Assignment 1

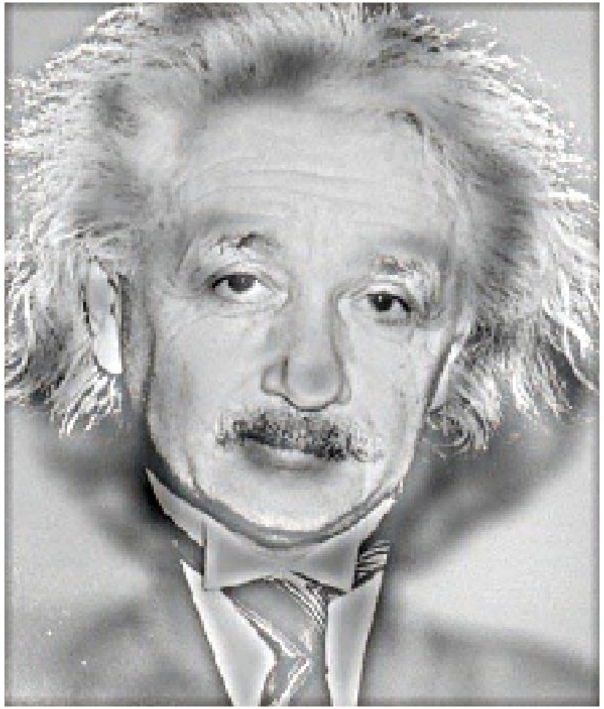
Question 1: Hybrid Images

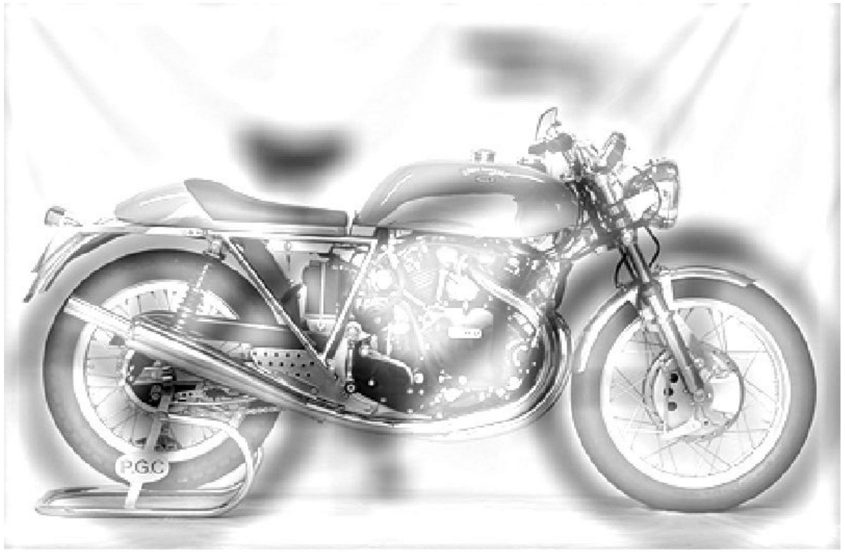
- Please provide three hybrid image results. For one of your favorite results, show

(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

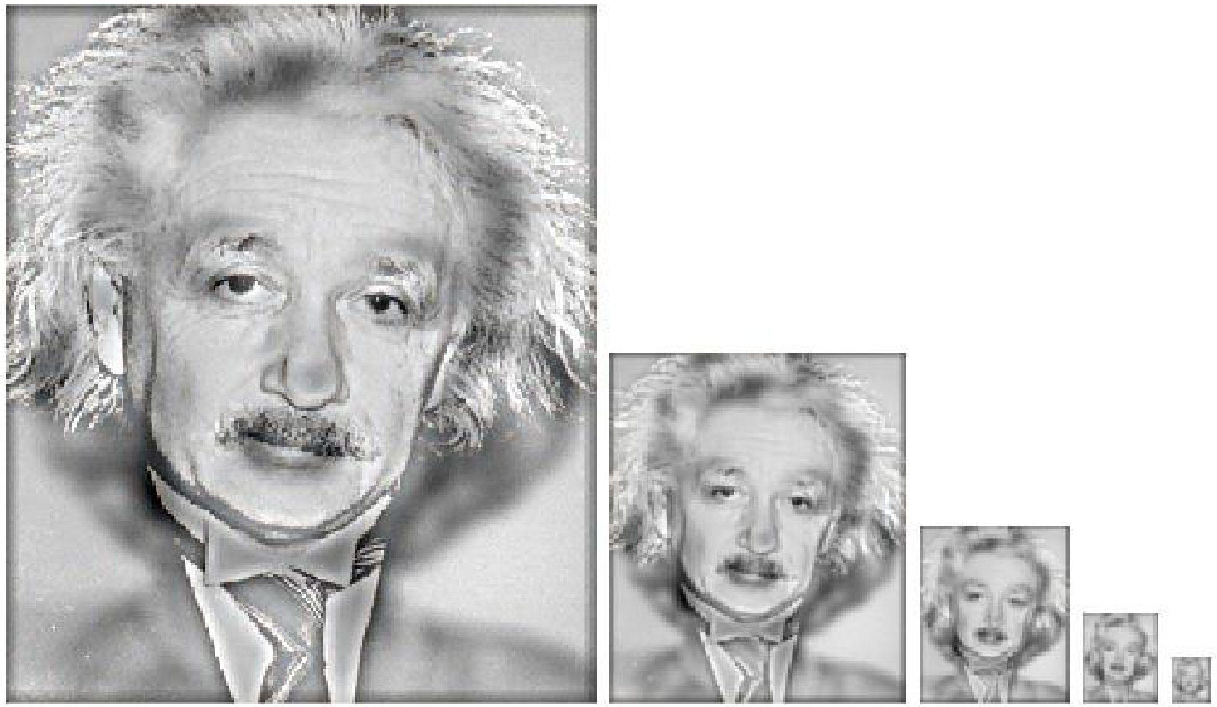
- Briefly (a few sentences) explain how it works, using your favorite results as illustrations.

Hybrid Images Results




P.G.C

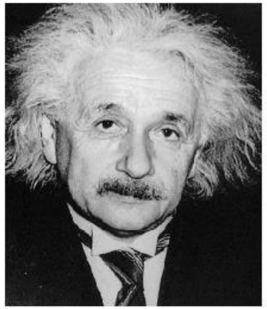
Hybrid Image Scaled



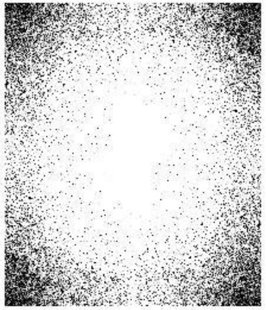
Log Magnitude spectrum of the original images

Image 1

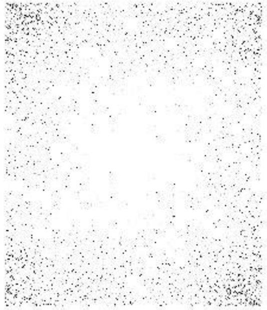
Image 2



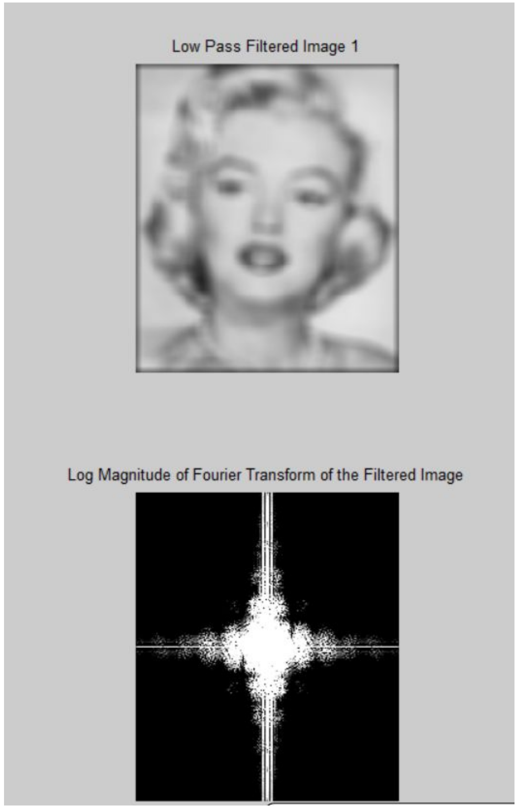
Log Magnitude of Fourier Transform of Image 1

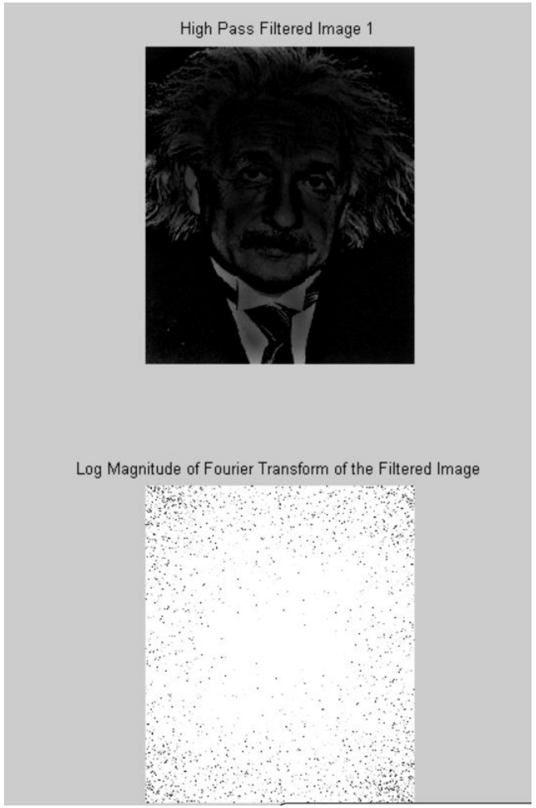


Log Magnitude of Fourier Transform of Image 2

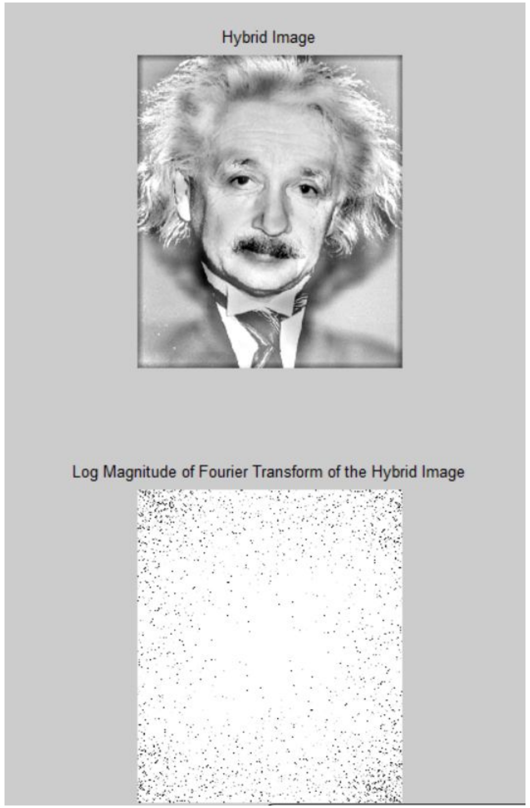


Log Magnitude spectrum of the filtered images


Low Pass Filtered Image 1
Log Magnitude of Fourier Transform of the Filtered Image


High Pass Filtered Image 1
Log Magnitude of Fourier Transform of the Filtered Image

Log Magnitude Spectrum of the Hybrid Image



Hybrid images

1. We first filter the high frequency component of an image(image1).
2. Then we filter the low frequency component of another image(image2).
3. We add these images on each other, to generate the hybrid image.

This makes the image hybrid as when viewed from a distance we see image1 and when viewed at a closer distance we see the other image, i.e. image 2.

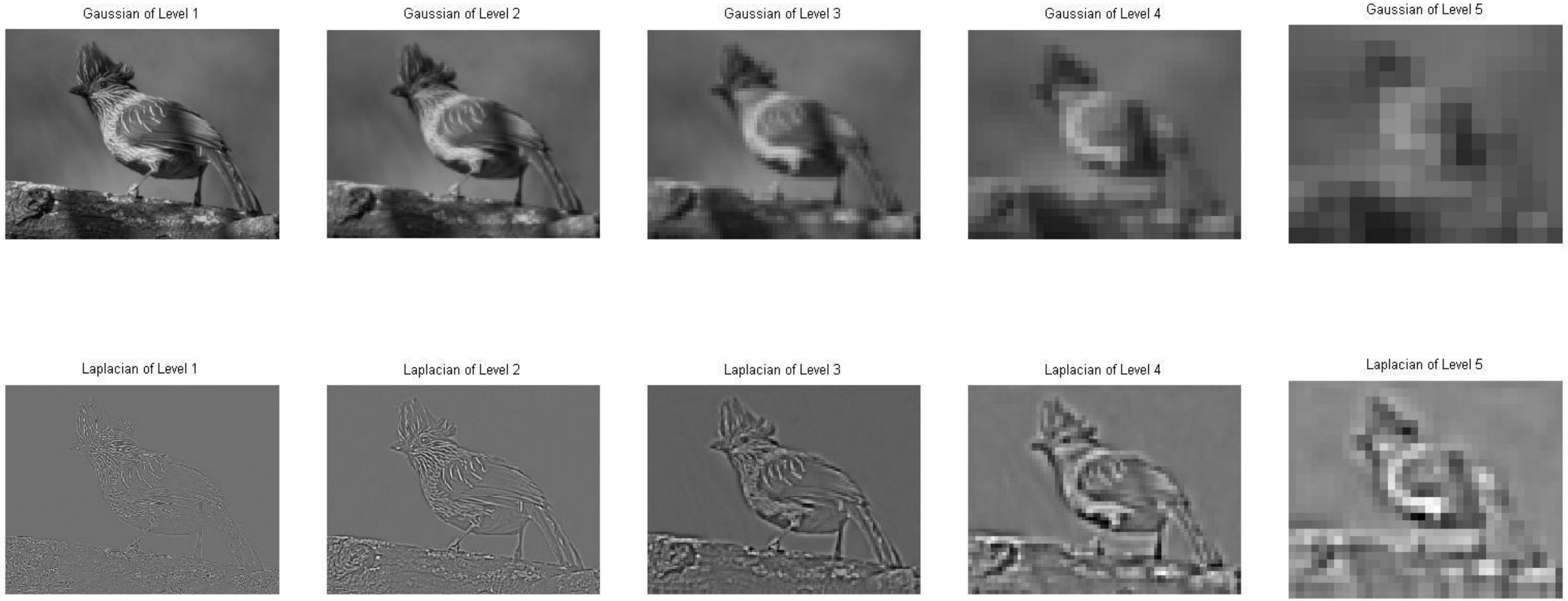
This is because our perception is based on multi-scale; global to local analysis of an image. We first do a global structure analysis and then the spatial relationships between the components are made clear to us.

http://cvcl.mit.edu/publications/OlivaTorralb_Hybrid_Siggraph06.pdf

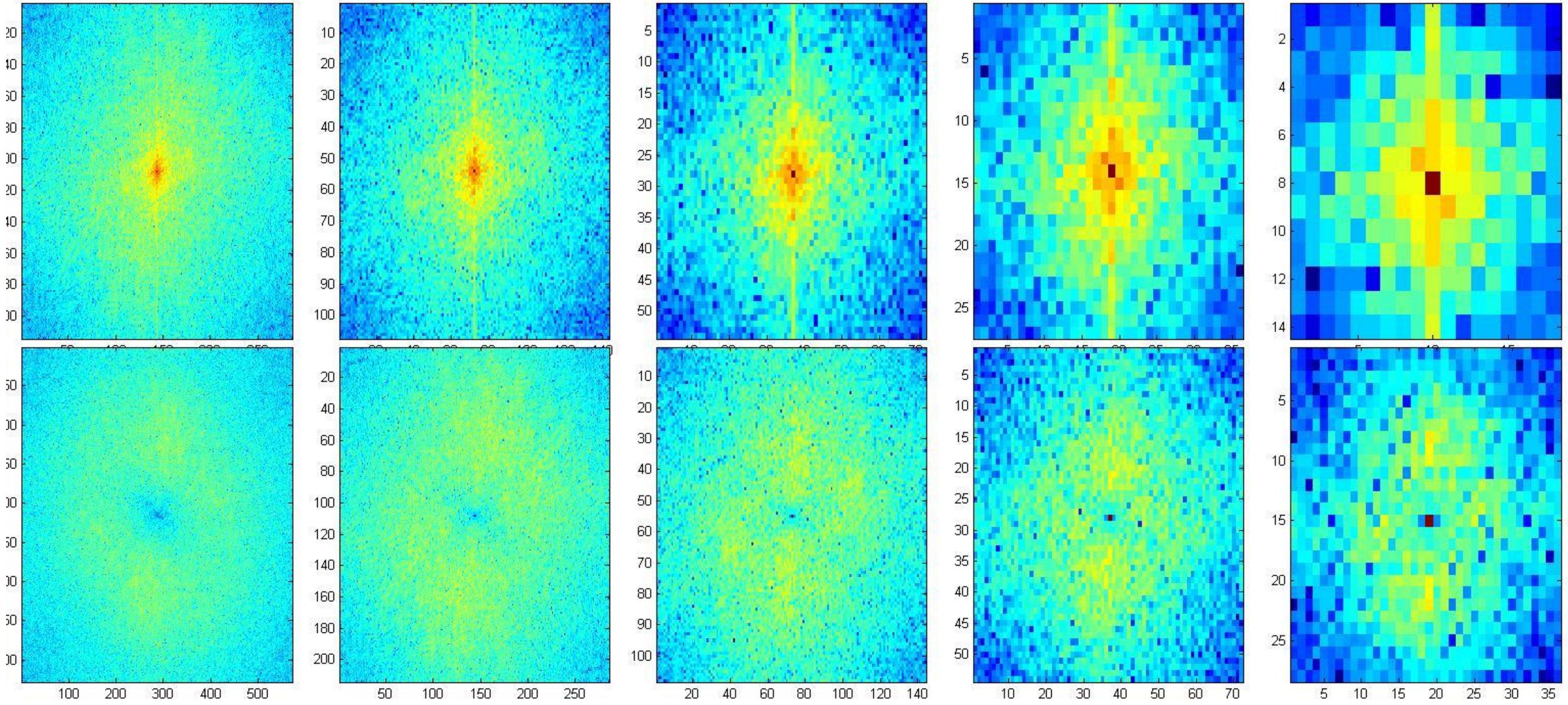
Question 2: Gaussian and Laplacian Pyramids

- 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.
- 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.

Gaussian and Laplacian Pyramids



Fourier Transforms of Gaussians and Laplacians



Explanation

The Gaussian pyramid acts as a low pass filter. With each subsequent filtering the high frequency components are lost. It essentially smoothes the image.

The laplacian on the other hand works as a band pass filter. Each of the filter band being being the difference between the successive Gaussians.

Every level of Laplacian hence captures a different frequency band.

Question 3: Edge Detection

- 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).

Part 1: Gradient Based Edge Detection

Gradient Based Edge Detection

Following are the design choices used

1. Gaussian filtering with sigma = 3, to make sure that the image is not overtly filtered and hence losing all high frequency components.
2. Used a single pixel based filter for creating gradient maps, this was done so that all edges could be detected and to prevent smoothing of edges.

gradientMagnitude.m

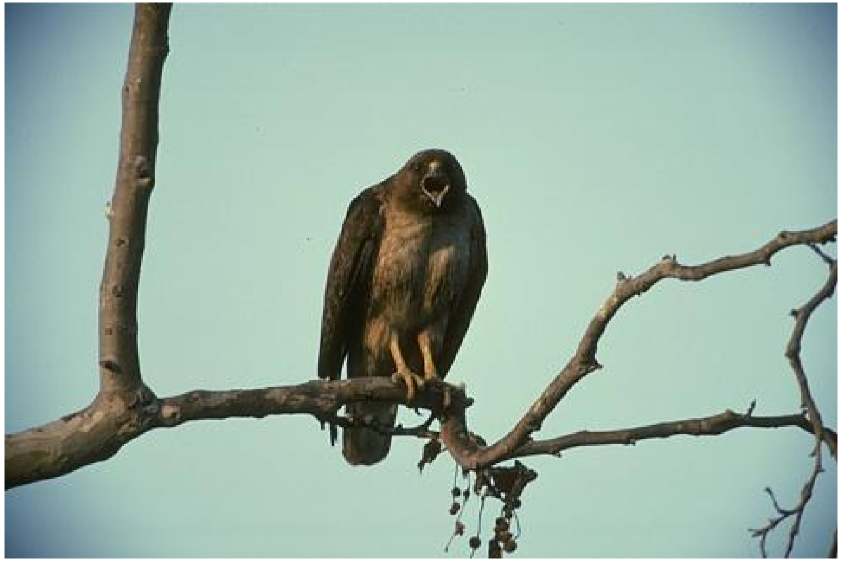
1. Cast the image to a double array
2. Filter using gaussian filter
3. Generate X and Y gradients
4. Get L2 norm of the X and Y gradients
5. Mag = L2 norm of R,G,B Components
6. Get the channel with largest magnitude of gradient
7. Calculate the theta for that channel
8. Theta = atan2(gradY/gradX)

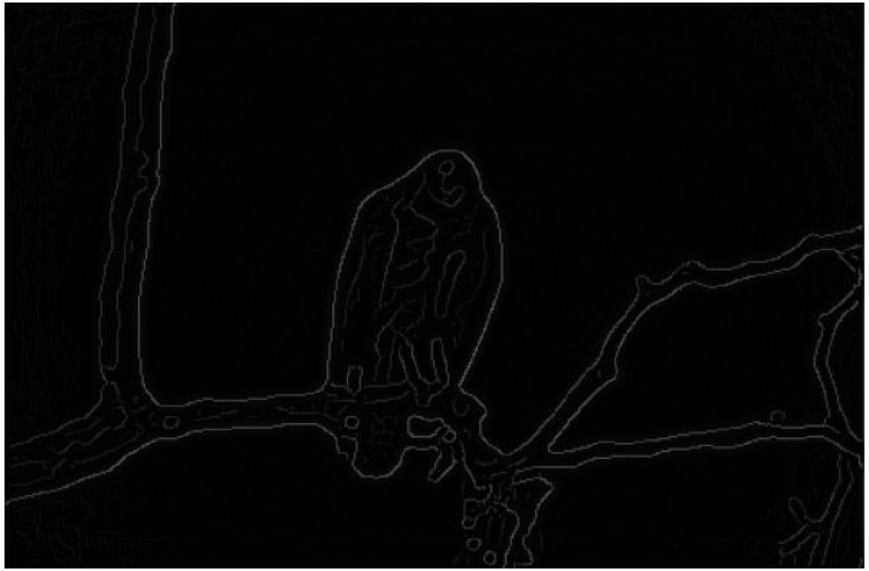
edgeGradient.m

1. Get mag,theta from gradientMagnitude(im,sigma)
2. Increase the mag contrast for better visibility mag = mag.^0.7
3. Theta = theta + pi/2 // to make sure the orientation is correct for non max suppression
4. Used the edited nonmax.m function for non max supression.

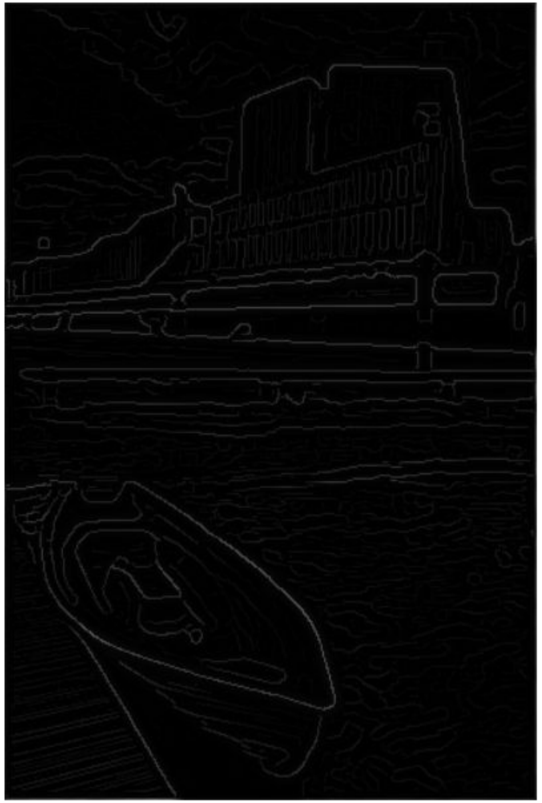
(Changed the masks used in nonmax function.)

Quantitative Results





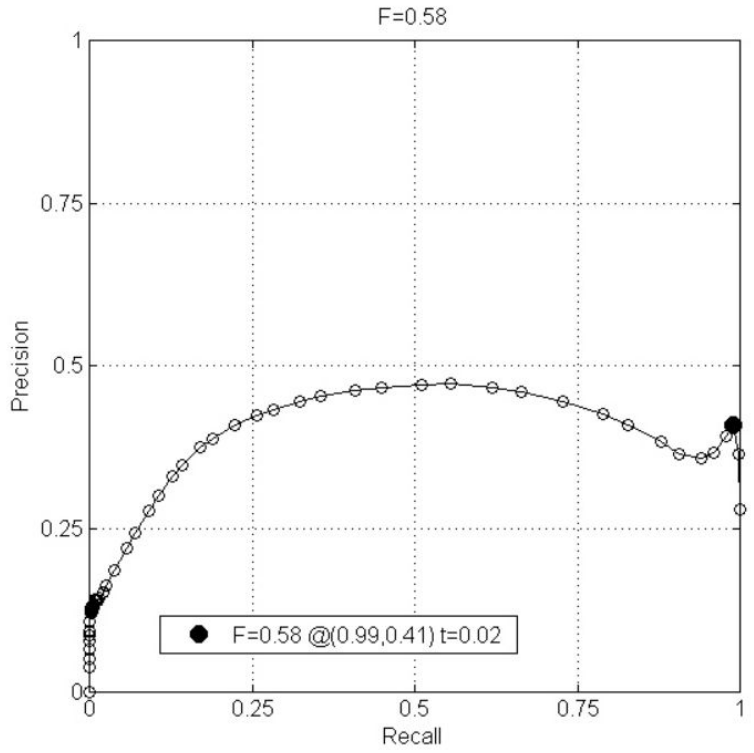
Quantitative Results - 2



F-scores

Method gradient: overall F-score = 0.580 average F-score = 0.598

Precision Recall Plots



Part 2: Oriented Filters Based Edge Detection

Oriented Filters Based Edge Detection

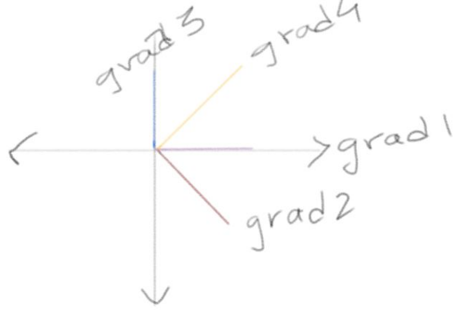
Following are the design choices used

1. Used the following filters for getting elongated gradients:-

Gaussians along x-axis, y-axis, y=x and y=-x

2. Used a single pixel based filter for creating gradient maps, this was done so that all edges could be detected and to prevent smoothing of edges.

orientedFilterMagnitude.m



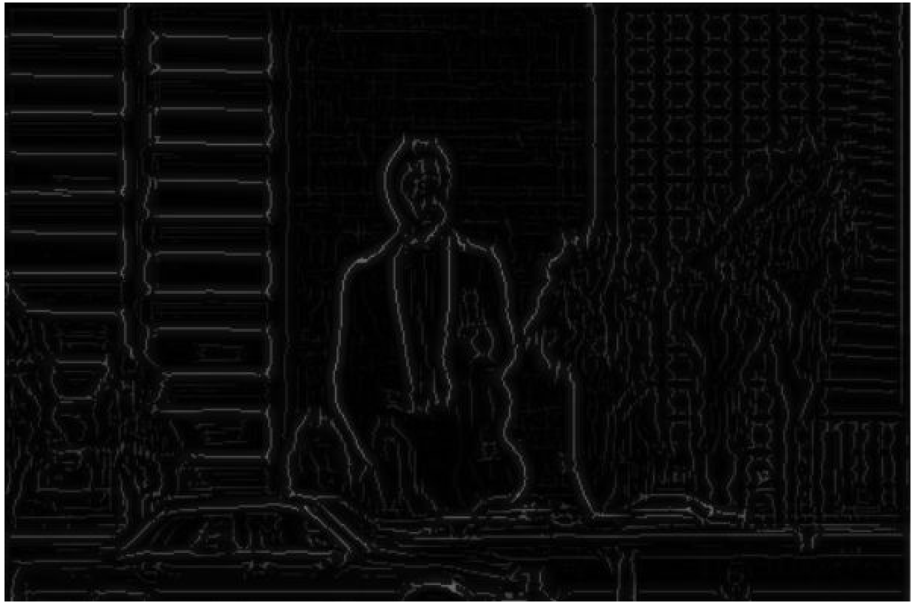
1. Cast the image to a double array
2. Filter using gaussian filter
3. Generate 4 different gradients along 4 directions
4. Gradient = max(direction of absolute maximum gradient)
5. Mag = L2 norm of R,G,B Components
6. Find theta using vector algebra, with the 4 vectors being along the 4 given directions along with their magnitudes.
7. atan2((grad1+grad2/sqrt(2) + grad4/sqrt(2))/(grad3+grad2/sqrt(2) - grad4/sqrt(2)))

orientedEdgeGradient.m

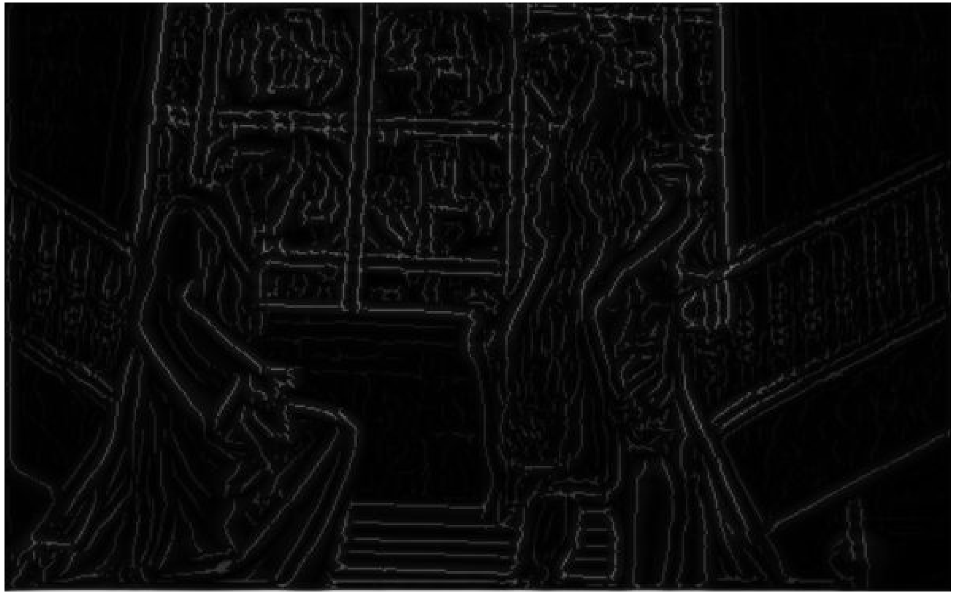
1. Get mag,theta from gradientMagnitude(im,sigma)
2. Increase the mag contrast for better visibility mag = mag.^0.7
3. Theta = theta + pi/2   // to make sure the orientation is correct for non max suppression
4. Used the edited nonmax.m function for non max supression.

(Changed the masks used in nonmax function.)

Quantitative Results



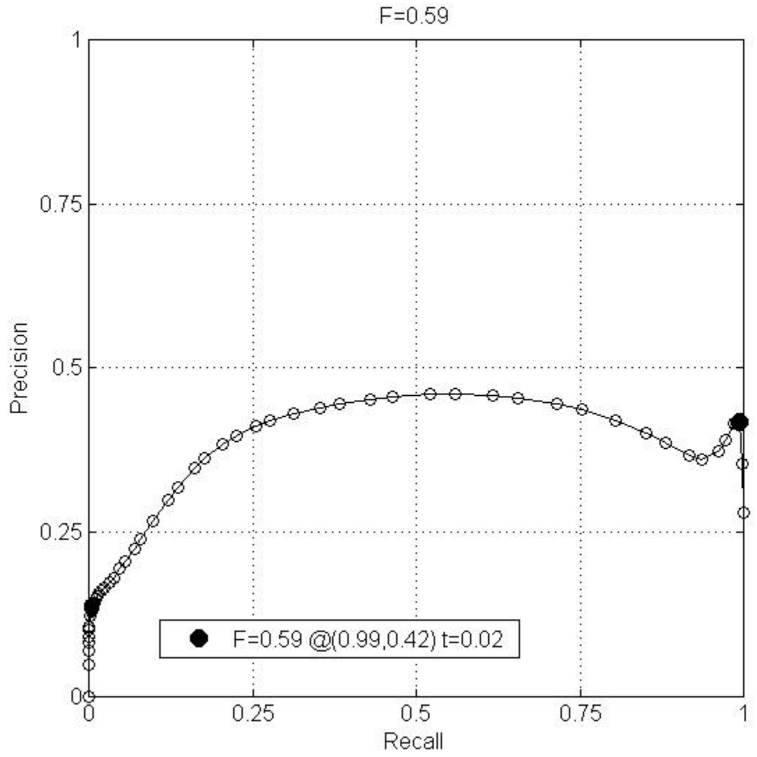
Quantitative Results - 2



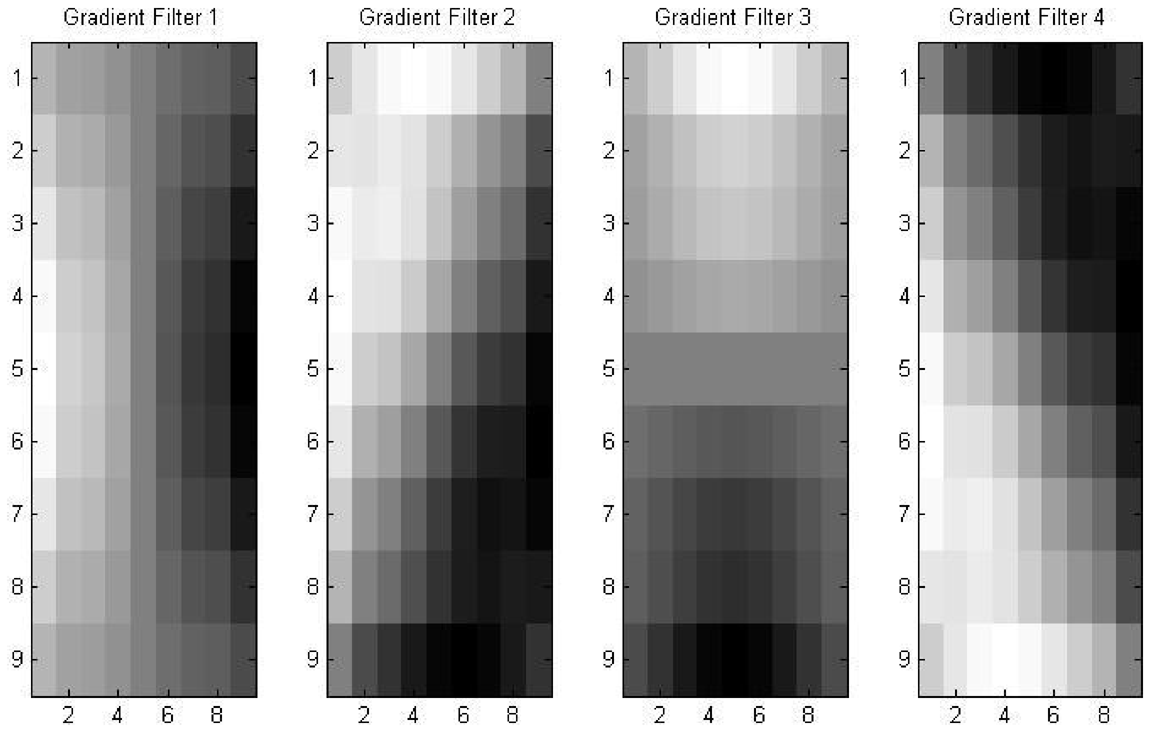
F-score

Method oriented: overall F-score = 0.589 average F-score = 0.604

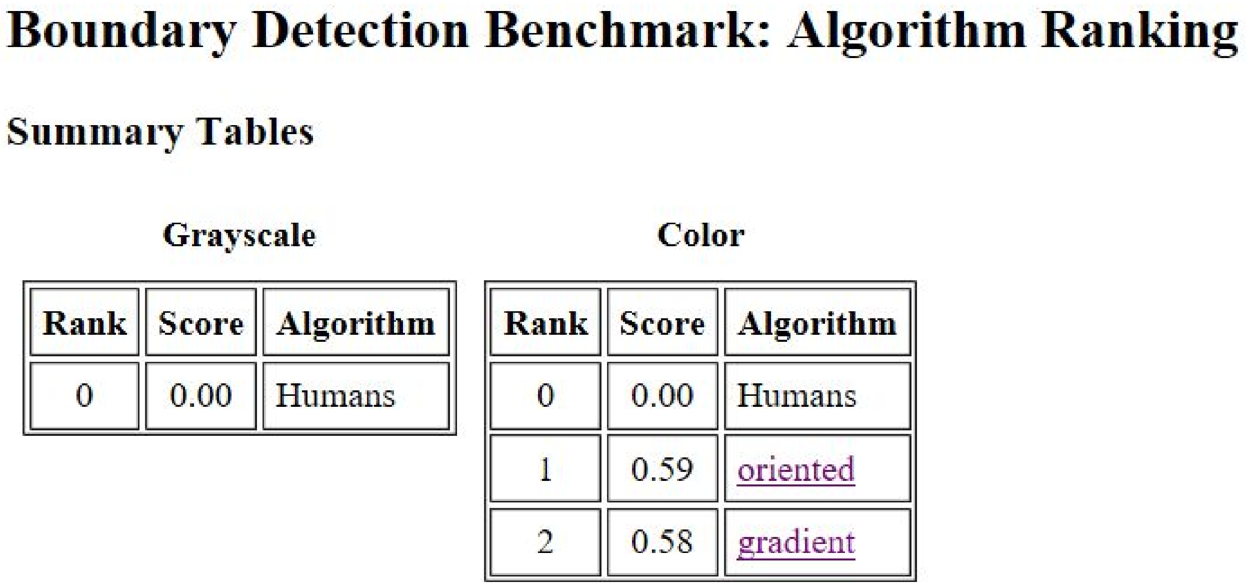
Precision Recall Plots



Bank of Filters used for Oriented Filters Approach



Overall F-Scores and their tables



Boundary Detection Benchmark: Algorithm Ranking

Summary Tables

Grayscale

Rank	Score	Algorithm
0	0.00	Humans

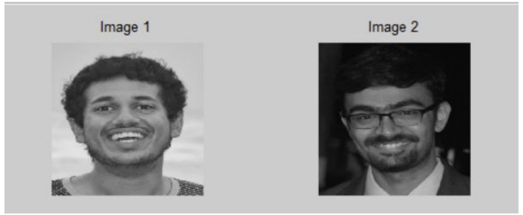
Color

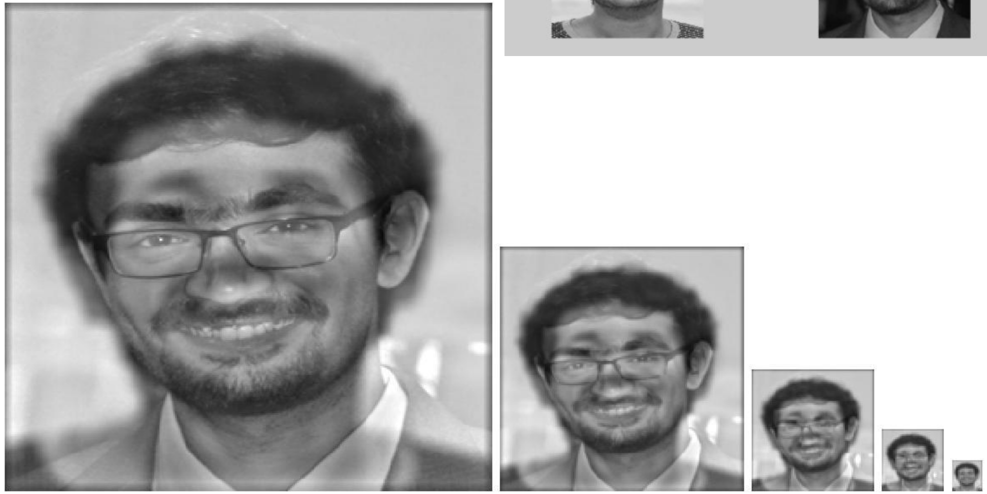
Rank	Score	Algorithm
0	0.00	Humans
1	0.59	oriented
2	0.58	gradient

Graduate Points

Question 1: Show us at least two more hybrid image results that are not included in the homework material, including one that doesn't work so well (failure example). Briefly explain how you got the good results (e.g., chosen cut-off frequencies, alignment tricks, other techniques), as well as any difficulties and the possible reasons for the bad results.

Pass test case



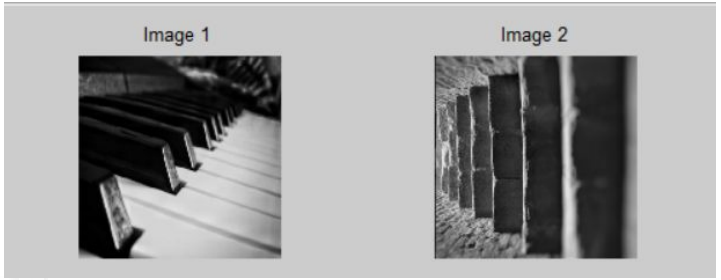


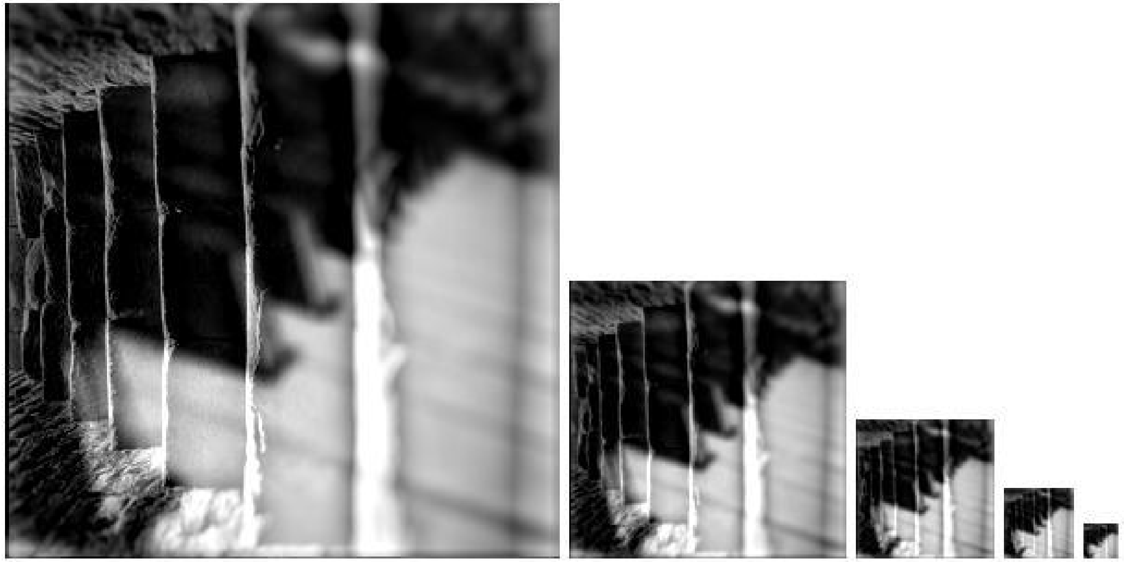
Tricks used to achieve this

1. Alignment of the dark features of the low pass image with the darker features of the high pass image. This ensures that when viewed from a distance the features appear globally correct.
2. The eyes of the two images are overlapping(aligned in Photoshop) and the different distinct hairstlyes make the image distinct at two different distances.
3. The accidental alignment of the beard and the teeth make the illusion even more better.

http://cvcl.mit.edu/publications/OlivaTorralb_Hybrid_Siggraph06.pdf

Failed Test Case





1. This is a failed test case at alignment as there is lot of symmetry and repetitiveness in the low spatial domain form a percept which acts like a bias.Hence the hybrid image is not perceived properly.

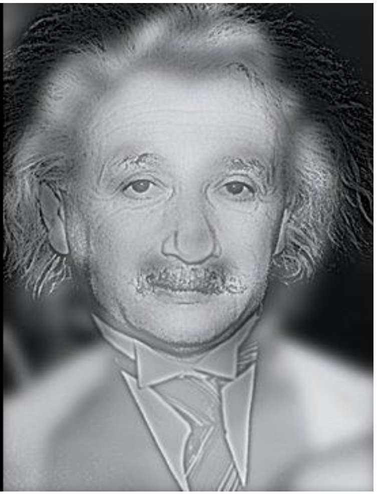
2. In a hybrid image it is necessary for us to perceive the other image as noise, which is hard when there is symmetry involved. Hence this is a failed attempt.

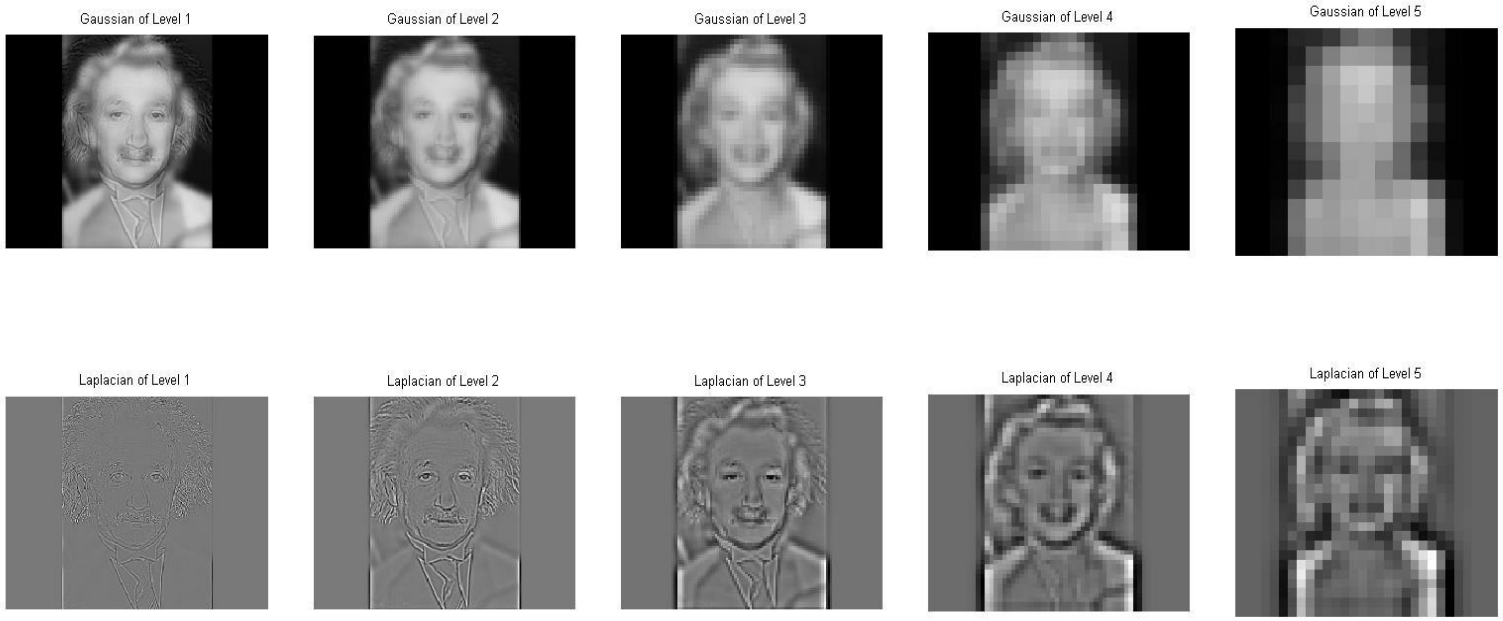
http://cvcl.mit.edu/publications/OlivaTorralb_Hybrid_Siggraph06.pdf

Graduate Points

Illustrate the hybrid image process by implementing Gaussian and Laplacian pyramids and displaying them for your favorite result. This should look similar to Figure 7 in the Oliva et al. paper.

Hybrid Image in Gaussian and Laplacian Pyramids





Explanation:-

The hybrid image when decomposed into gaussian pyramids, the low frequency components are shown dominantly, hence the original image is seen.

Whereas the Laplacian captures the high frequency component hence the the image which was high pass filtered is seen more clearly.

These pyramids are useful for decomposing the images into spatial components globally and locally.

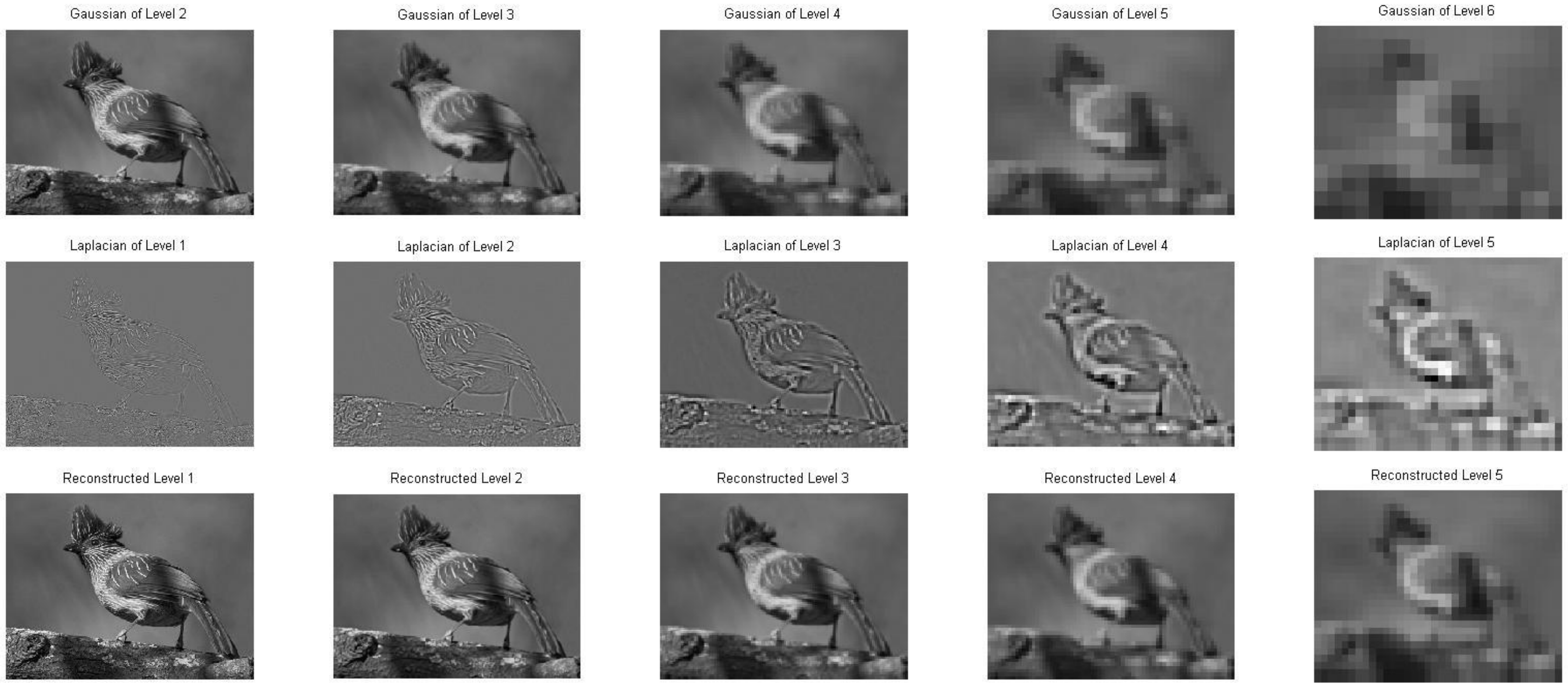
The global component corresponds to the low pass whereas the high frequency components are the locally correlated components.

Graduate Points

Implement the reconstruction process from Laplacian pyramid. Draw the diagram to illustrate how you reconstruct the original input image using Laplacian pyramid. Report the reconstruction errors.

$$\begin{tikzcd}
\text{original image} = & L_1 & + \text{smooth}(\text{upscale}(G_2)) & & & & \\
& L_1 \arrow[u] & G_2 = \arrow[u] & L_2 & + \text{smooth}(\text{upscale}(G_3)) & & \\
& & & L_2 \arrow[u] & G_3 = \arrow[u] & L_3 & + \text{smooth}(\text{upscale}(G_4)) \\
& & & & & L_3 \arrow[u] & \vdots
\end{tikzcd}$$

Reconstruction of the Image



Error Reported

For each level of the reconstruction going from 1:N, the following output is shown:-

```
error =
 Columns 1 through 5
      0.0072     0.0057     0.0058     0.0071  0.0073
```

Graduate Points

Describe at least one idea for improving the results. Sketch an algorithm for the proposed idea and explain why it might yield improvement. Improvements could provide, for example, a better boundary pixel score or a better suppression technique. Your idea could come from a paper you read, but cite any sources of ideas.

Using Different Colour Spaces

One of the methods that I read about was using different colour spaces to perform edge detection.

The features used for detection of these boundaries is taken from changes in brightness, color and texture. A set of filter banks are created based on these features.

A model is trained using human label images. It is a classifier which predicts the probability of a boundary at every pixel based on the model.

https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/papers/mfm-pami-boundary.pdf

Features Used in this Model

1. Oriented Energy - pair of quadrature even and odd symmetric filters at different orientations
2. Gradient Based Features - These are used to encode the changes in colour,brightness and texture.
    a. Colour Based Features - based on density estimation using histograms.
    b. Brightness Based Features - Based on luminosity of the image in a 2-D space.
    c. Texture Based Features - Based a set of filter banks which can capture a variety of textures.

M. Morrone and D. Burr, “Feature Detection in Human Vision: A Phase Dependent Energy Model,” Proc. Royal Soc. of London B, vol. 235, pp. 221-245, 1988.
M. Ruzon and C. Tomasi, “Corner Detection in Textured Color Images,” Proc. Int’l Conf. Computer Vision, pp. 1039-1045, 1999.

Idea of an Algorithm based on the previous paper

One of the ways which we can create a boundary detection classifier would be using only texture based filter banks.

1. Train a model to recognize which textures are more relevant in natural images.
2. The features in this model will be filter bank of various textures.
3. The output is a function of dominant texture and can be mapped from the filtermap.
4. The trained model can then be used to predict whether the each pixel is a boundary or not based on the fact that it is a part of a particular texture.