HOMEWORK 1 ANSWERS WRITE UP

PART 1

<u>QUESTION:</u> Please provide three hybrid image results. For one of your favorite results, show

- (a) the original and filtered images
- (b) the hybrid image and hybrid_image_scale (generated using `vis_hybrid_image.m` provided in the starter code)
- (c) log magnitude of the Fourier transform of the two original images, the filtered images, and the hybrid image

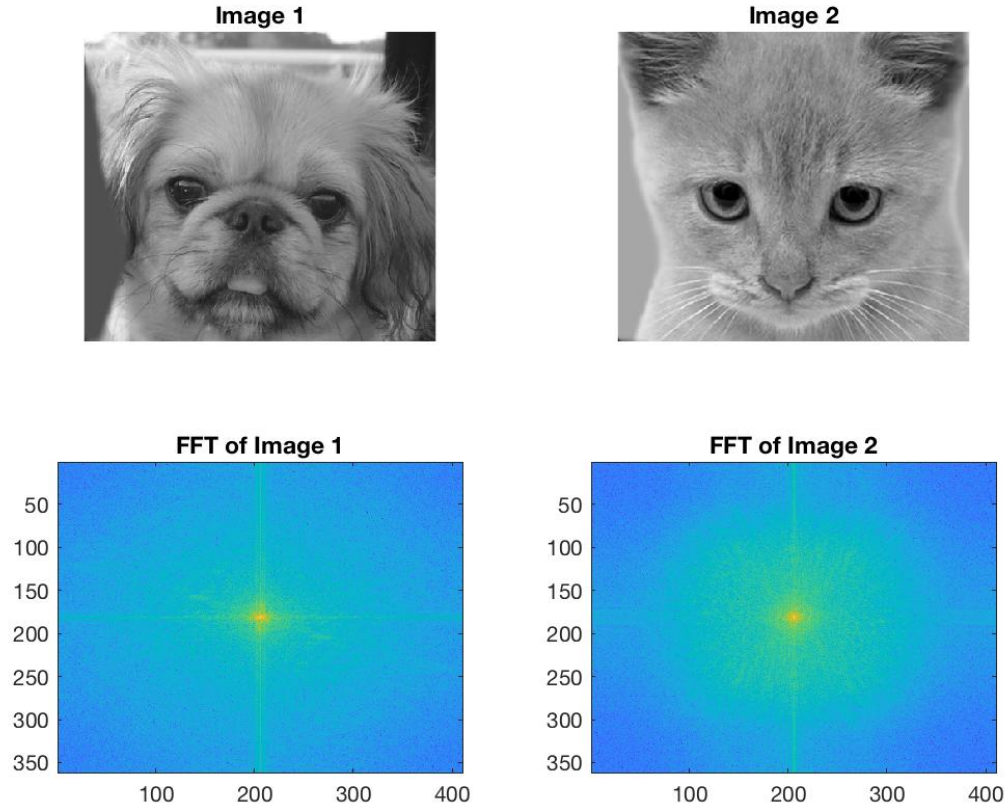
Briefly (a few sentences) explain how it works, using your favorite results as illustrations.

<u>Answer:</u>

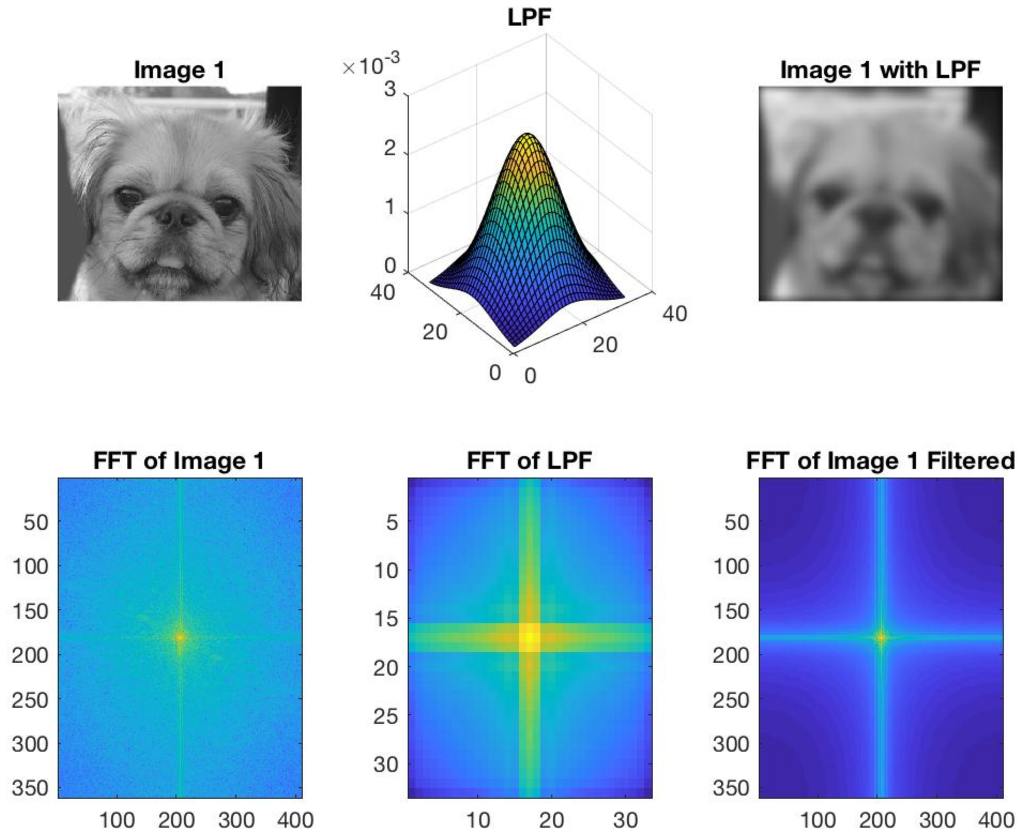
3 Hybrid Image Results:

Result 1

Original Images and FFTs



Filtered Images and FFTs



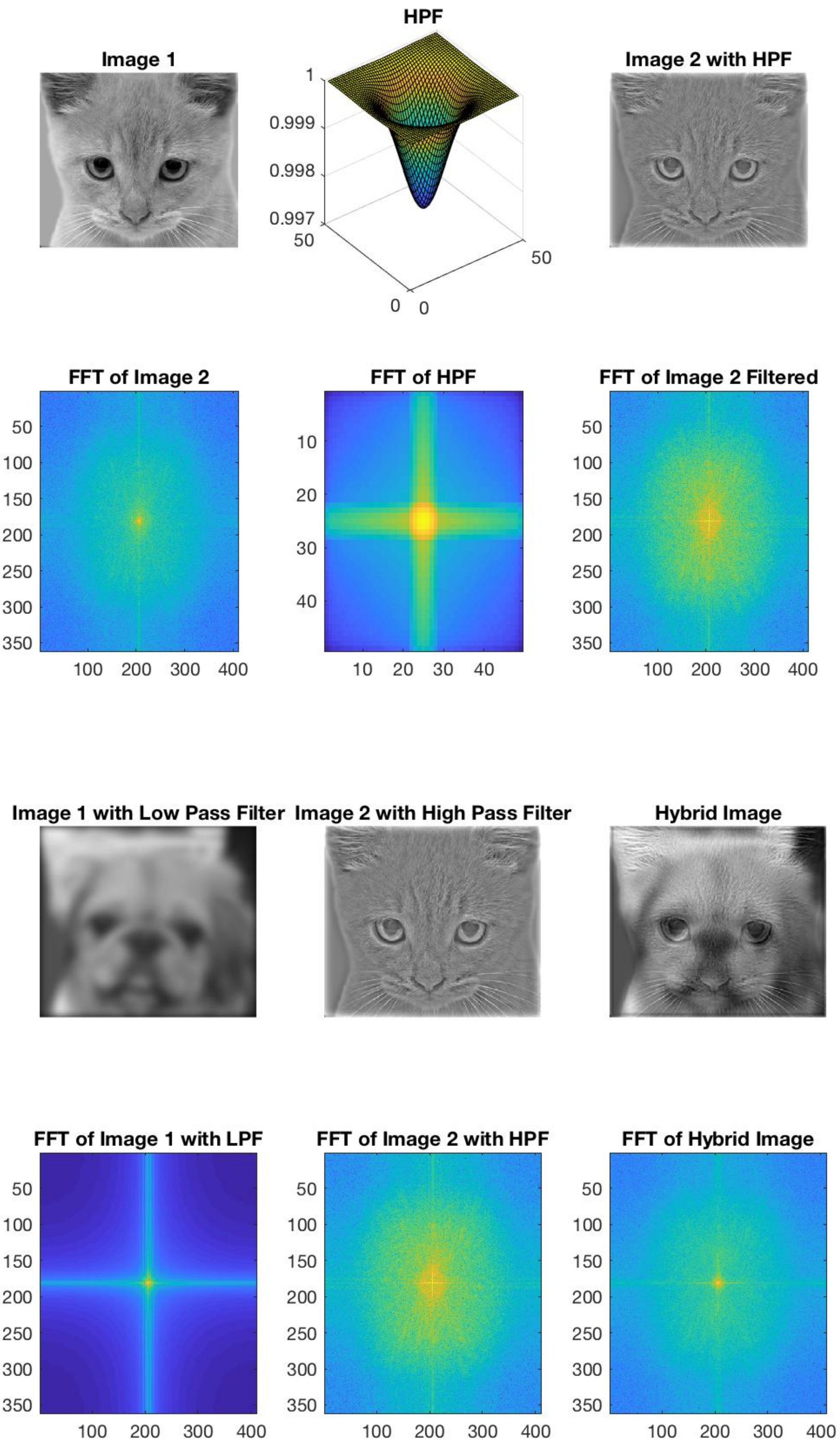
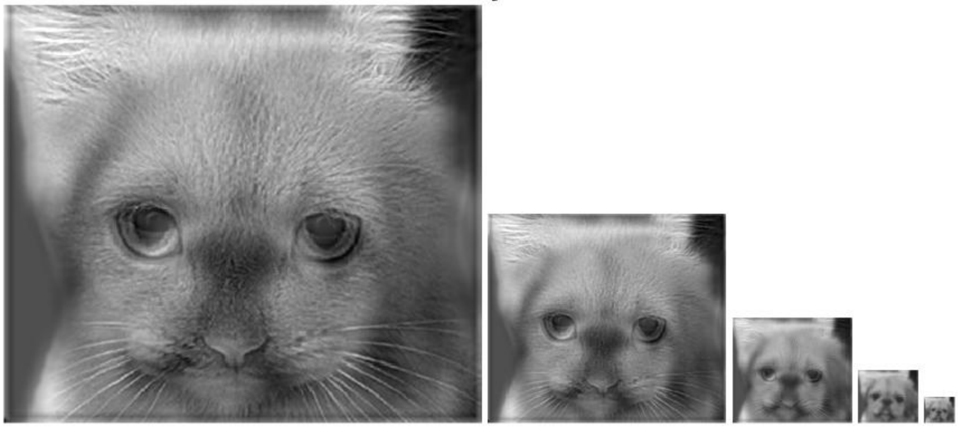
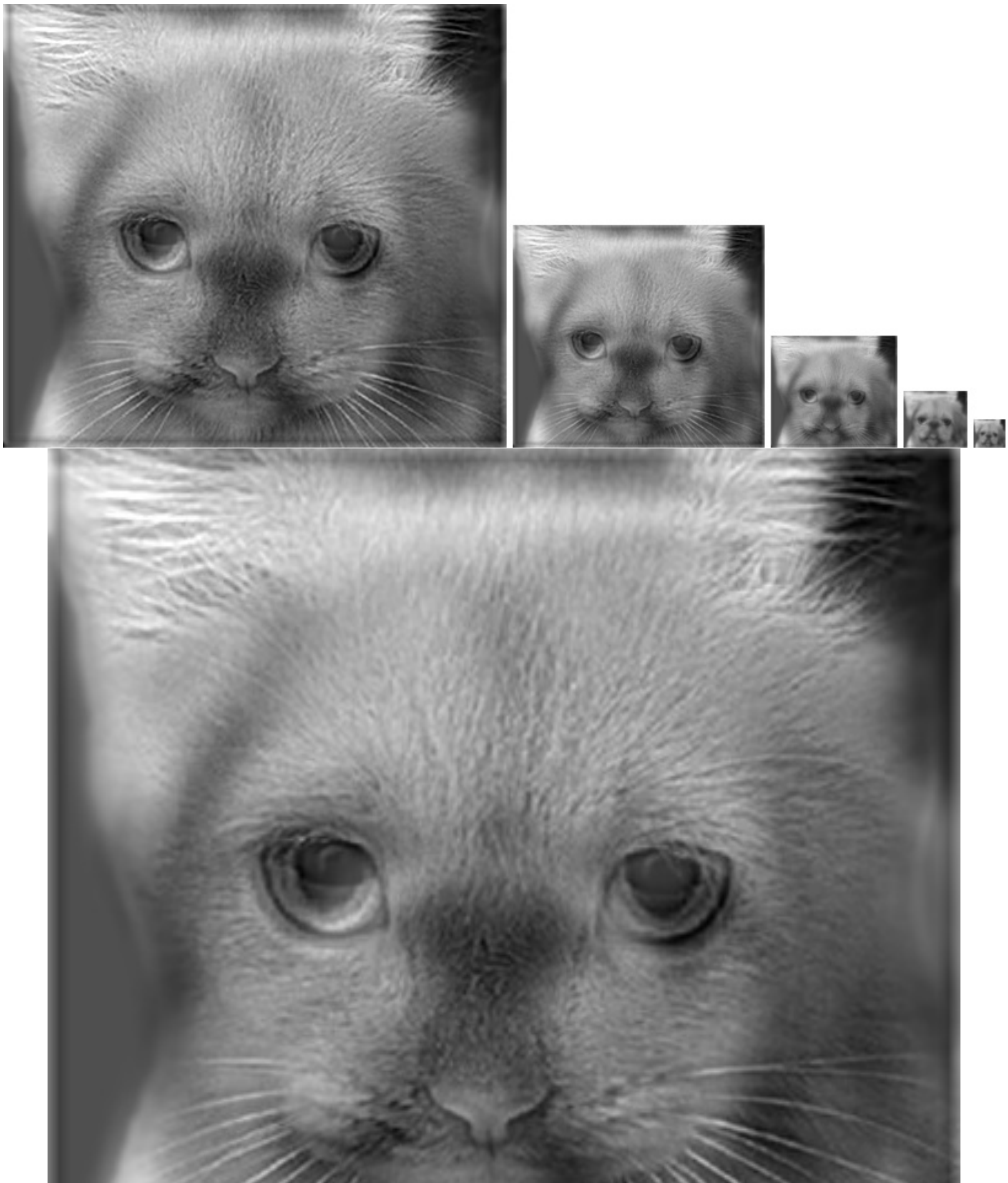


Image 1
HPF
1
0.999
0.998
0.997
50
0
0
50
Image 2 with HPF
FFT of Image 2
50
100
150
200
250
300
350
100
200
300
400
FFT of HPF
10
20
30
40
10
20
30
40
FFT of Image 2 Filtered
50
100
150
200
250
300
350
100
200
300
400
Image 1 with Low Pass Filter
Image 2 with High Pass Filter
Hybrid Image
FFT of Image 1 with LPF
50
100
150
200
250
300
350
100
200
300
400
FFT of Image 2 with HPF
50
100
150
200
250
300
350
100
200
300
400
FFT of Hybrid Image
50
100
150
200
250
300
350
100
200
300
400

Final Hybrid

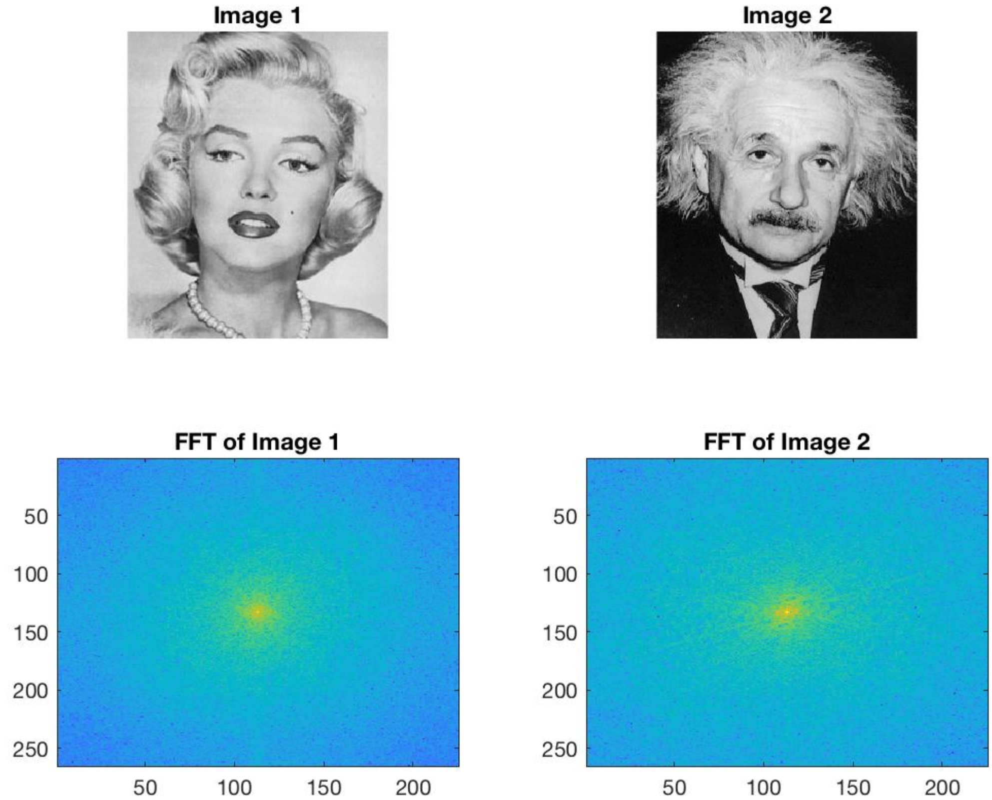


Hybrid Image Scale



Result 2

Original Images and FFTs



Filtered Images and FFTs

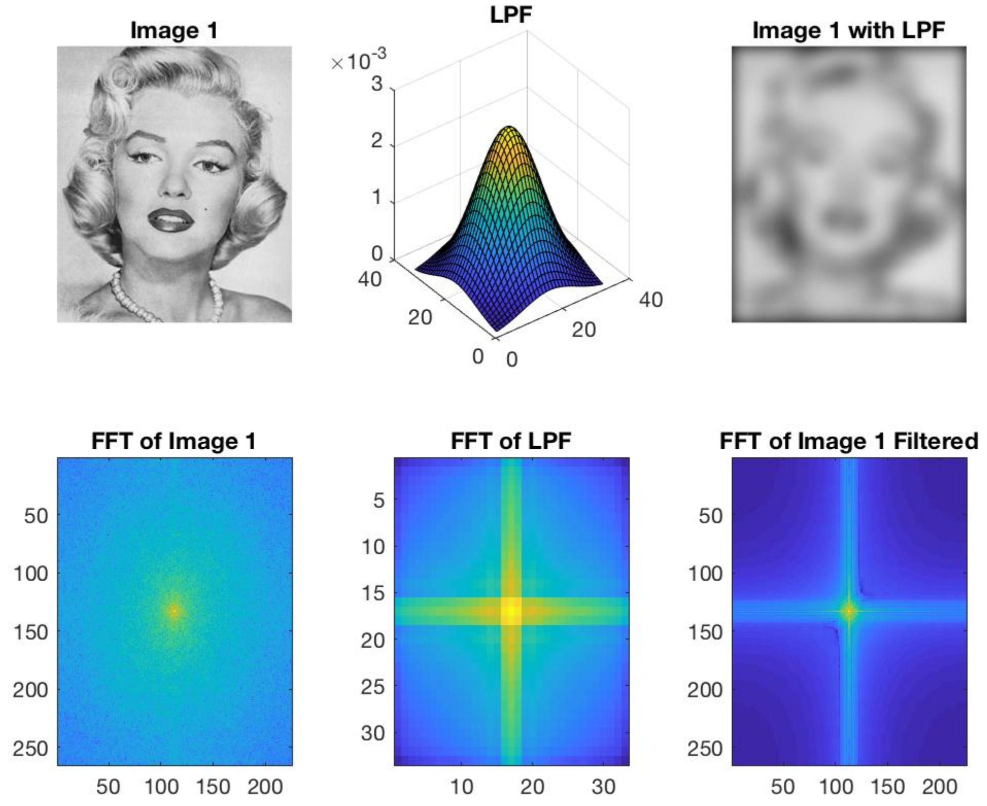


Image 1
LPF
Image 1 with LPF
FFT of Image 1
FFT of LPF
FFT of Image 1 Filtered

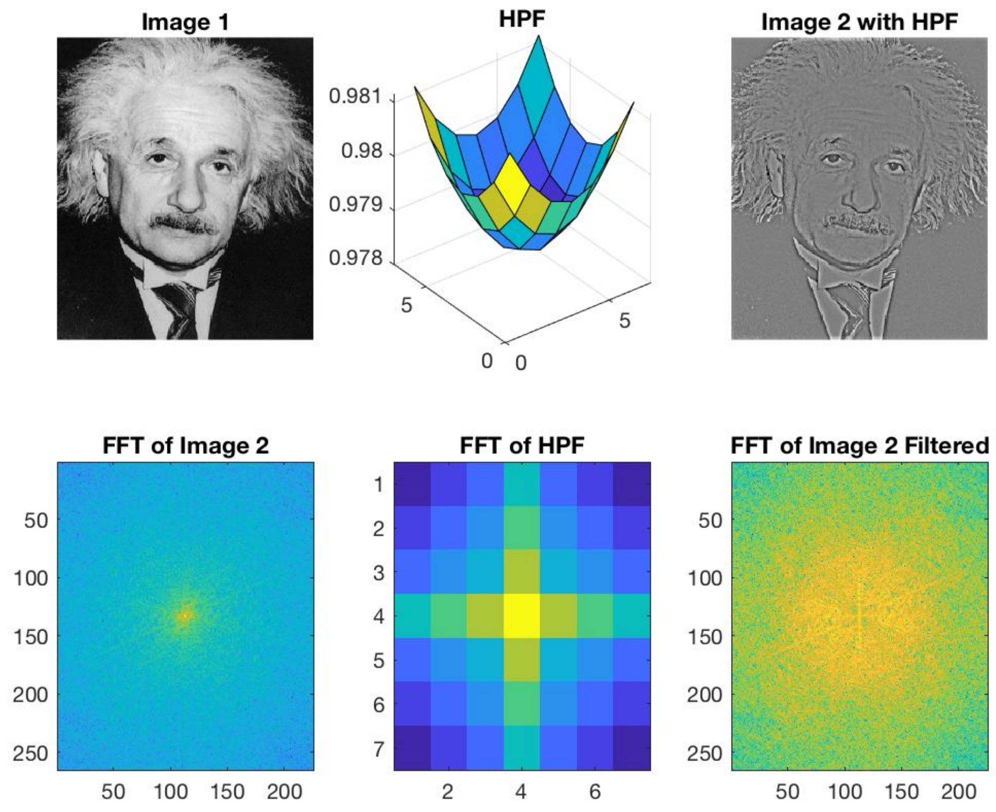


Image 1
HPF
Image 2 with HPF
FFT of Image 2
FFT of HPF
FFT of Image 2 Filtered

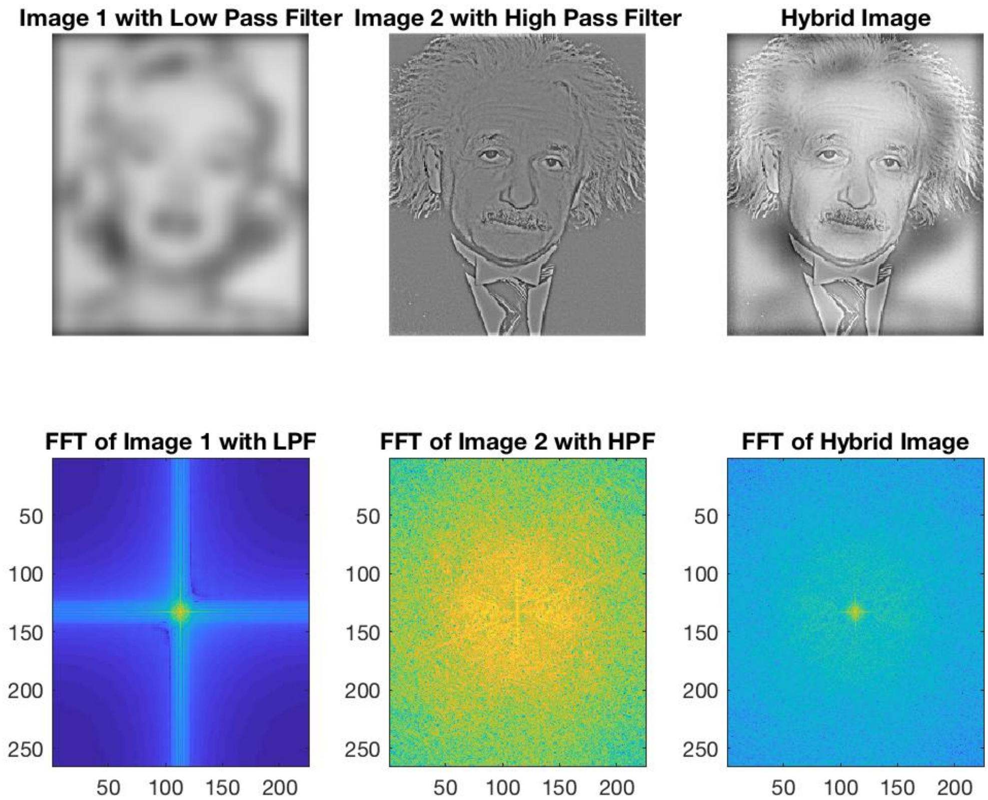
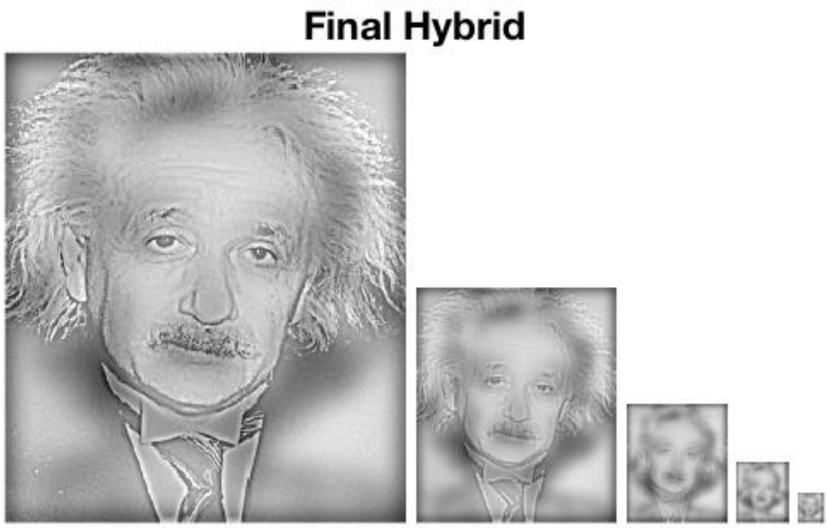
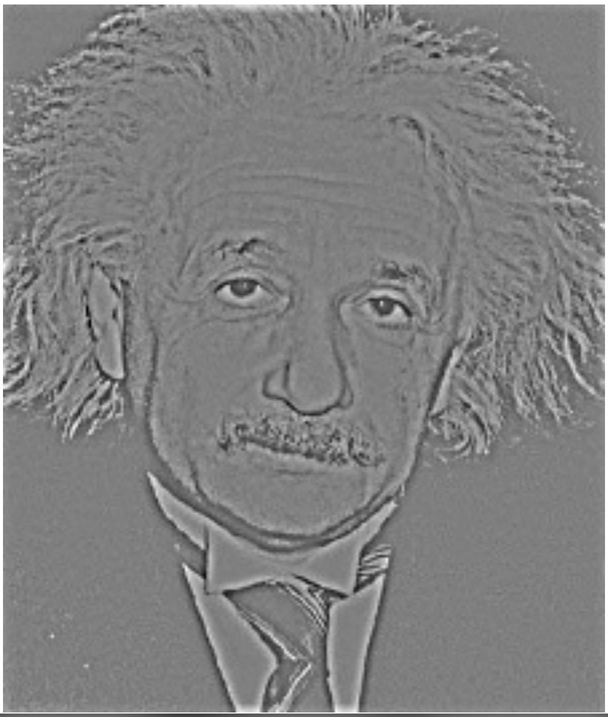
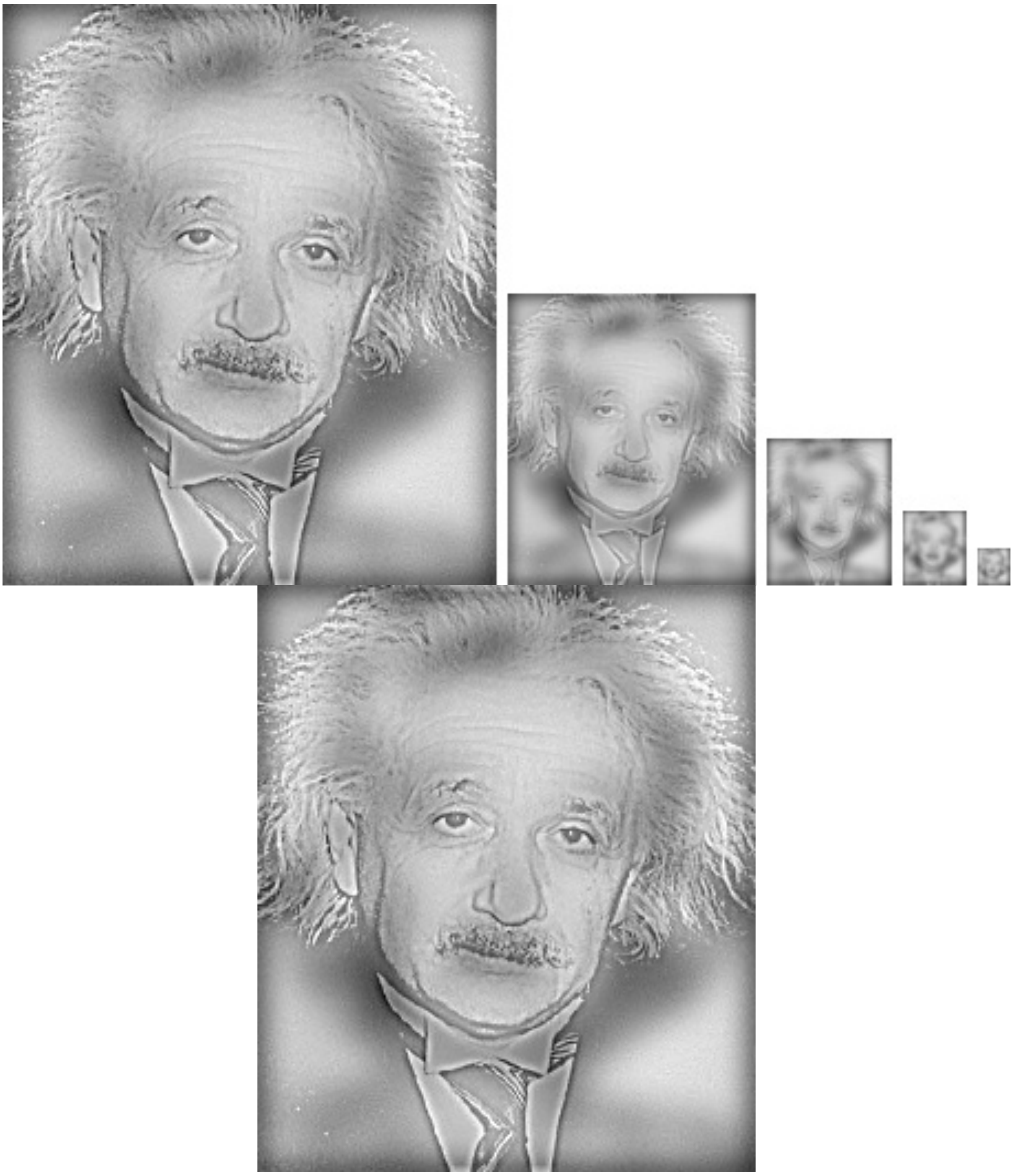


Image 1 with Low Pass Filter
Image 2 with High Pass Filter
Hybrid Image
FFT of Image 1 with LPF
FFT of Image 2 with HPF
FFT of Hybrid Image
50
100
150
200
250
50
100
150
200

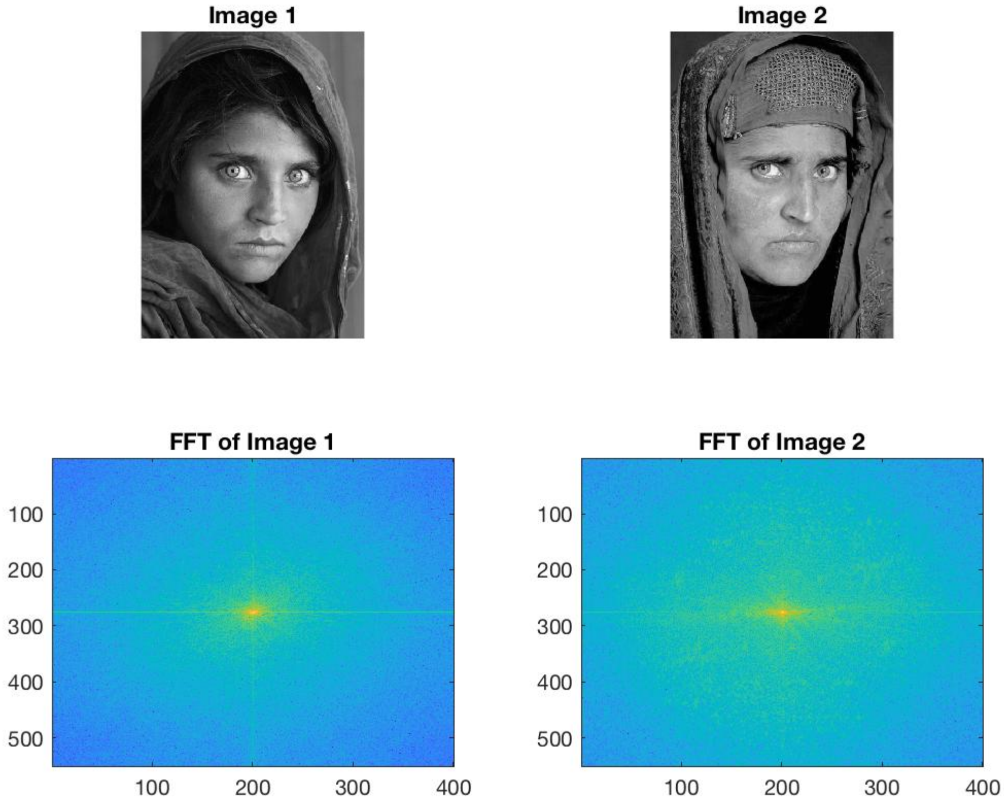

Final Hybrid





Result 3

Original Images and FFTs



Filtered Images and FFts

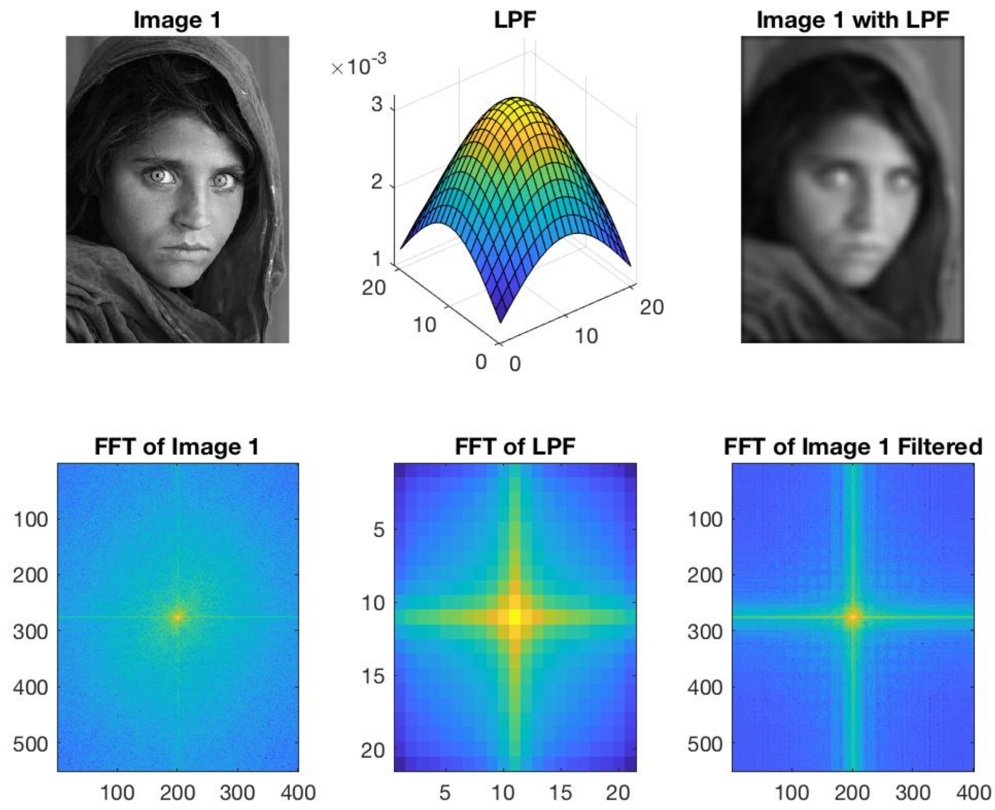


Image 1
LPF
Image 1 with LPF
FFT of Image 1
FFT of LPF
FFT of Image 1 Filtered

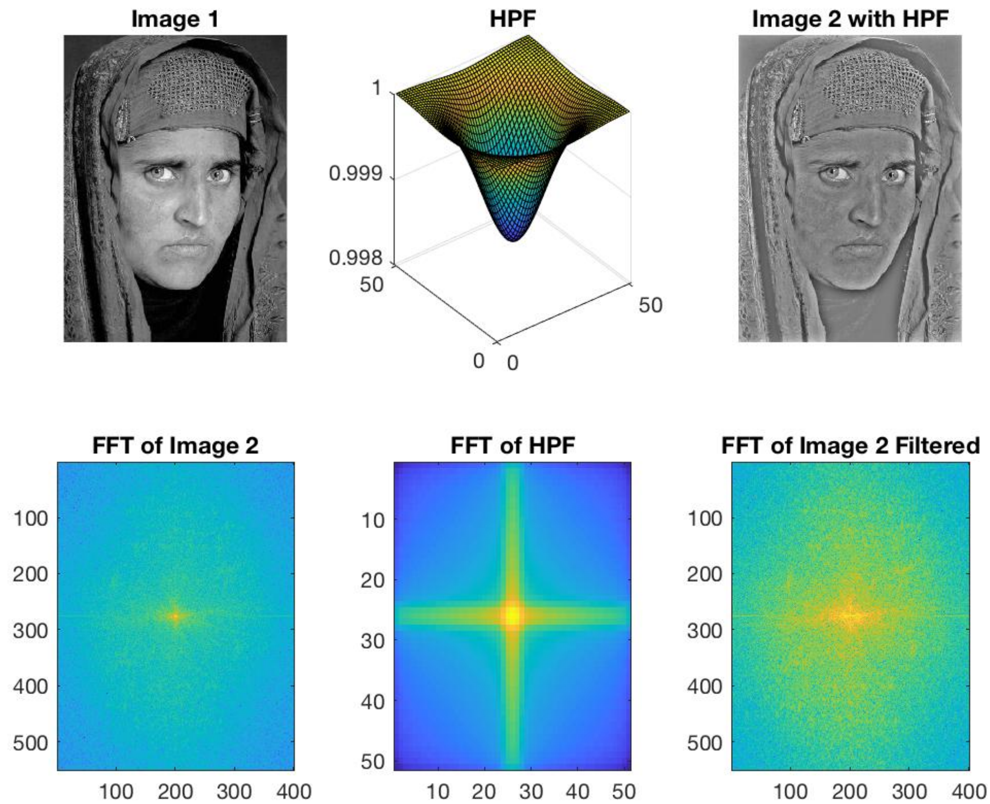
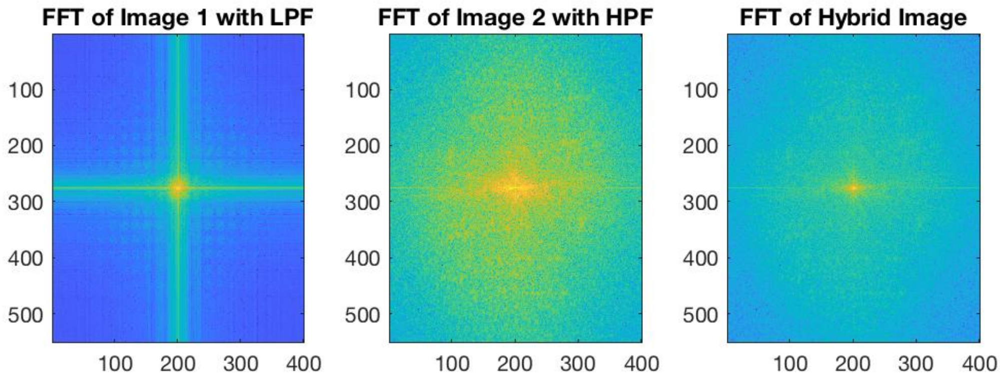
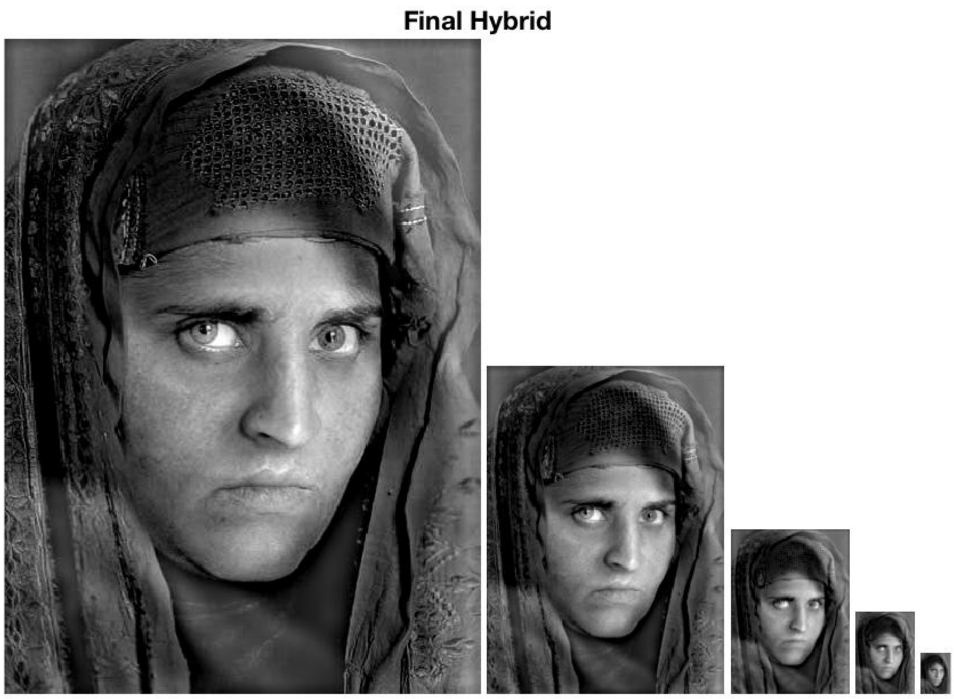


Image 1
HPF
Image 2 with HPF
FFT of Image 2
FFT of HPF
FFT of Image 2 Filtered

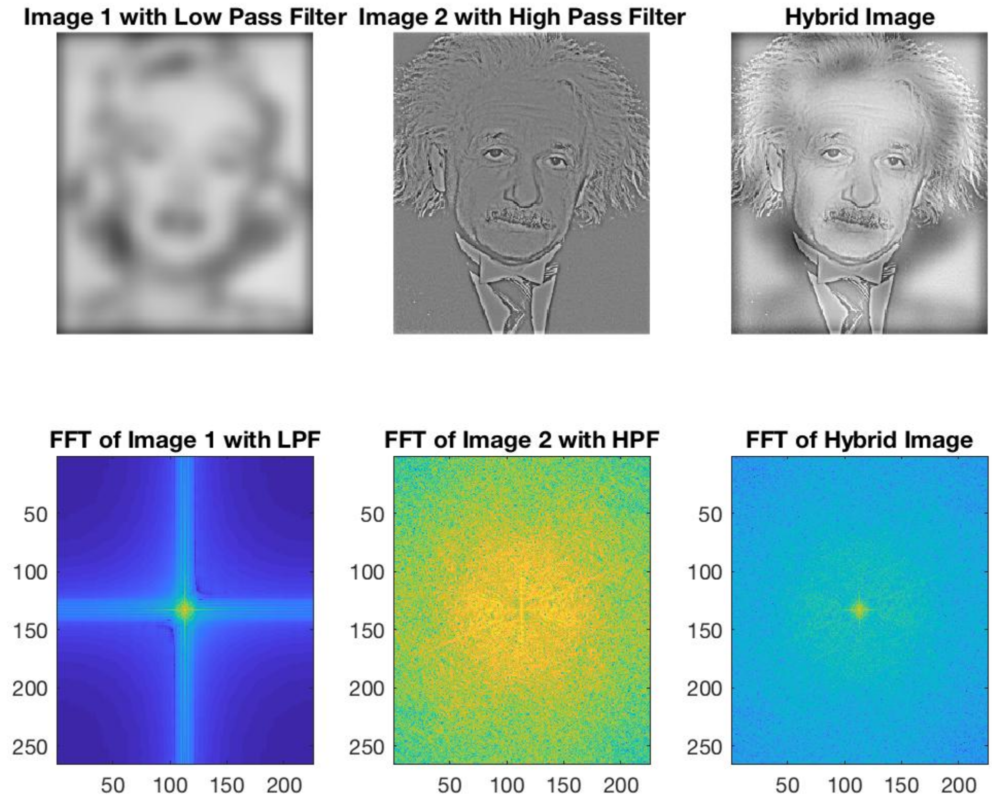

Image 1 with Low Pass Filter
Image 2 with High Pass Filter
Hybrid Image


FFT of Image 1 with LPF
FFT of Image 2 with HPF
FFT of Hybrid Image
100
200
300
400
500
100
200
300
400


Final Hybrid

Working of Hybrid Images:

Hybrid images can be obtained by combining mutually exclusive frequency bands of two different images. For example, summing low frequency information obtained by passing an image through a low pass filter, and high frequency information from another image obtained using a high pass filter gives us a hybrid image. This is illustrated in the figure below:

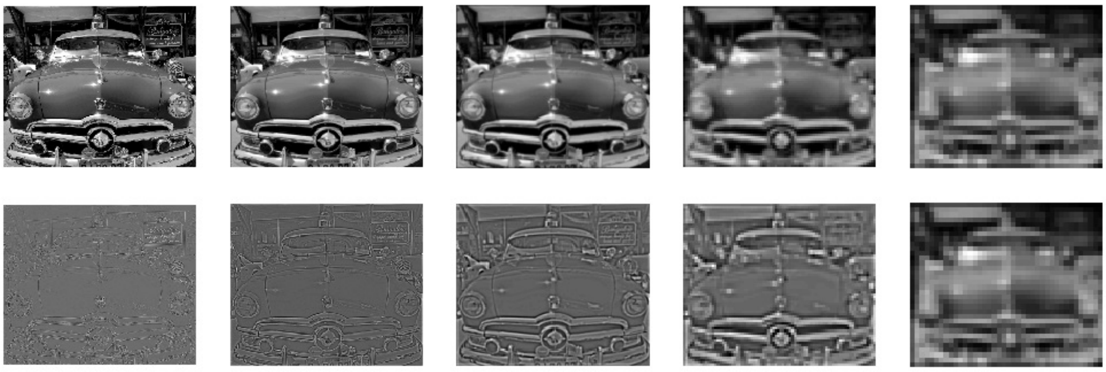


In this example, image 1 is passed through a low pass filter and image 2 through a high pass filter. Then summation is performed at the pixel level to obtain a hybrid image containing low frequencies from the first image and high frequencies from the second. The Fourier transform of the images and the hybrid image is also shown below for reference.

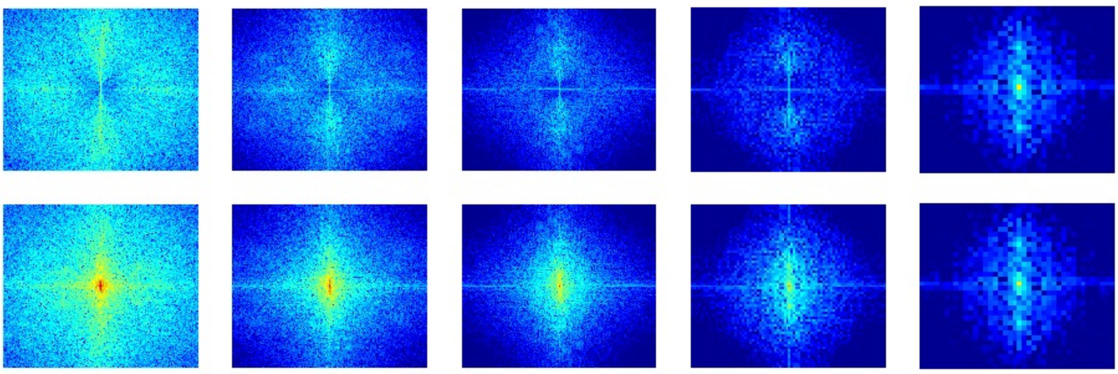
PART 2

QUESTION:

- (a) Display a Gaussian and Laplacian pyramid of level 5 (using your code). It should be formatted similar to the figure below. You may find tight_subplot.m, included in hw1.zip, to be helpful.



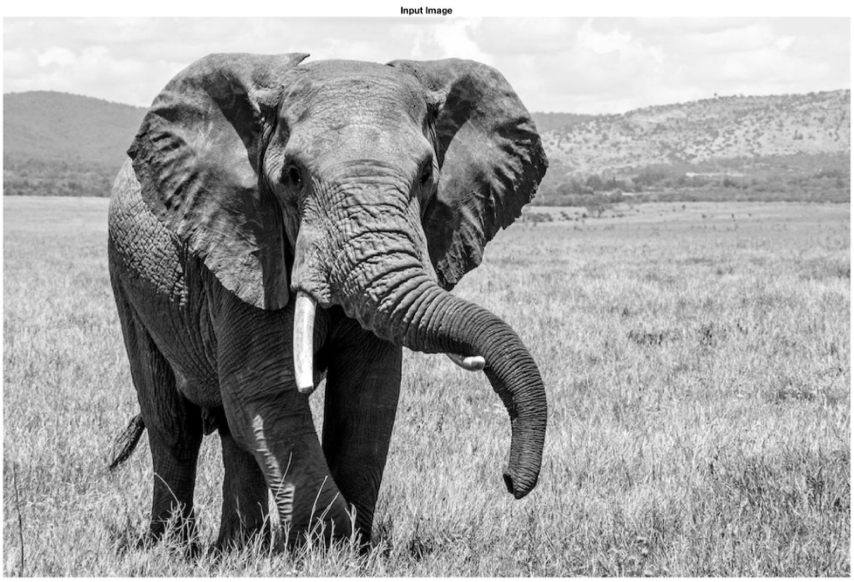
- (b) Display the FFT amplitudes of your Gaussian/Laplacian pyramids. Appropriate display ranges (using `imagesc`) should be chosen so that the changes in frequency in different levels of the pyramid are clearly visible. Explain what the Laplacian and Gaussian pyramids are doing in terms of frequency.



Answer:

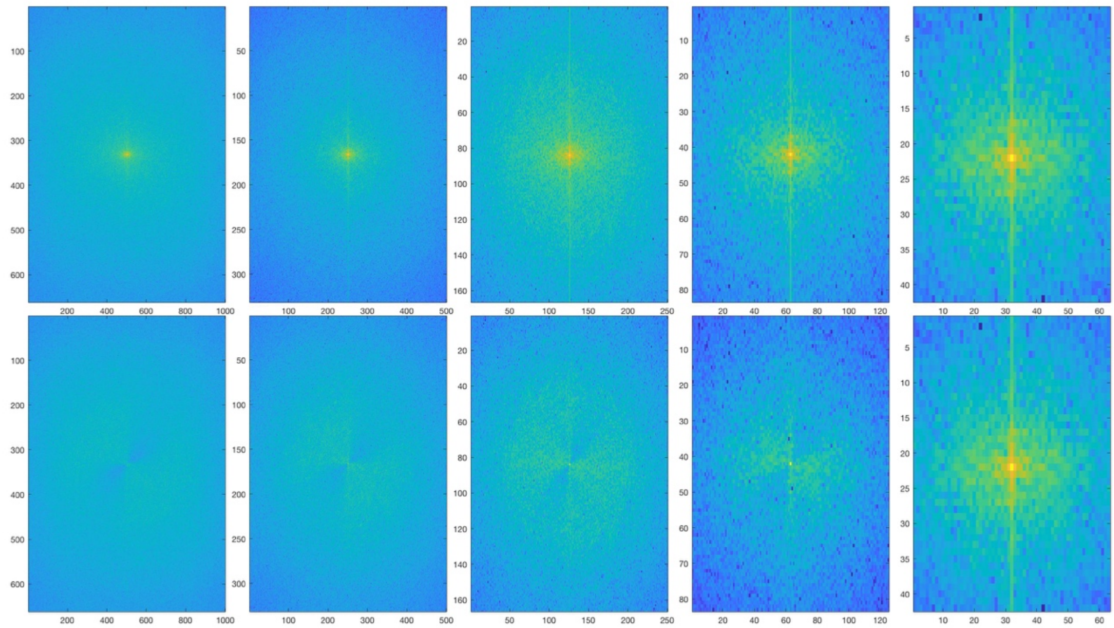
Gaussian and Laplacian Pyramid:

Input Image:



Gaussian and Laplacian Pyramids:

FFTs of Gaussian and Laplacian Pyramids:



What Gaussian and Laplacian pyramids do in terms of frequency:

Gaussian and Laplacian pyramids split the image frequencies into distinct orthogonal frequency bands, similar to applying band pass filters to the input image and forming a new image for each frequency band in the spectrum. So, each image contains frequencies in a range with almost no overlapping of frequencies.

PART 3

QUESTION:

- Description of any design choices and parameters
- The bank of filters used for part (b) (`imagesc` or `mat2gray` may help with visualization)
- Qualitative results: choose two example images; show input images and outputs of each edge detector
- Quantitative results: precision-recall plots (see “pr_full.jpg” after running evaluation) and tables showing the overall F-score (the number shown on the plot) and the average F-score (which is outputted as text after running the evaluation script).

Answer:

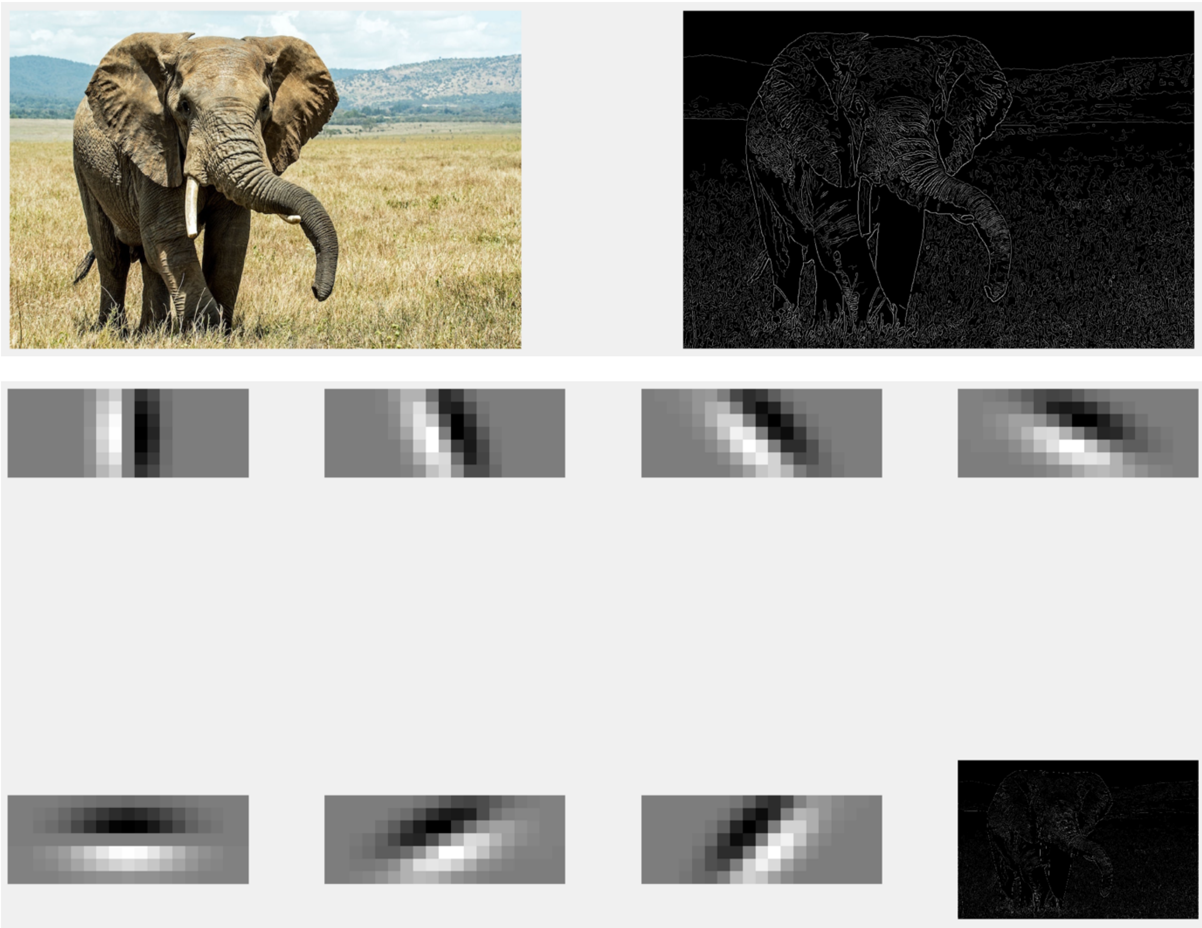
Bank of Filters used for part (b):

Example Images:

Example 1

Input Image:

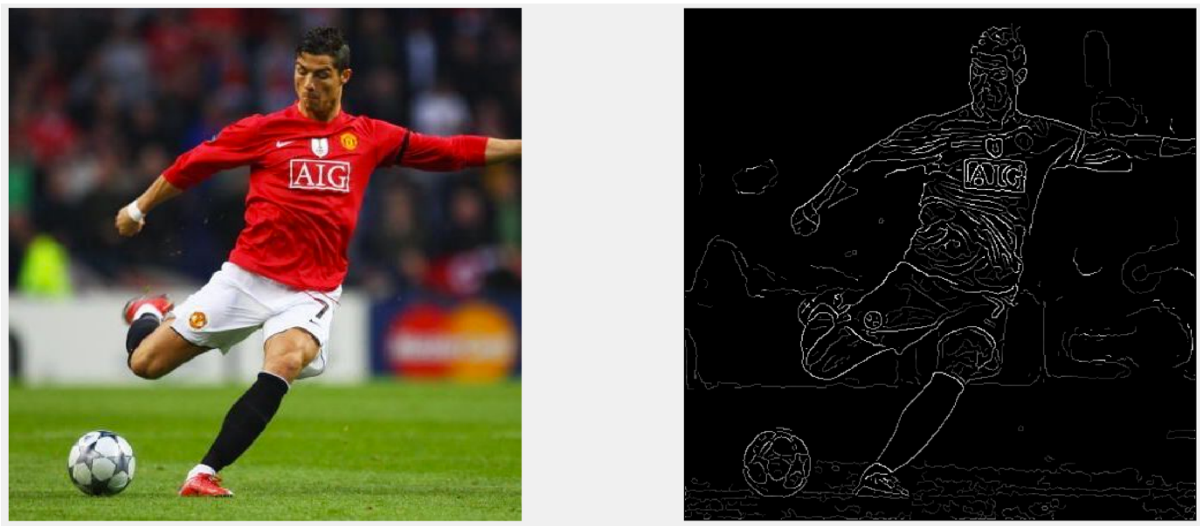
Outputs of Edge detectors:



Example 2

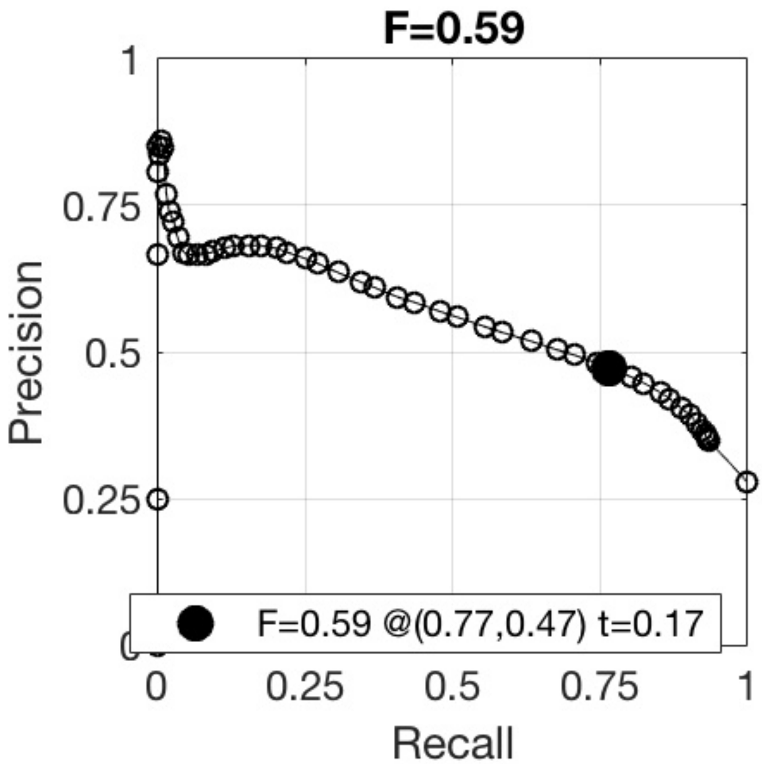
Input Image:

Outputs of Edge detectors:

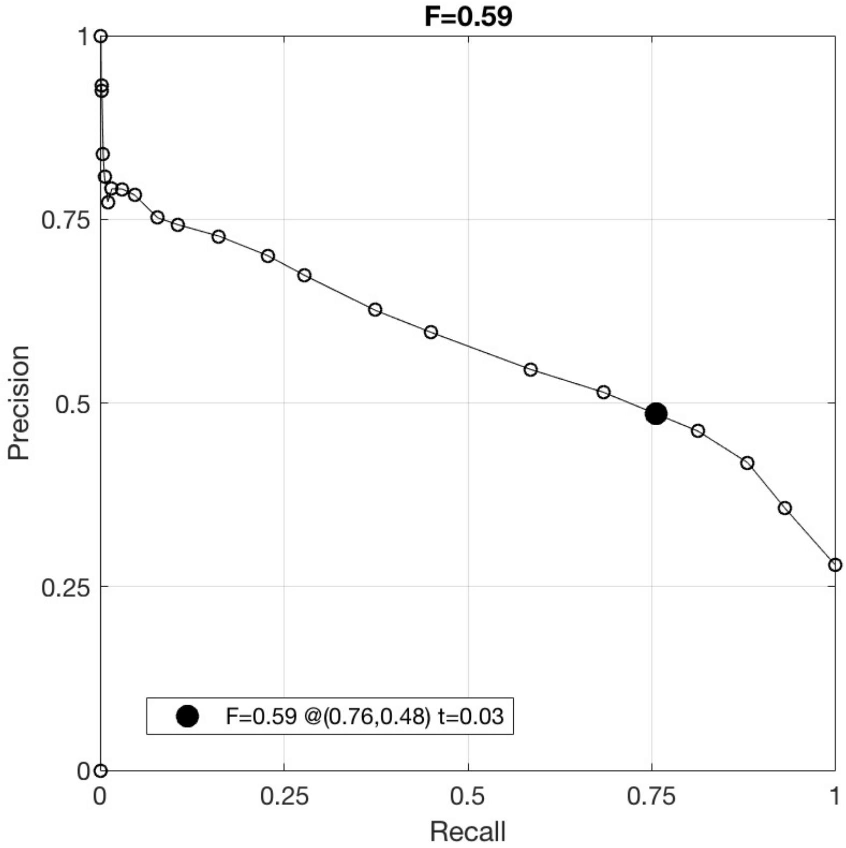


Precision Recall Plots:

Gradient:



Edge Oriented Filters:



Tables of Scores:

Gradient:

Method gradient: overall F-score = 0.585 average F-score = 0.620

Edge Oriented Filters:

Method oriented: overall F-score = 0.593 average F-score = 0.623

Graduate Credits:

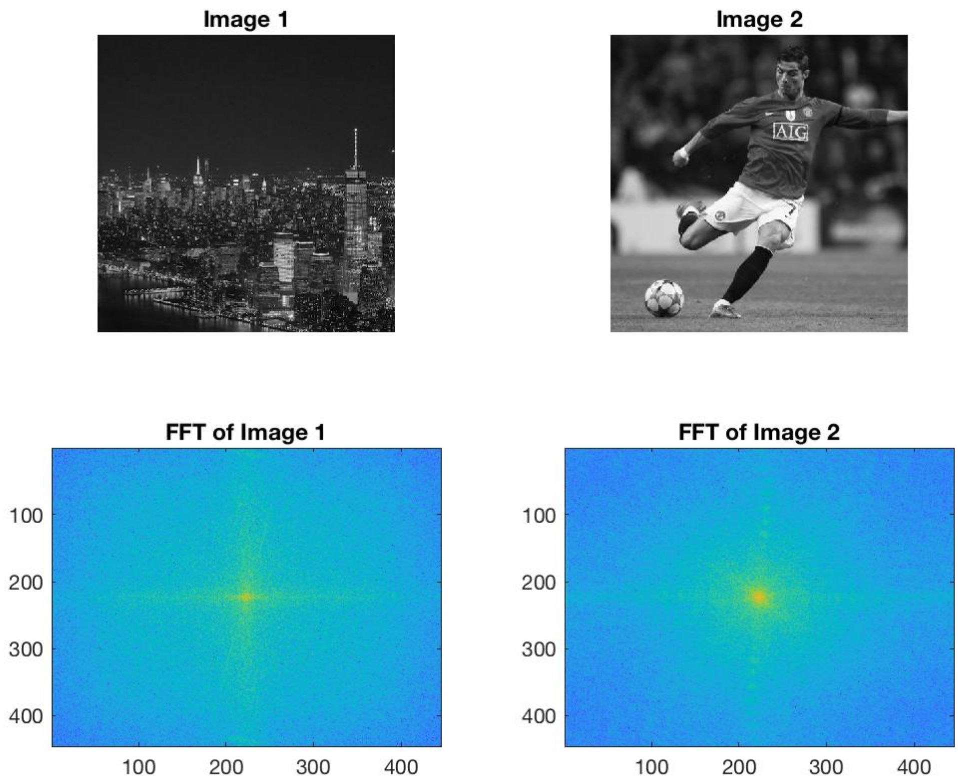
PART 1

Answer 1:

2 Examples:

Example 1: Failure Example

Original Images with FFTs



Filtered Images with FFTs

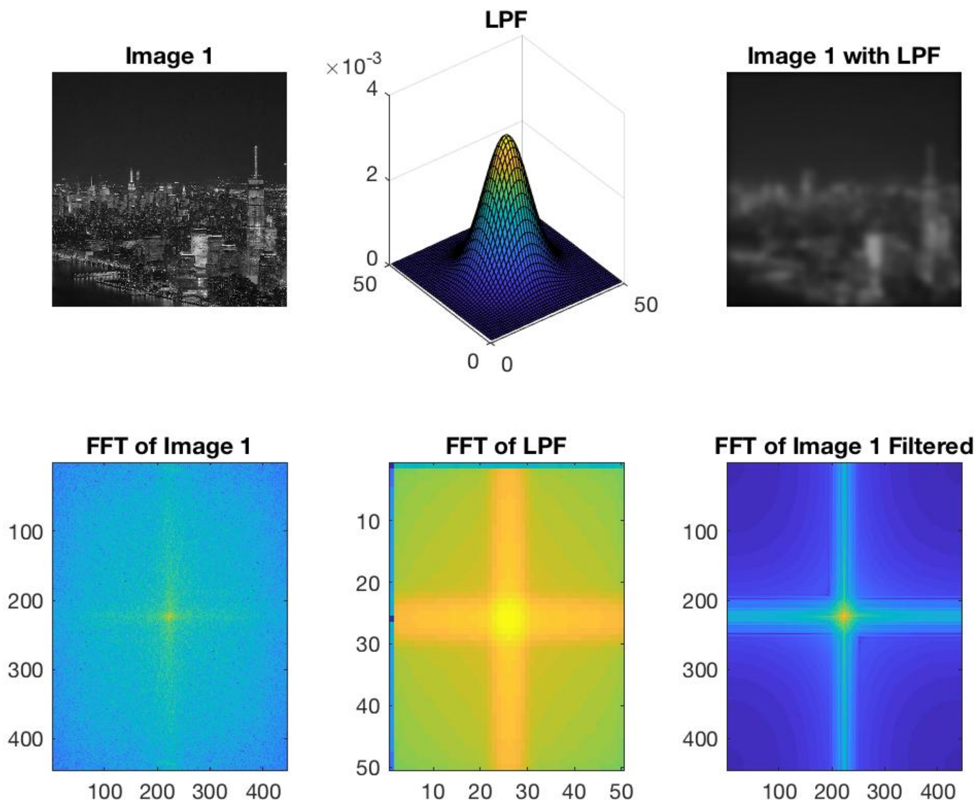


Image 1
LPF
×10-3
4
2
0
50
50
0 0
Image 1 with LPF
FFT of Image 1
100
200
300
400
100 200 300 400
FFT of LPF
10
20
30
40
50
10 20 30 40 50
FFT of Image 1 Filtered
100
200
300
400
100 200 300 400

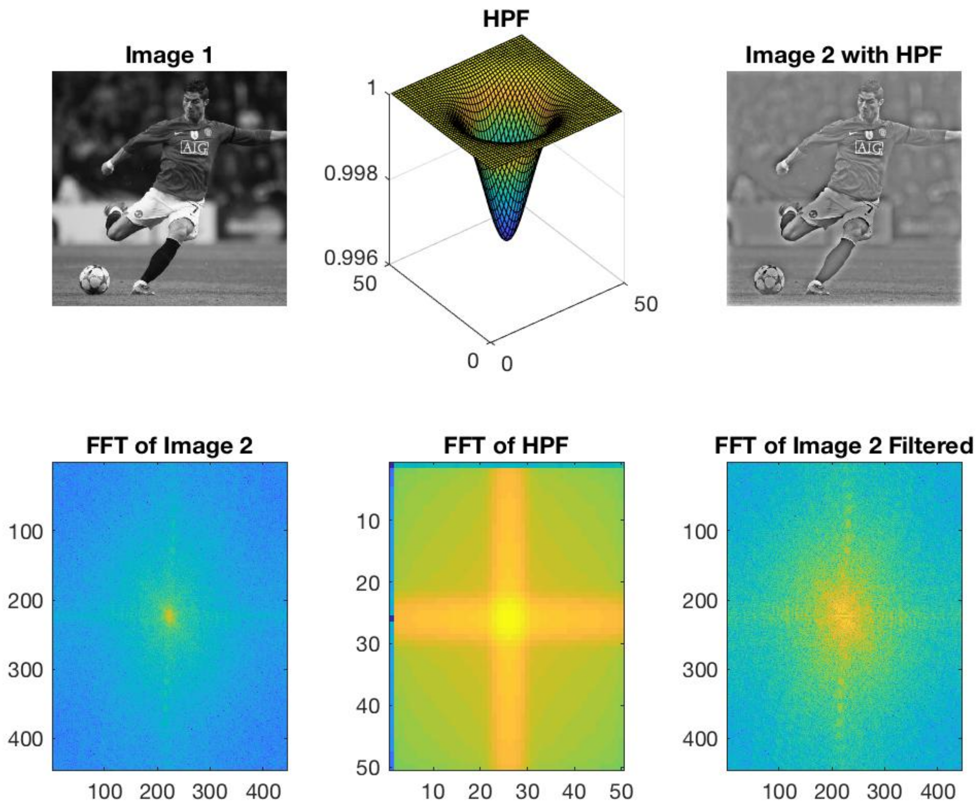


Image 1
HPF
1
0.998
0.996
50
50
0 0
Image 2 with HPF
FFT of Image 2
100
200
300
400
100 200 300 400
FFT of HPF
10
20
30
40
50
10 20 30 40 50
FFT of Image 2 Filtered
100
200
300
400
100 200 300 400

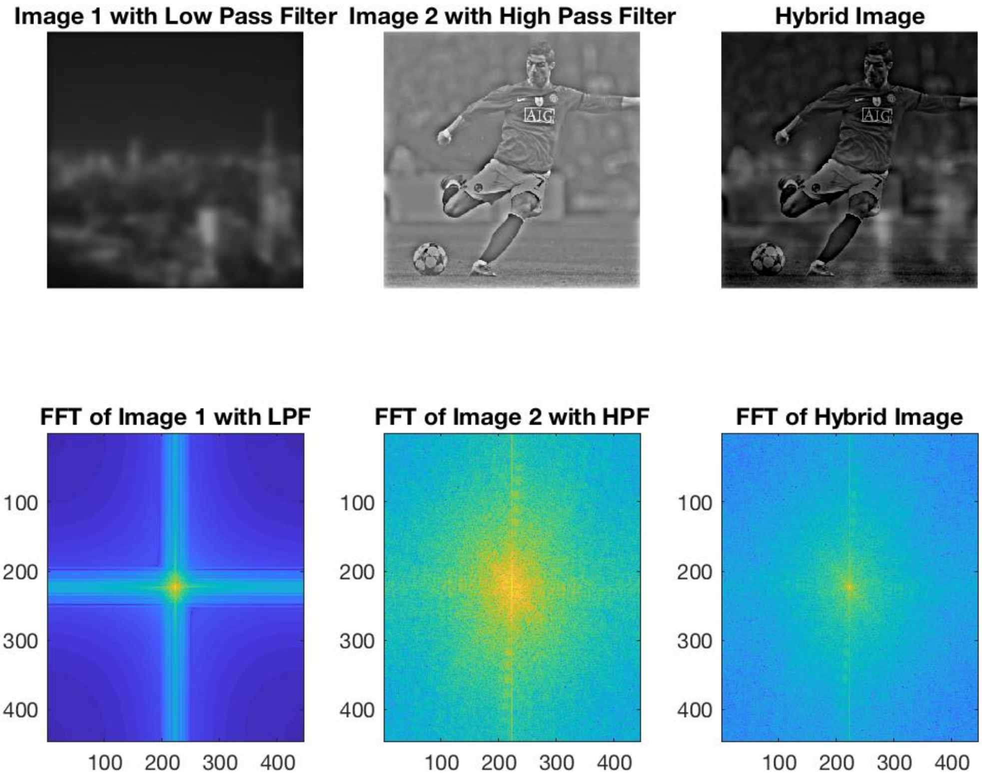
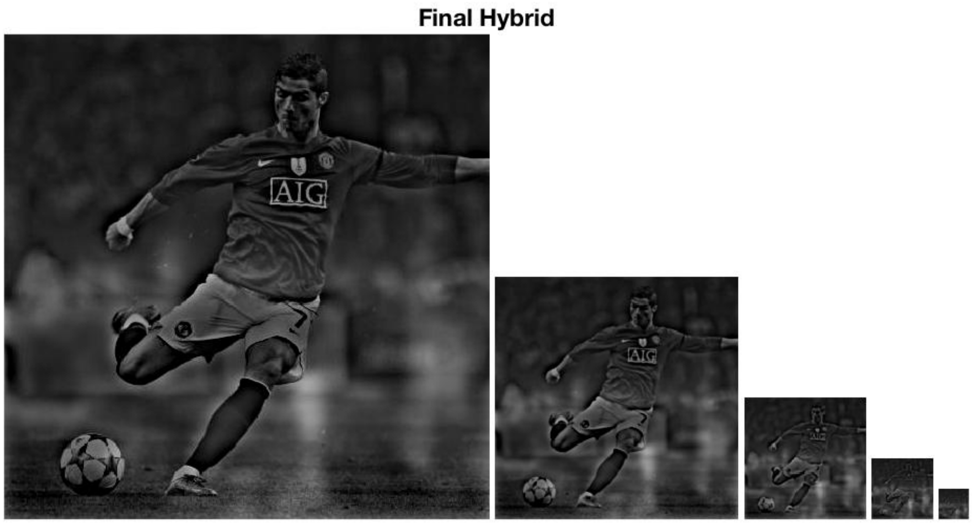


Image 1 with Low Pass Filter
Image 2 with High Pass Filter
Hybrid Image
FFT of Image 1 with LPF
FFT of Image 2 with HPF
FFT of Hybrid Image
100
200
300
400
100 200 300 400


Final Hybrid

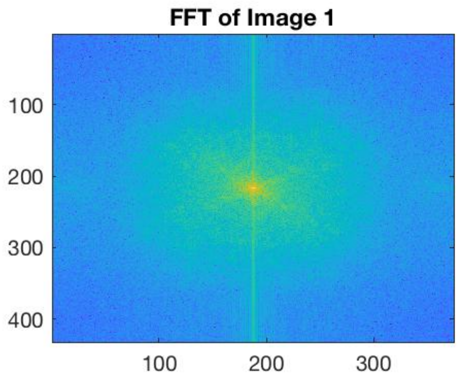

AIG

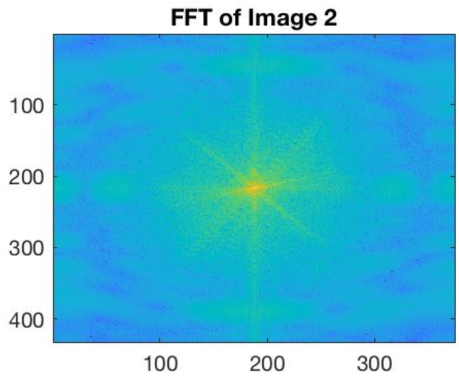
Hybrid Image Scale


AIG

Example 2:

Original Images with FFTs





Filtered Images with FFTs

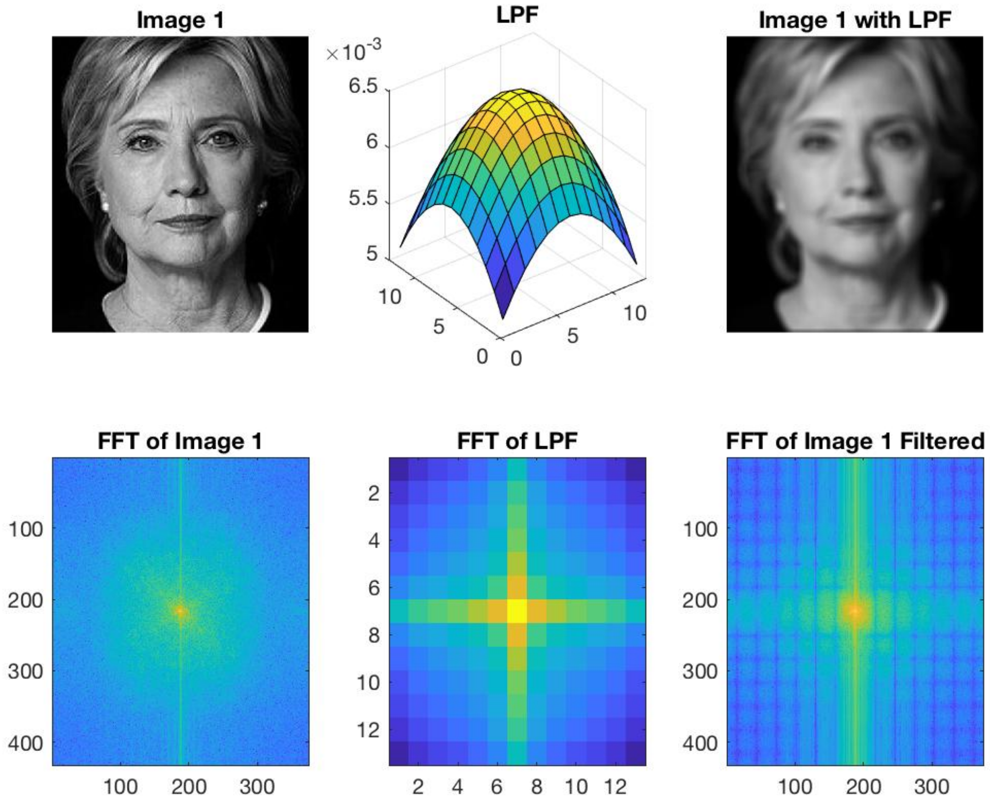


Image 1
LPF
Image 1 with LPF
FFT of Image 1
FFT of LPF
FFT of Image 1 Filtered

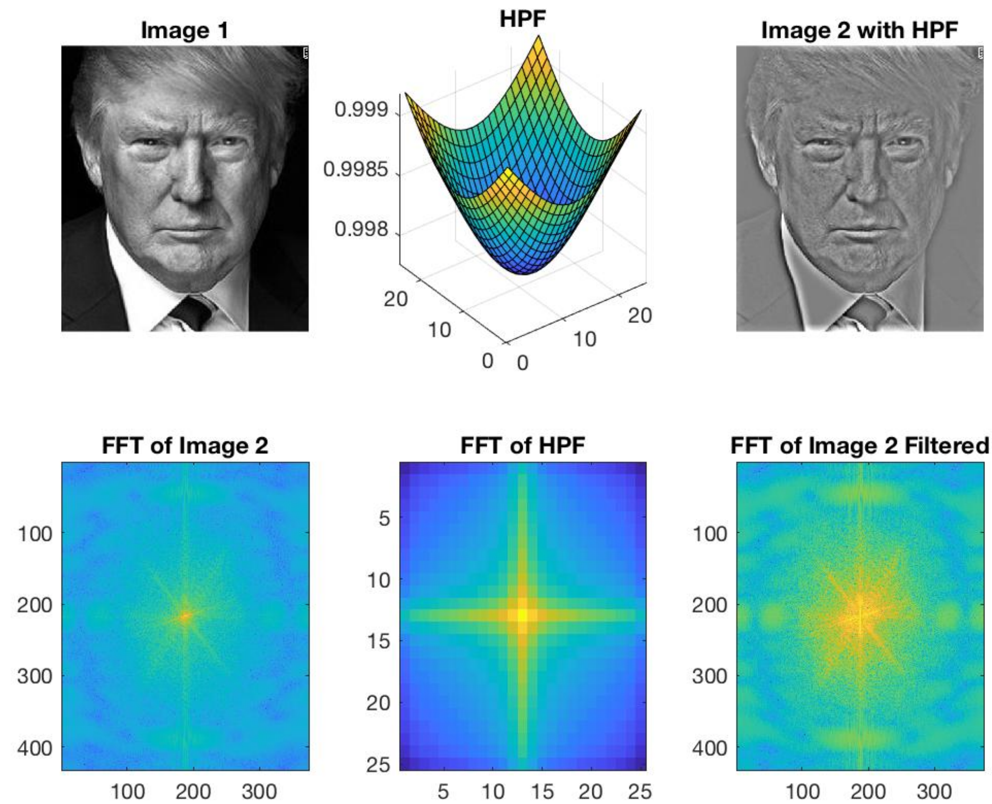


Image 1
HPF
Image 2 with HPF
FFT of Image 2
FFT of HPF
FFT of Image 2 Filtered

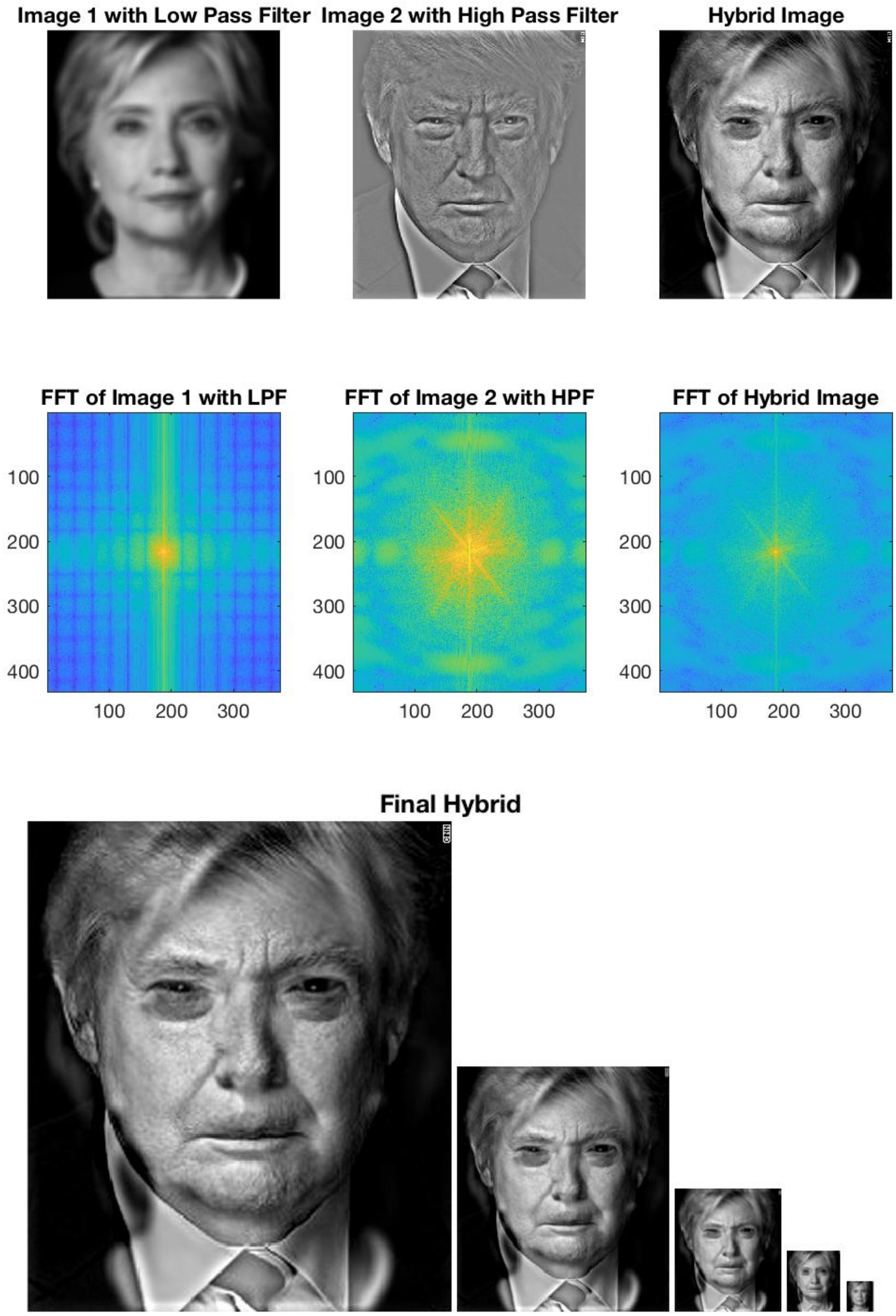
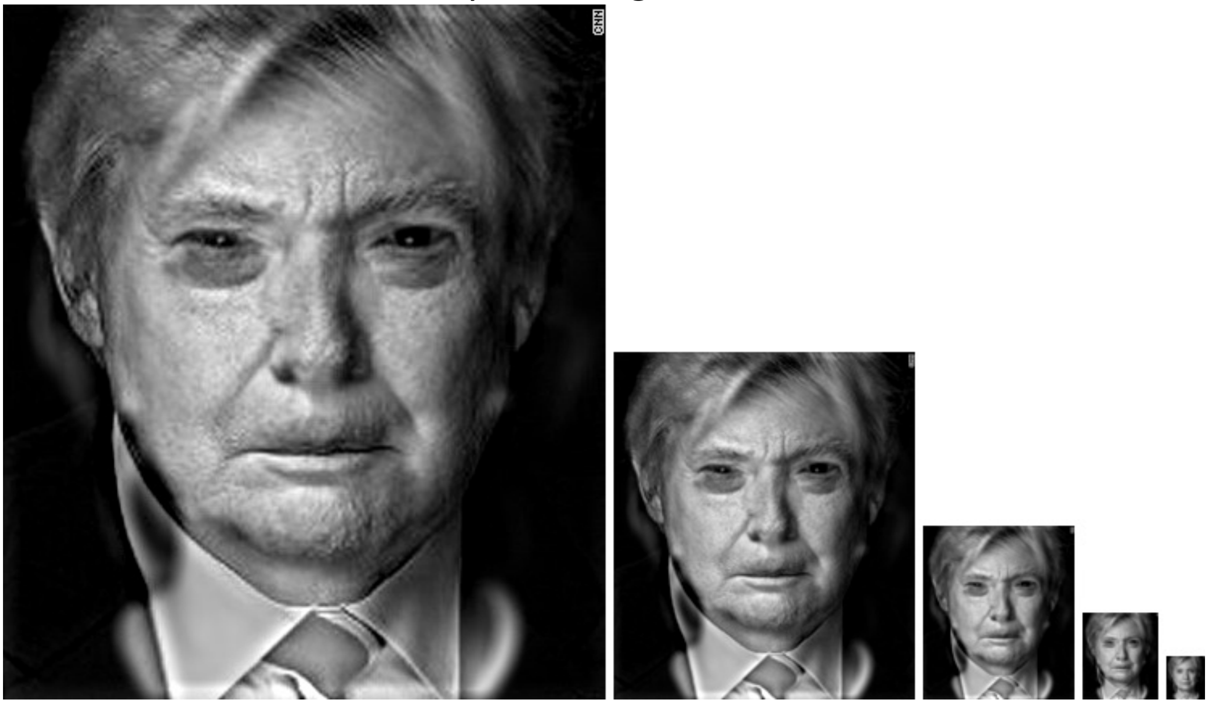


Image 1 with Low Pass Filter
Image 2 with High Pass Filter
Hybrid Image
FFT of Image 1 with LPF
FFT of Image 2 with HPF
FFT of Hybrid Image
100
200
300
400
100
200
300
Final Hybrid


CNN

Hybrid Image Scale



Good results of hybrid images can be obtained by using the following techniques:

1. Choosing cut off frequencies and tune coefficients based on the spectrum of the input images. This also involves some trial and error iterations.
2. Using image with high frequency content as the high pass filtered image and image with low frequency content, passed through a low pass filter.
3. Using related images to get a good illusion of an hybrid image.

4. Aligning features in the image with precision to get a better output.

Bad results arise due to the following reasons:

1. Frequency overlapping.
2. Non aligned images.
3. Images with very low high frequency content pass through a high pass filter and similarly for images with low frequency content.
4. Using non related images or with different subjects.

Answer 2:

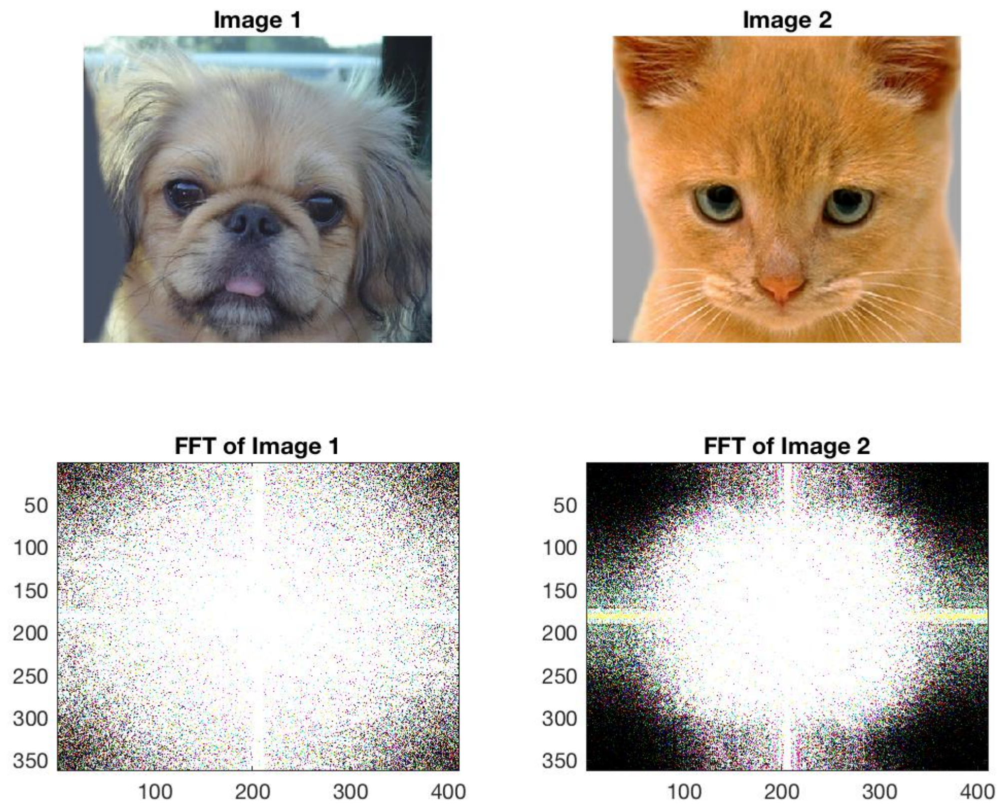


Image 1
Image 2
FFT of Image 1
FFT of Image 2
50
100
150
200
250
300
350
100
200
300
400

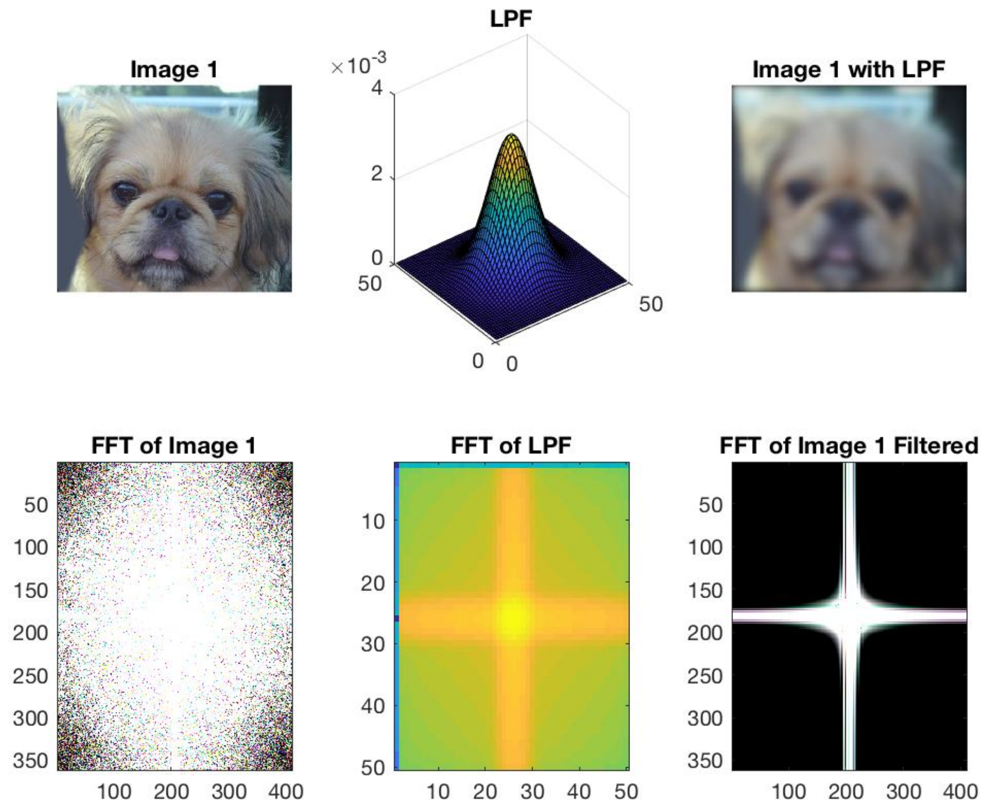


Image 1
LPF
Image 1 with LPF
$\times 10^{-3}$
4
2
0
50
0
0
50
FFT of Image 1
FFT of LPF
FFT of Image 1 Filtered

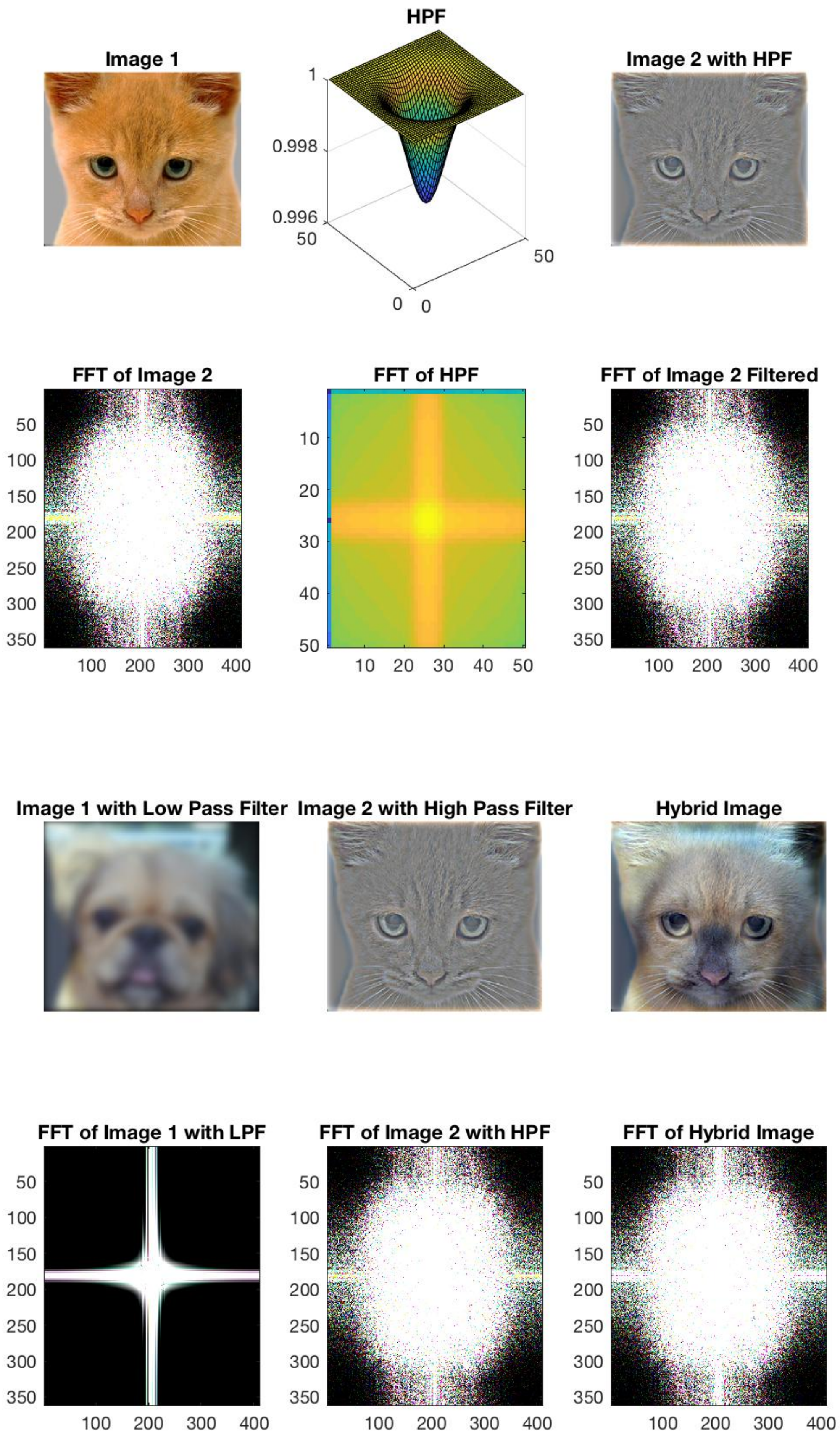
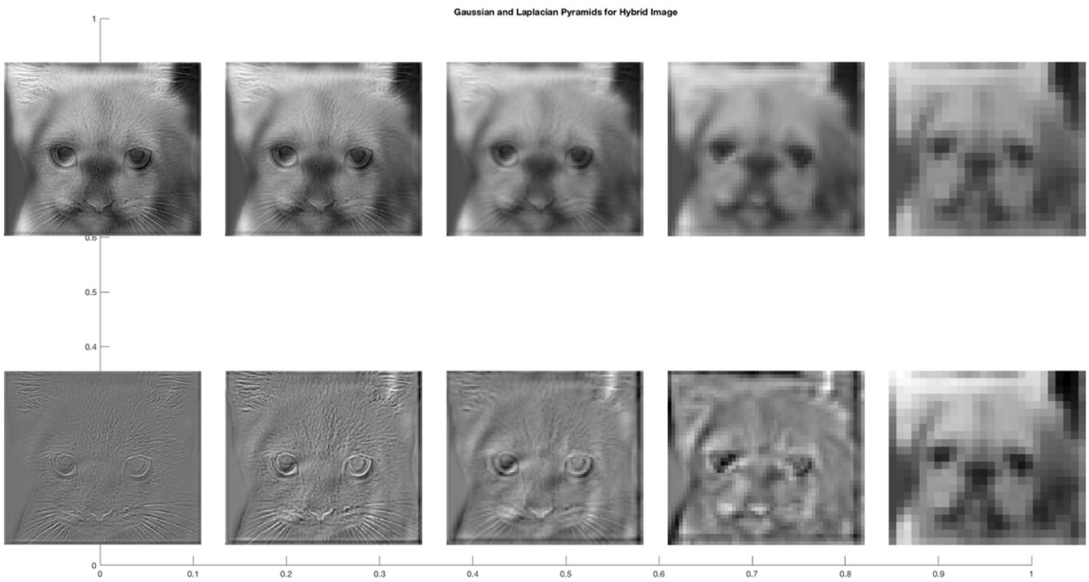


Image 1
HPF
1
0.998
0.996
50
0
0
50
Image 2 with HPF
FFT of Image 2
FFT of HPF
FFT of Image 2 Filtered
50
100
150
200
250
300
350
100 200 300 400
10
20
30
40
50
10 20 30 40 50
Image 1 with Low Pass Filter
Image 2 with High Pass Filter
Hybrid Image
FFT of Image 1 with LPF
FFT of Image 2 with HPF
FFT of Hybrid Image

Final Hybrid

When using images with colour, the low frequency information is better seen in the hybrid image, as the high frequency information becomes scarce at farther distances and colour information helps in interpreting the image, which allows for better representation of low frequency information. Therefore, it is better to use colour for displaying low frequency information in the hybrid image.

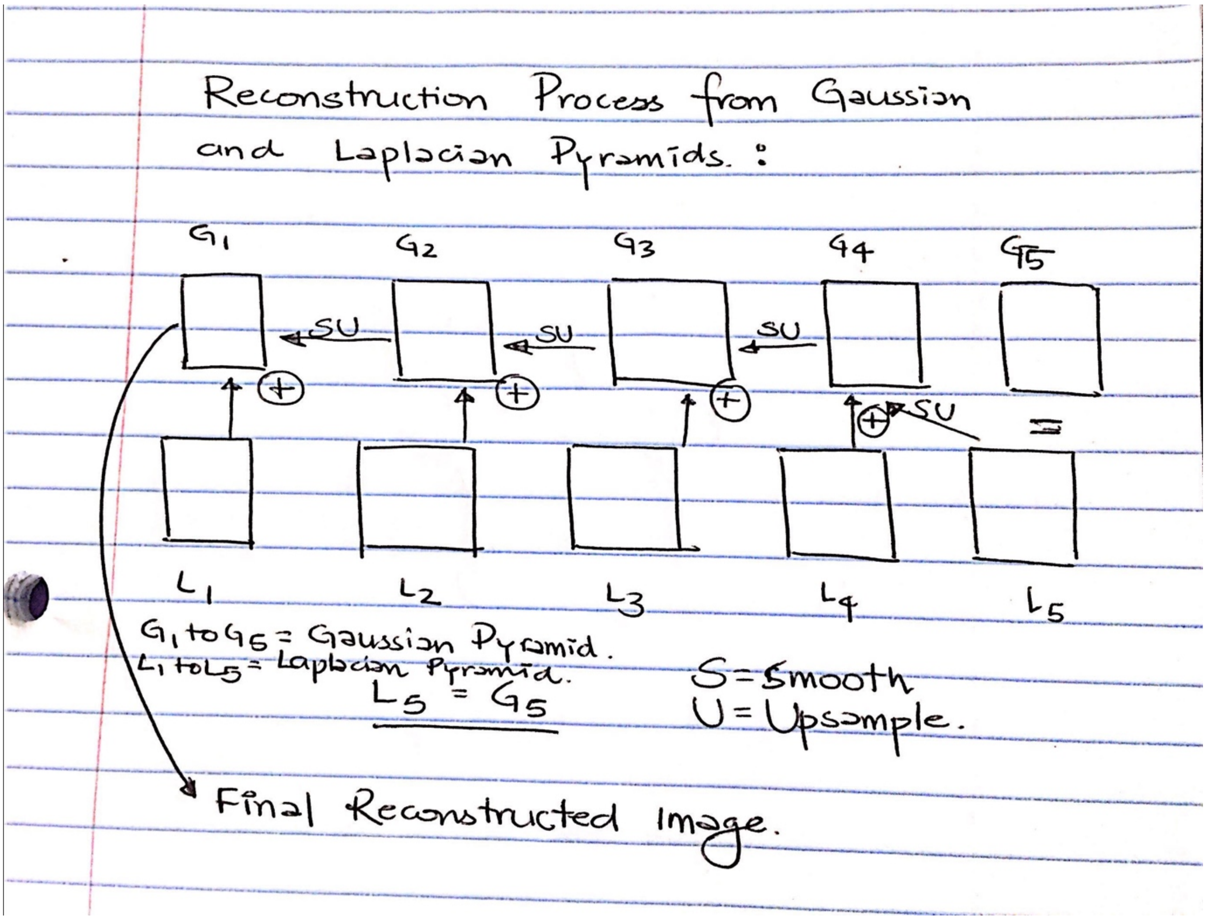
Answer 3:



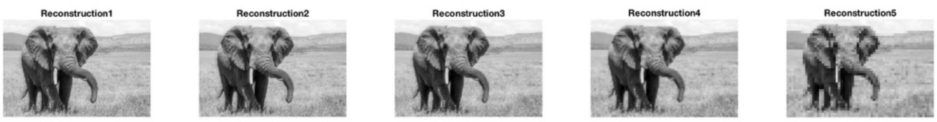
PART 2

Answer 1:

Reconstruction Process from Laplacian Pyramid:



Reconstruction Result:



Reconstruction Errors:

Reconstruction Error for this image(elephant.jpg) = 457.6319

Answer 2:

Laplacian multiplying vs 3x3 Sharpening filter: The Laplacian pyramid is essentially a set of band pass filters containing partial frequency information of the original image.
Sharpening involves passing the image through a high pass filter and adding a scaled version of the filtered output to the original image to enhance the edges and boundaries in the sharpened output.
Addition of a multiplied laplacian layer adds a band pass filter output, whereas a sharpening filter adds a high pass filter output.

PART 3

Answer 1:

Edge detection can be improved by making improvements to the Canny detector. Some improvements that can be made are:

- Improvements may be along the lines of using an adaptive filter instead of the Gaussian filter.
- Performing morphological thinning operations on the gradient output to improve the edge quality. (An Improved CANNY Edge Detection Algorithm ISBN: 978-1-4244-5285-9).
- Trying various colour spaces to get the best edge detection results(implemented in the code included with the writeup.) The F scores saw about a 15% improvement after using the HSV colour space.

Answer 2:

Changed the color space in the code to HSV color space. The F scores saw about a 15% improvement after using the HSV colour space.

Gradient Magnitude:

```
function [mag, theta] = gradientMagnitude(im, sigma)
%
% gaussianFilter = fspecial('Gaussian', sigma * 3, sigma);
% lowPassedImage = imfilter(im, gaussianFilter);
%
% xKernel = [-0.5, 0, 0.5];
% yKernel = [-0.5; 0; 0.5];
%
% xGradient = imfilter(lowPassedImge, xKernel)
% yGradient = imfilter(lowPassedImge, yKernel);
%
% mag = sqrt(xGradient.^2 + yGradient.^2);
% [maxValues, maxIndices] = max(mag, [], 3);
%
% imageSize = size(yGradient, 1) * size(yGradient, 2);
% theta = atan2(yGradient( (1:imageSize) + ( maxIndices(:)'-1 ) * imageSize
), gx( (1:imageSize) + ( maxIndices(:)'-1 ) * imageSize ) )+pi/2;
% theta = reshape(theta, [size(yGradient, 1) size(yGradient, 2)]);
%
% mag = sqrt(sum(mag.^2, 3));

% Graduate Credit: Changing to the HSV Colorspace give a higher overall F
% score.
im = rgb2hsv(im);

gauss = fspecial('gaussian', max(round(sigma*3)*2+1,3), 1);
lap = fspecial('laplacian', 0.5);
```

```
im = imfilter(im, gauss);
[gx, gy] = gradient(im);



% compute gradient magnitude for each channel (r, g, b)
mag = sqrt(gx.^2 + gy.^2);
% get orientation of gradient with largest magnitude
[mv, mi] = max(mag, [], 3);% max over third dimension
N = size(gy, 1)*size(gy, 2);
theta = atan2(gy( (1:N) + ( mi(:)'-1 ) * N ), gx( (1:N) + ( mi(:)'-1 ) * N
) )+pi/2;
theta = reshape(theta, [size(gy, 1) size(gy, 2)]);

% compute overall magnitude as L2-norm of r g b
mag = sqrt(sum(mag.^2, 3));
```

Edge Oriented Magnitude:

```
function [mag, theta] = orientedFilterMagnitude(im, sigma_long,
sigma_short, ntheta)
% [mag, theta] = orientedFilterMagnitude(im, sigma_long, sigma_short,
ntheta)

% Graduate Credit: Changing to the HSV Colorspace give a higher overall F
% score.
im = rgb2hsv(im);

figure(1), hold off;
thetas = -pi/2:pi/ntheta:(ntheta-1)*pi/ntheta;
mag = zeros(size(im, 1), size(im, 2), ntheta);
for t = 1:ntheta
  fil = orientedEdgeFilter(sigma_long, sigma_short, thetas(t));
  subplot(2, ntheta/2, t), imagesc(fil), axis image, colormap gray, axis
off;
  resp = imfilter(im, fil);
  mag(:, :, t) = sqrt(sum(resp.^2, 3));
end
[mag, mi] = max(mag, [], 3);
theta = thetas(mi);


function fil = orientedEdgeFilter(sigma_long, sigma_short, theta)
% fil = orientedEdgeFilter(sigma_long, sigma_short, theta)

sigmax = sigma_long;
sigmay = sigma_short;

[filx, fily] = meshgrid(-round(sigmax*3):round(sigmax*3), -
round(sigmay*3):round(sigmay*3));
fil = exp(-(filx.^2./sigmax.^2 + fily.^2./sigmay.^2)/2);
fil = fil ./ sum(fil(:));
fil = imfilter(fil, [0.25 ; 0 ; -0.25]);
fil = imrotate(fil, theta/pi*180, 'bilinear', 'crop');
```