1 Hybrid image

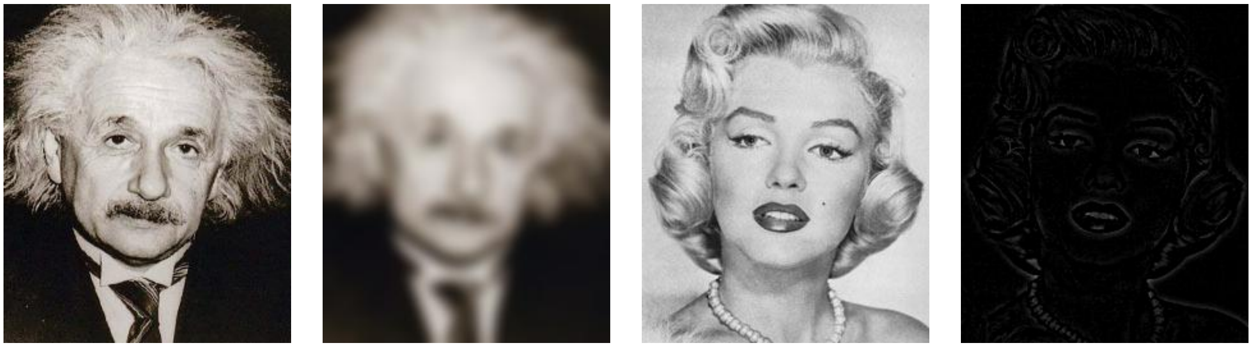
A hybrid image is a combination of a low-pass filtered (i.e. blurry) image and a high-pass filtered (i.e. sharp) image. Recall that one can obtain a sharp image by subtracting the blurry version of an image from itself. Mathematically this can be written as $I = blurry(I) + sharp(I)$. Thus a hybrid image of I_1 and I_2 can be obtained as:

$$I_{hybrid} = blurry(I_1, \sigma_1) + sharp(I_2, \sigma_2) = I_1 * g(\sigma_1) + I_2 - I_2 * g(\sigma_2). \tag{1}$$

Here, $g(\sigma_1)$ and $g(\sigma_2)$ are Gaussian filters with standard deviations σ_1 and σ_2 and * denotes the filtering operator. Figure 1 shows the result of filtering with Gaussians of two different σ values.

I used three methods to perform the filtering namely imfilter,imgaussfilt,myfilter. myfilter is slower method to perform filtering but it i's functionality is similar to imfilter.

Function HybridImage accepts following filtering methods namely ('imfilter','imgaussfilt','myfilter')and performs filtering based on type of specified method. It returns hybrid Image, Blurred image and Sharpened Image



Einstein image

$\sigma = 5$

Marilyn image

$\sigma = 3$

Figure 1: Effect of filtering with a Gaussian. The bigger the sigma the more blurry it is.

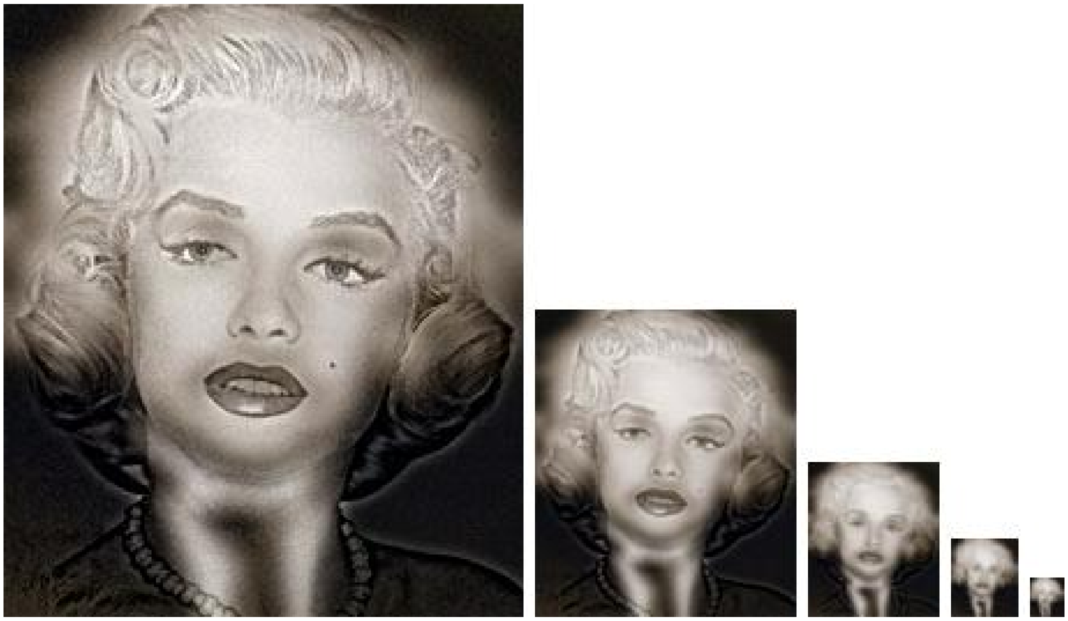


Figure 2: Hybid image of the Einstein and Marilyn. The large image looks like the Marilyn while the small image looks like the Einstein. The image was created with $\sigma_1 = 5$ and $\sigma_2 = 3$.

- Pad array to handle missing information in image. Mirror the image content over the boundaries for padding.
- Transpose filter and convert each w by h block from the image into a column, using im2col function.
- Calculate dot product of filter and sliding array from image.
- Convert columns back into image and return the result, using col2im function.

filtering method: myfilter

```
[filter_row, filter_column] = size(filter);
 r_margin = (filter_row - 1) / 2;
 c_margin = (filter_column - 1) / 2;
  %Loop through each color channel
 for layer = 1:size(image, 3)
            paddedImage = padarray(image(:, :, layer), [r_margin c_margin], ’symmetric’);
            filtered = filter(:)’*im2col(paddedImage, size(filter), ’sliding’);
            result(:,:,layer) = col2im(filtered, [1 1], [size(image, 1) size(image, 2)]);
        end
```