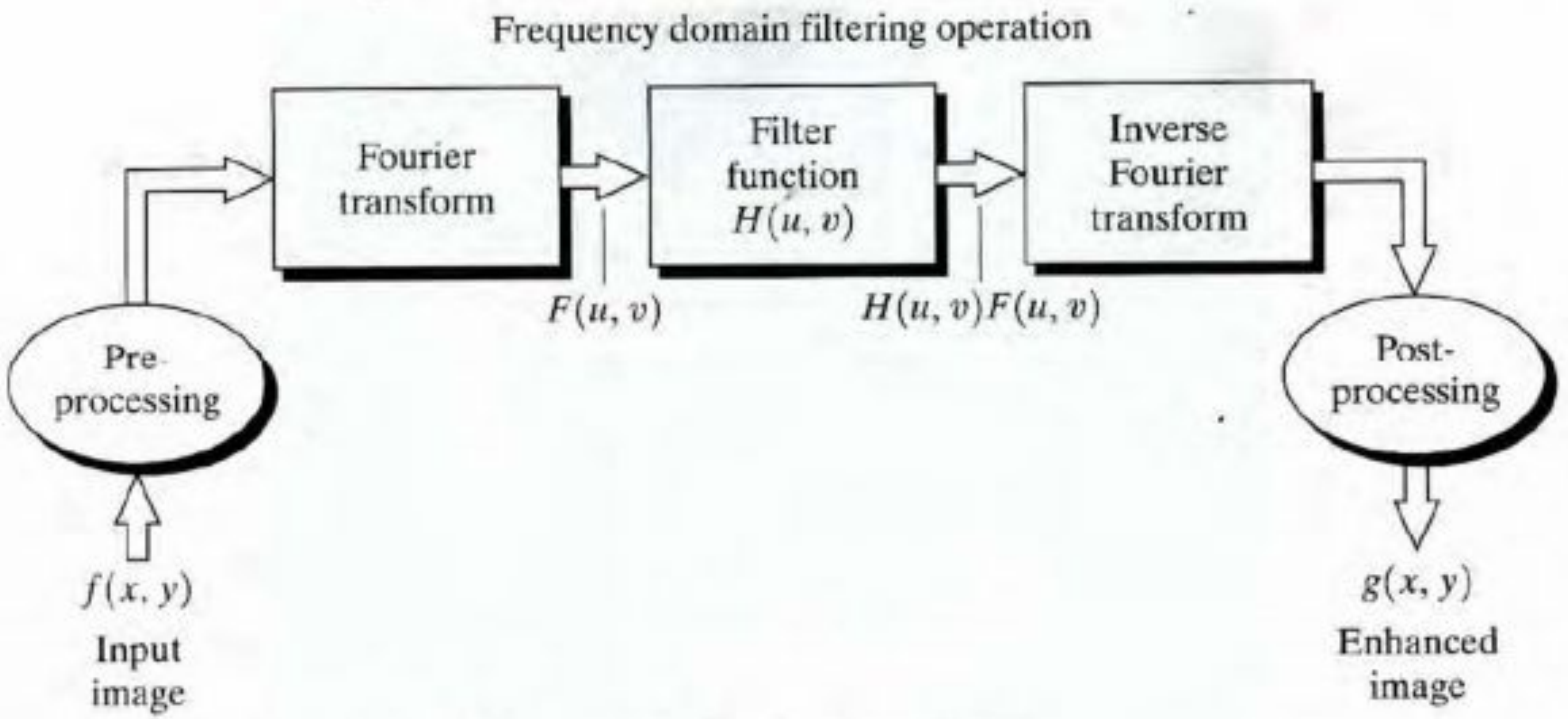
The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression.

Space and Frequency



$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$



Filtering in frequency Domain

Discrete Fourier Transformation formulae used.

- F(x,y) for fft2 command.
- f(m,n) for ifft2 command.

$$F(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi(x\frac{m}{M}+y\frac{n}{N})}$$

$$f(m,n) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(x,y)e^{j2\pi(x\frac{m}{M}+y\frac{n}{N})}$$

High pass Filter on Image

- High pass Filter (HPF) is used for image sharpening in the frequency domain. Image Sharpening is a technique to enhance the fine details and highlight the edges in a digital image. It removes low-frequency components from an image and preserves high-frequency components.
- A high pass filter attenuate the frequency below specified frequency. Allow frequencies higher than the specified frequency.
- High pass filter function's code:
  - function[c] = highpass(im)
  - h = size (im,1);
  - w = size(im,2);
  - [x,y] =mesgrid(-floor(w/2):floor((w-1)/2),-floor(h/2):floor((h-1)/2));

  - z=sqrt(x.^2+y.^2);
  - c= z>30; (cutoff frequency)

Steps to Apply Highpass Filter on Image

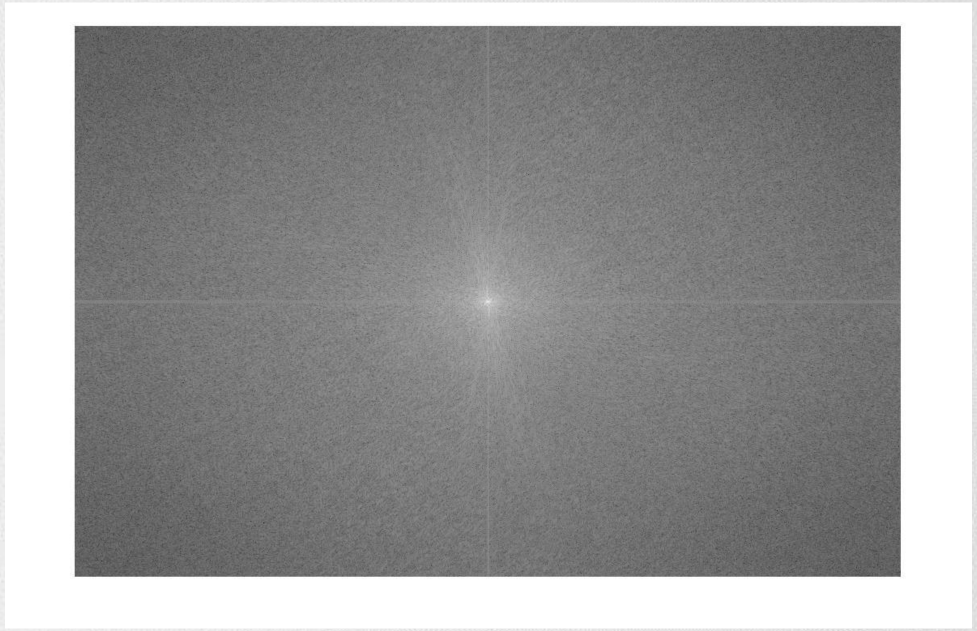
code:-

```
a=imread('A:/mouse/lion.jpg');

imshow(a)
```

Step 3: Get the Fourier Transform of the input_image

```
af=fftshift(fft2(a));
fftshow(af)
function [] = fftshow(f)
f1=log (1+abs(f));
fm = max(f1(:));
figure, imshow(f1/fm);
end
```



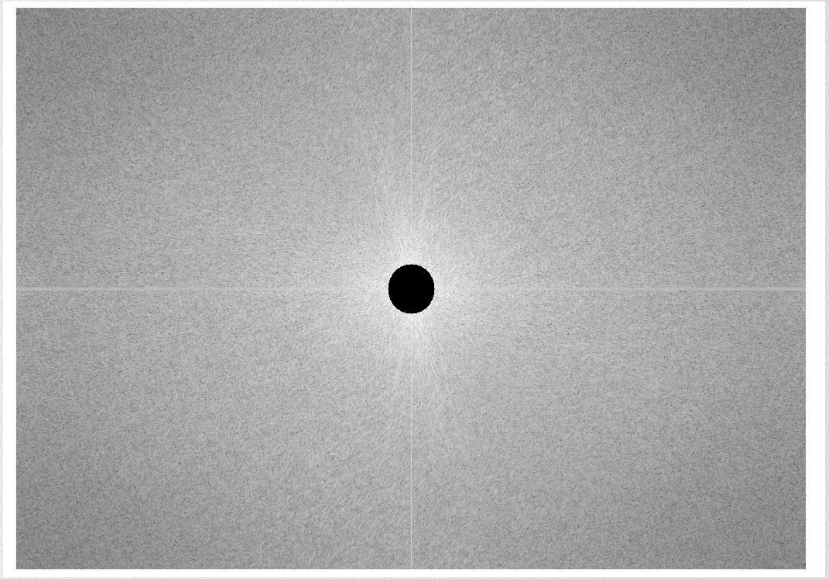
Step 4: Assign the Cut-off Frequency

Step 5: Designing filter: Ideal High Pass Filter

Step 6: Convolution between the Fourier Transformed input image and the filtering mask.

```
g= highpass(af);
conv=af.*g;
fftshow(conv)
```

```
function [ c ] = highpass(im)
 h=size(im,1);
w=size(im,2);
 [x,y] = meshgrid(-floor(w/2):floor((w-1)/2),-floor (h/2):floor((h-1)/2));
z= sqrt(x.^2+y.^2);
c= z>30;   % cutoff frequency
end
```



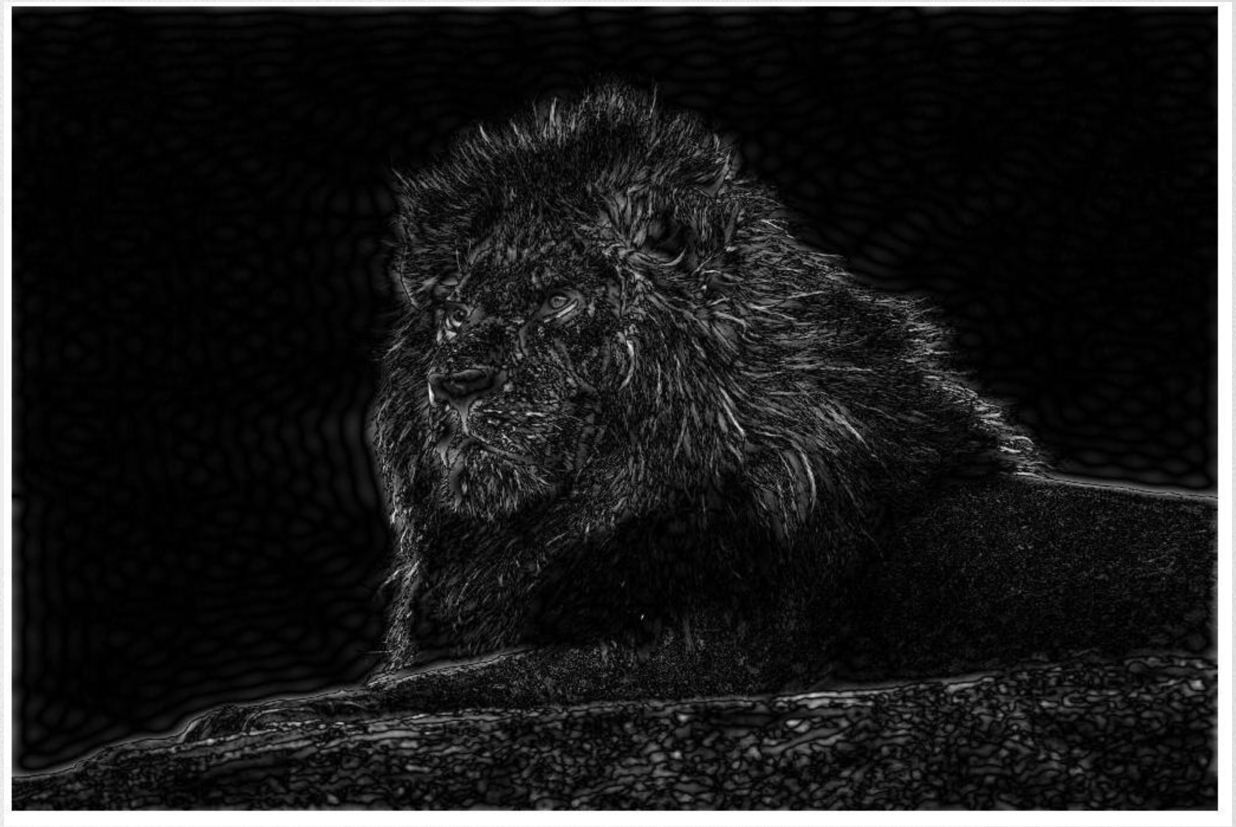
Step 7: Take Inverse Fourier Transform of the convoluted image

Step 8: Display the resultant image as output

```
convi=ifft2(conv);

ifftshow(convi)


Function [] = ifftshow(f)
f1 = abs(f);
fm = max(f1(:));
figure, imshow(f1/fm);
end
```



Here low frequency components(which are lower than the cutoff frequency) has been attenuated completely.

EE202

THANK YOU

ARJIT GIRI 190002006

SATYAM GUPTA 190002057

RAVI VERMA 190002052