# Homework Assignment 2

# CS696, applied computer vision

## Algorithm:

The method my_imfilter() is able to filter any given image with its dimensions as Odd Height x Odd Width with the specified matrix filter.
It involves a couple of steps:

- It initially checks the dimensions of the image to see if the image is a colored image or a grayscale image.
- If the image is a grayscale image, it will call the my_fil function once else it will call the function each time for one of its channels.
- The my_fil function first pads the original image with the size of the half of the filter height and width respectively since it uses up those pixels to calculate the values (sum of the multiplied values) in the border of the original image.
- After padding, it will iterate sub-matrices of the size of the filter one by one, calculating the value of the center pixel.
- The value of the center pixel is then used as the output or the filtered image.

## Description:

After the algorithms checks for the colored/grayscale image, it determines the pad-size for the input image. It is modified such that the center pixel of the filter overlaps with the each pixel of the input image. The pad-size is set to floor value of the (height of filter)/2 and (width of filter)/2 respectively. The padded image is assigned to a new variable. To calculate the output, the padded pixel image is passed through 2 for loops starting from the top left corner. The 2 loops iterate the center pixels of the filter along with sub matrices from the padded matrix. The sum of the product of all elements of both the matrices are taken and stored in the output matrix. This entire process works thrice, one for each channel if the input image is a colored image and works once, if the input image is a grayscale image.

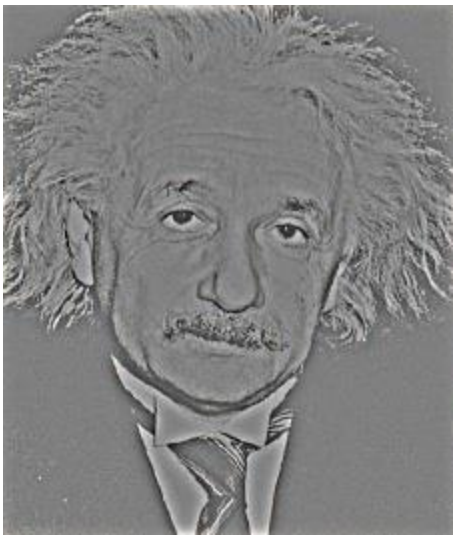
The function also uses a second parameter the filter. The filter are passed along with the combination of a cut-off frequency. The cut-off frequency can be altered to get different intensities of the output filtered image.
If the cut-off frequency is too high for the image, the output image will result in most of the features of the input image being filtered out. If the cut-off frequency is too low, the output image with retain most of the features from the original image. We can either use 2 different cut-off frequencies like the examples or just use the same frequency for both the results.

Examples:

1) Marilyn Monroe and Albert Einstein
   Here, two different cut-off frequencies are used for both of the images, namely 4 and 7
   The Low frequency image of the Marilyn results in a blurred out image, this can be considered as the base of the hybrid image.

**Low Frequency: Marilyn**

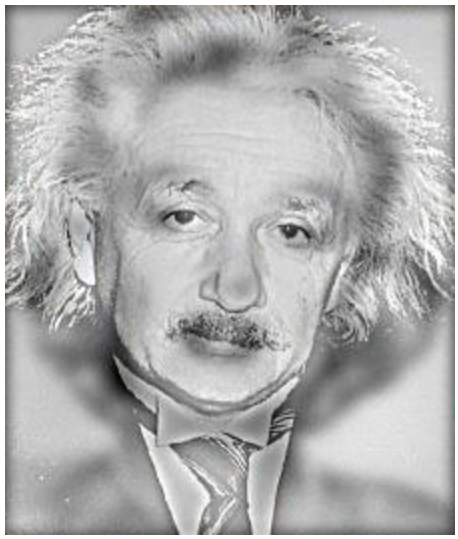
The high frequency image of the albert Einstein results in a detailed image of the edges, this is the image that has the features shown vividly in the hybrid image.



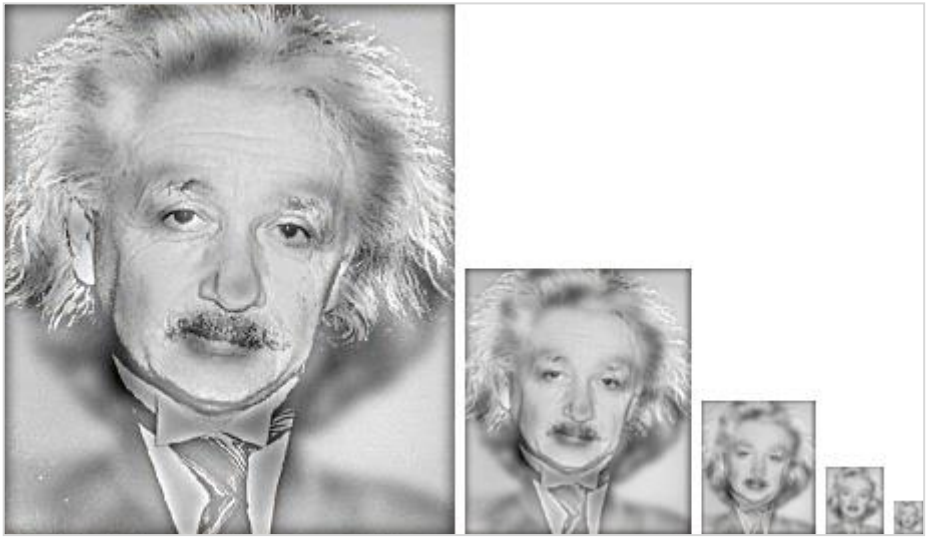
**High Frequency: Albert**

After merging both the low frequency image and the high frequency image, it results in the following output:

**Hybrid Image**



**Hybrid Image Scale**



Another Example of my own with same cut-off frequencies (5):

**Low Frequency**

**High Frequency**

**Hybrid Image**

**Hybrid Image Scaled**