

CV Assignment-01

Meghana.K(S20160010040)

K. Sai Prasanna Kumar(S20160010047)

Navneeth.R(S20160010071)

Q1) HYBRID IMAGES:

A Hybrid image is an image that perceived in one of two different ways, depending on viewing distance based on the way humans process visual input. Hybrid image combine the low spatial frequencies of one image with high spatial frequencies with another image producing image with an interpretation that changes with viewing distance.

To read image:-

```
img1 = plt.imread('E:/STUDY/ACADEMIC/6th sem/CV/assign1/HW1_Q1/bird.bmp')
img2 = plt.imread('E:/STUDY/ACADEMIC/6th sem/CV/assign1/HW1_Q1/plane.bmp')
```

Gaussian Matrix:-

```
def gaussian_filter(rows,columns,sigma):
    if rows%2 == 0:
        row_center = rows/2
    else:
        row_center = rows/2 + 1

    if columns%2 == 0:
        col_center = columns/2
    else:
        col_center = columns/2 + 1
    def gaussian(i,j):
        return math.exp(-1.0 * ((i - row_center)**2 + (j - col_center)**2) / (2 * sigma**2))
    gaussian_array = np.zeros((rows,columns))
    for i in range(0,rows):
        for j in range(0,columns):
            gaussian_array[i][j] = gaussian(i,j)
    return gaussian_array
```

1. Gaussian filter function takes rows, columns and standard deviation as inputs and returns a 2D gaussian matrix
2. Find center of the image i.e., row_center and col_center.
3. Compute values of each cell using gaussian distribution function given below.

$$g(x, y) = e^{-\frac{(x-a)^2 + (y-b)^2}{2\sigma^2}}$$

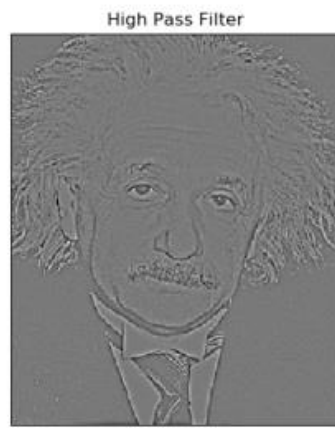
4. x,y represents position of pixel. a,b represents center. σ represents standard deviation.

5. After computing values store them in a matrix called gaussian matrix and return that matrix.

Low pass and high filter:-

```
def Apply_Fitler(image,flag,sigma):
    n,m = image.shape
    if flag == True:                # High pass
        gaussian_matrix = (1 - gaussian_filter(n,m,sigma))
    else:                            #Low pass
        gaussian_matrix = gaussian_filter(n,m,sigma)
    dft = np.fft.fft2(image)
    dftshift = np.fft.fftshift(dft)
    filterImage = dftshift * gaussian_matrix
    ifftshift = np.fft.ifftshift(filterImage)
    ifftImage = np.fft.ifft2(ifftshift)
    plt.imshow(np.real(ifftImage),cmap = 'gray')
    if flag == True:
        plt.title('High Pass Filter'),plt.xticks([]),plt.yticks([])
    else:
        plt.title('Low Pass Filter'),plt.xticks([]),plt.yticks([])
    plt.show()
    return np.real(ifftImage)
```

1. Take image and standard deviation as input.
2. image.shape returns shape of the image mxn.
3. gaussian_filter(n,m,sigma)[g(x,y)] returns a gaussian matrix as discussed above.
4. np.fft.fft2() function computes 2-D Fast Fourier transform of given image.
5. np.fft.fftshift() Shifts zero-frequency terms to the center of the array. For two-dimensional input, swaps first and third quadrants, and second and fourth quadrants.
 - a. **Low Pass Filter:-** Multiply shifted FFT of image with gaussian matrix g(x,y) to smooth the given image in frequency domain.
 - b. **High Pass Filter:-** Multiply shifted FFT of image with gaussian matrix to sharpen the given image in frequency domain but here matrix is gaussian_matrix = (1-g(x,y)).
6. Now compute Inverse Fast Fourier transform to get smoothen image and sharpened image and return it.



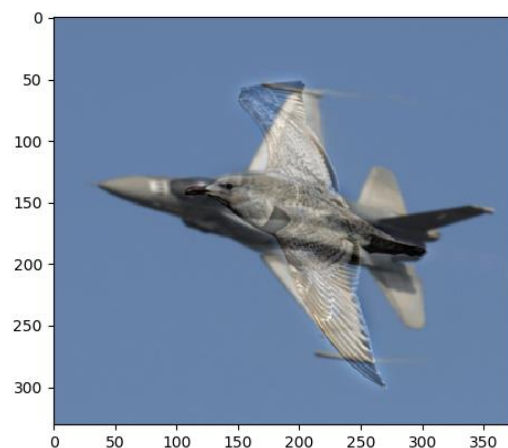
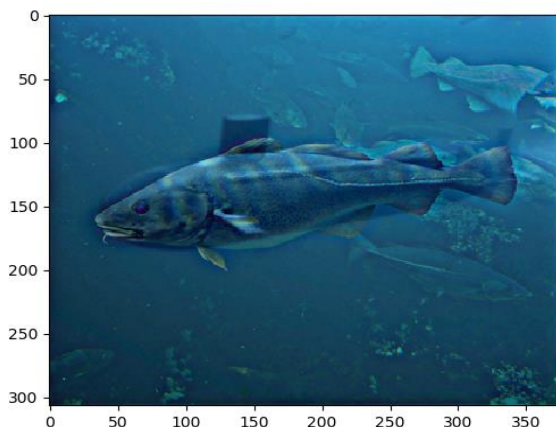
Main function:-

```
def hybrid_image(highpassImg, lowpassImg, alpha, beta):
    lowpass = Apply_Fitler(lowpassImg, False, beta)
    misc.imwrite("result/Marilyn-filtered.png", np.real(lowpass))
    highpass = Apply_Fitler(highpassImg, True, alpha)
    misc.imwrite("result/einstein-filtered.png", np.real(highpass))
    return np.real(highpass + lowpass)
```

After applying Filters sum up high frequency information and low frequency information to get Hybrid image of two images.

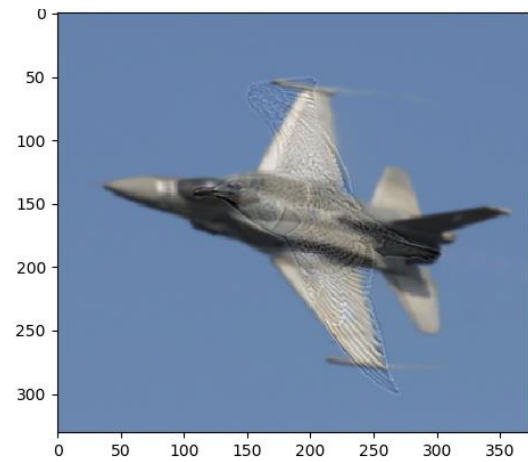
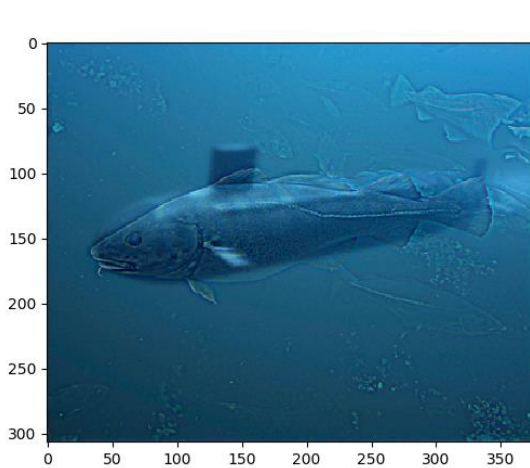
Analysis of the hybrid image with varying sigma for low pass and high pass :

I. High pass(alpha = 10) , Low pass(beta = 50):



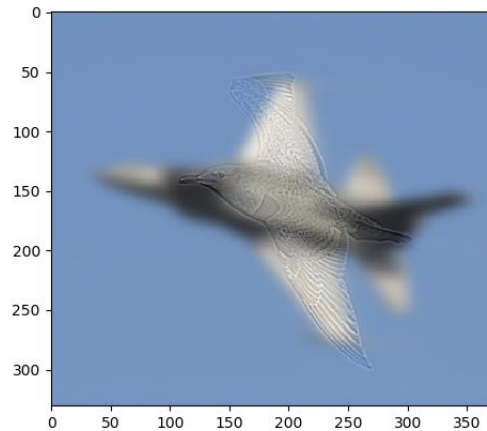
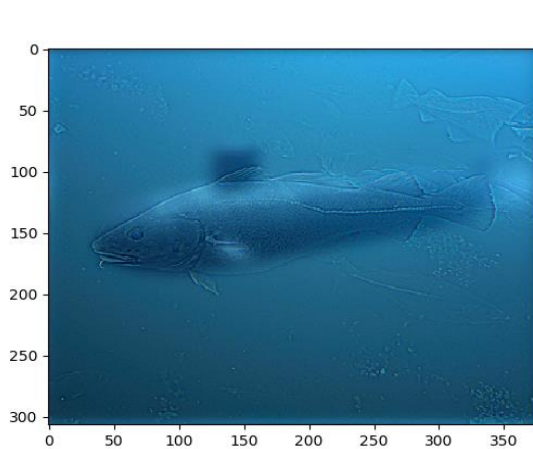
- Since α is low, the high pass filter allows most of the high frequency components and also since β is high, the image B is more blurred. Hence edges from Image A will be more prominent.

II. High pass($\alpha = 40$) , Low pass($\beta = 40$):



- Both α and β are equal, hence the high frequencies from Image A and low frequencies from Image B are equally preserved.

III. High pass($\alpha = 50$) , Low pass($\beta = 10$):



- Since α is high, the high pass filter rejects most of the frequencies, only strong edges are preserved and also since β is low, the image B is not much blurred. Hence Image B will be more prominent.