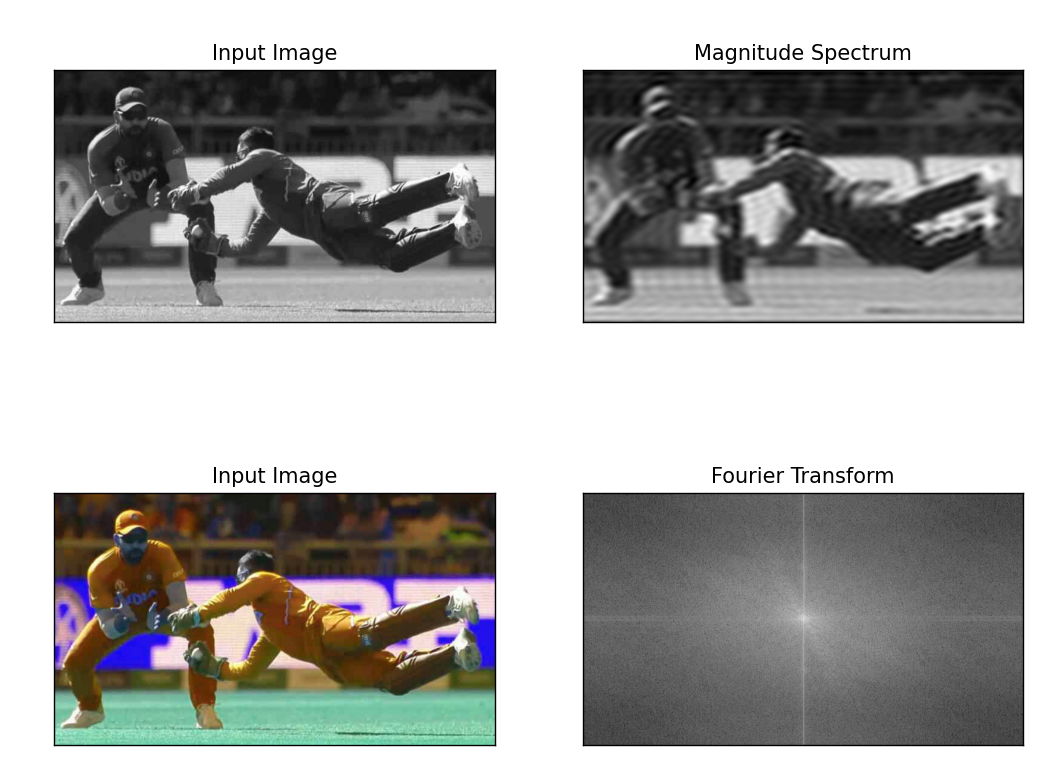
**Source Code :**

import cv2  
import numpy as np

from matplotlib import pyplot as plt  
  
# now we will be loading the image and converting it to grayscale  
image = cv2.imread(r"Dhoni-dive\_165121\_730x419-m.jpg")  
  
gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
  
# Compute the discrete Fourier Transform of the image  
fourier = cv2.dft(np.float32(gray), flags=cv2.DFT\_COMPLEX\_OUTPUT)  
  
# Shift the zero-frequency component to the center of the spectrum  
fourier\_shift = np.fft.fftshift(fourier)  
  
# calculate the magnitude of the Fourier Transform  
magnitude = 20\*np.log(cv2.magnitude(fourier\_shift[:,:,0],fourier\_shift[:,:,1]))  
  
# Scale the magnitude for display  
magnitude = cv2.normalize(magnitude, None, 0, 255, cv2.NORM\_MINMAX, cv2.CV\_8UC1)  
  
image\_path = r"Dhoni-dive\_165121\_730x419-m.jpg"  
image2 = cv2.imread(image\_path, 0)  
  
# calculating the discrete Fourier transform  
DFT = cv2.dft(np.float32(image2), flags=cv2.DFT\_COMPLEX\_OUTPUT)  
  
# reposition the zero-frequency component to the spectrum's middle  
shift = np.fft.fftshift(DFT)  
row, col = image2.shape  
center\_row, center\_col = row // 2, col // 2  
  
# create a mask with a centered square of 1s  
mask = np.zeros((row, col, 2), np.uint8)  
mask[center\_row - 30:center\_row + 30, center\_col - 30:center\_col + 30] = 1  
  
# put the mask and inverse DFT in place.  
fft\_shift = shift \* mask  
fft\_ifft\_shift = np.fft.ifftshift(fft\_shift)  
imageThen = cv2.idft(fft\_ifft\_shift)  
  
# calculate the magnitude of the inverse DFT  
imageThen = cv2.magnitude(imageThen[:,:,0], imageThen[:,:,1])  
  
# visualize the original image and the magnitude spectrum  
plt.figure(figsize=(10,10))  
plt.subplot(221), plt.imshow(image2, cmap='gray')  
plt.title('Input Image'), plt.xticks([]), plt.yticks([])  
plt.subplot(222), plt.imshow(imageThen, cmap='gray')  
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])  
plt.subplot(223), plt.imshow(image, cmap='gray')  
plt.title('Input Image'), plt.xticks([]), plt.yticks([])  
plt.subplot(224), plt.imshow(magnitude, cmap='gray')  
plt.title('Input Image'), plt.xticks([]), plt.yticks([])  
plt.show()

**Output :**

****