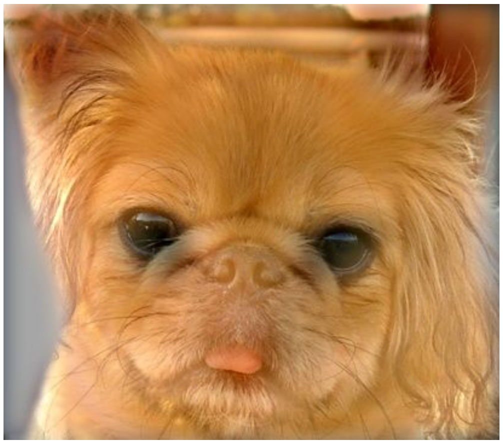
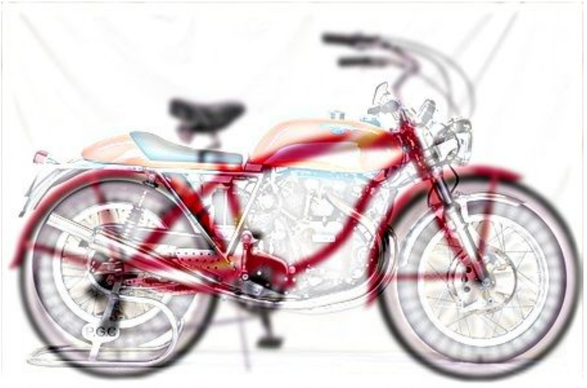
Michael Chan
RBE 549

2.2

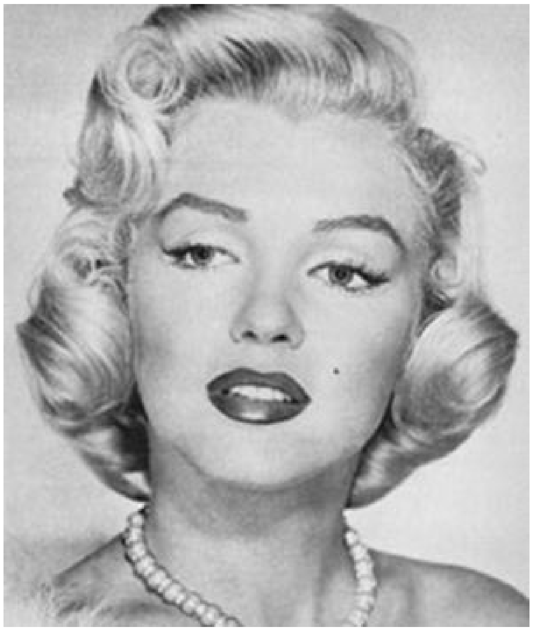
For this code, it blurs the given image, thus giving us a low-pass filtered image of the original. Then for the second image, we blur the image again, but this time, we subtract it from the original image, thus giving an only high pass filtered image. Then, it is simply a matter of adding the two images together, giving us a hybrid image.



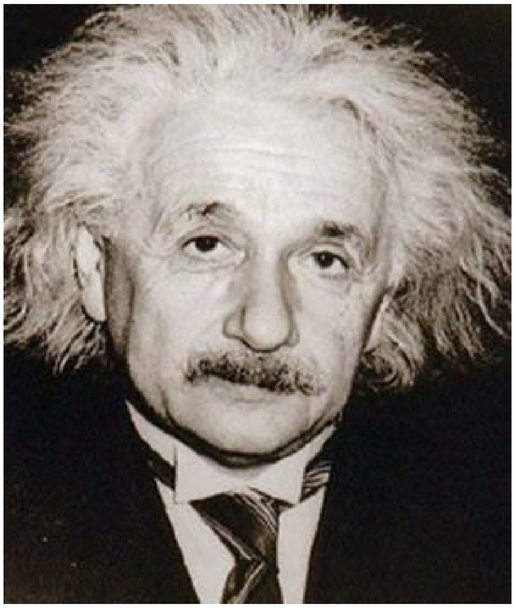
Hybrid Image 1



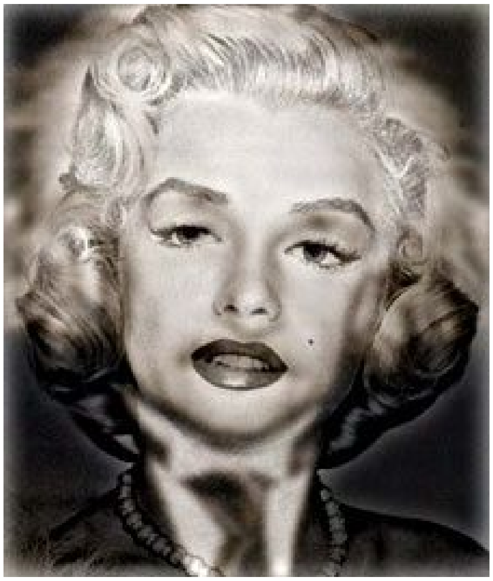
Hybrid Image 2



Original Image 1



Original Image 2



Hybrid Photo

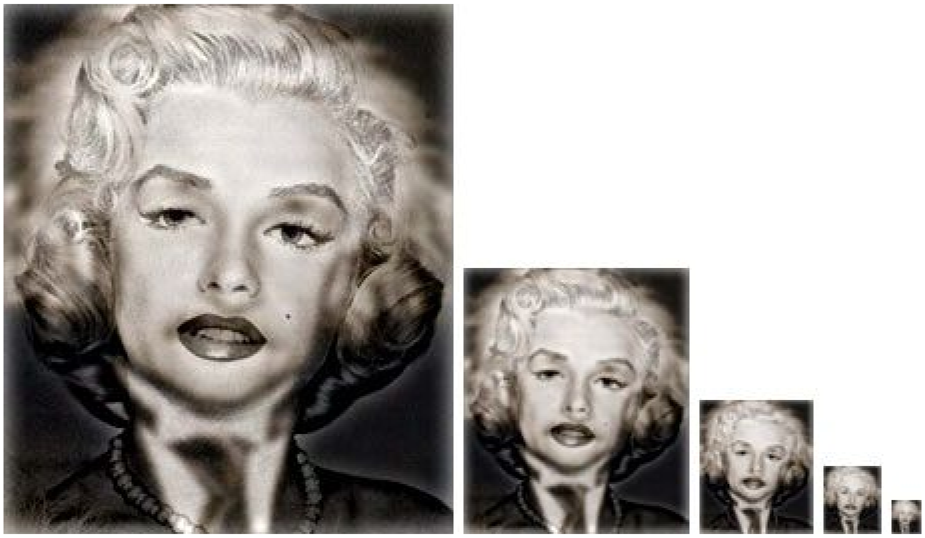


Image Scaling

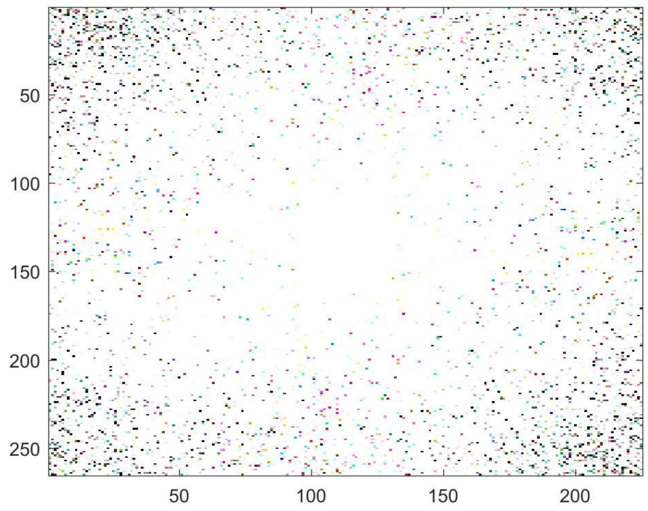


Image 1 FFT

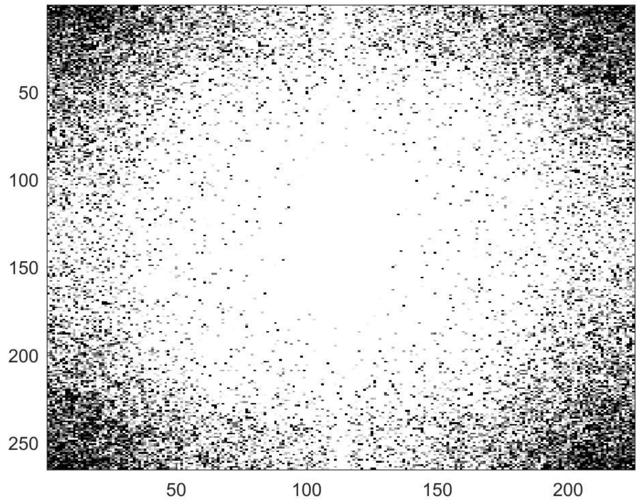
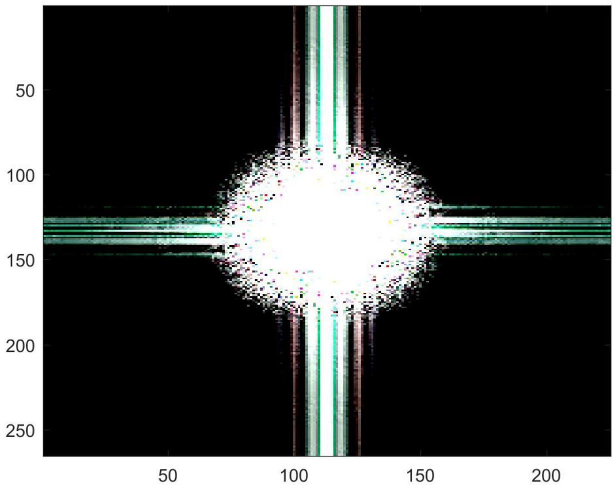
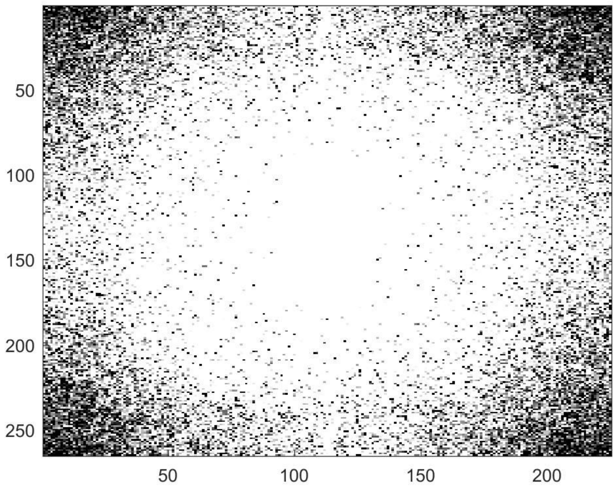


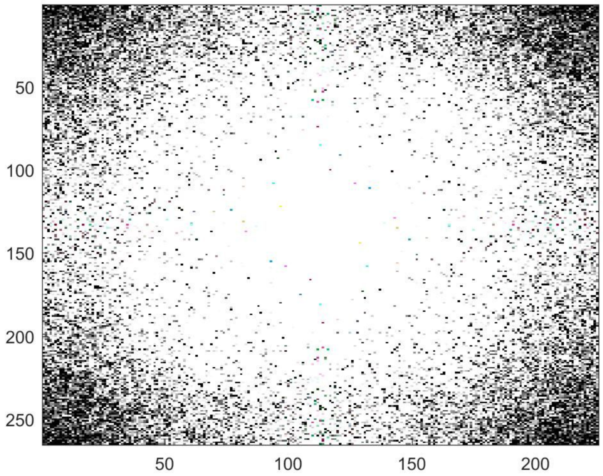
Image 2 FFT



Low Frequency FFT

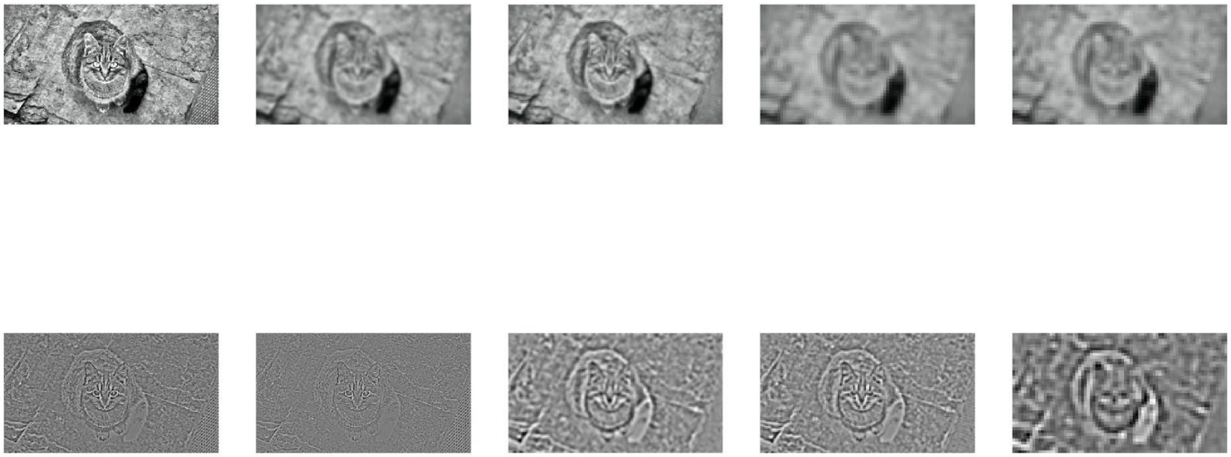


High Frequency FFT

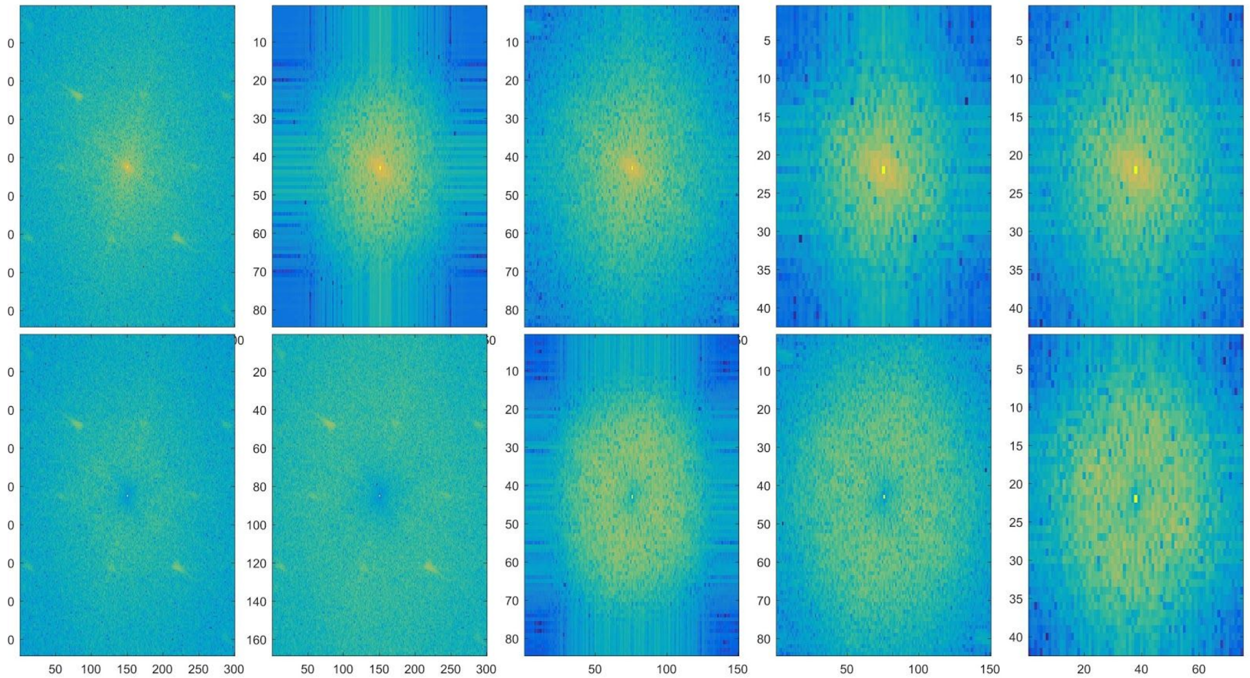


Hybrid FFT

3.2



Gaussian and Laplacian pyramids



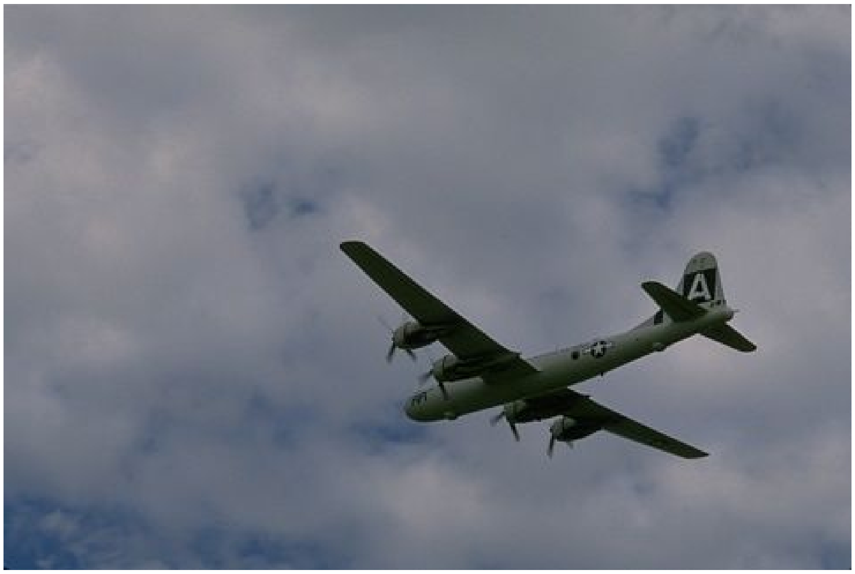
FFT images of Gaussian and Laplacian Pyramids

What’s happening is that the Gaussians are keeping the low frequencies of the image and the Laplacians are taking the high frequencies.

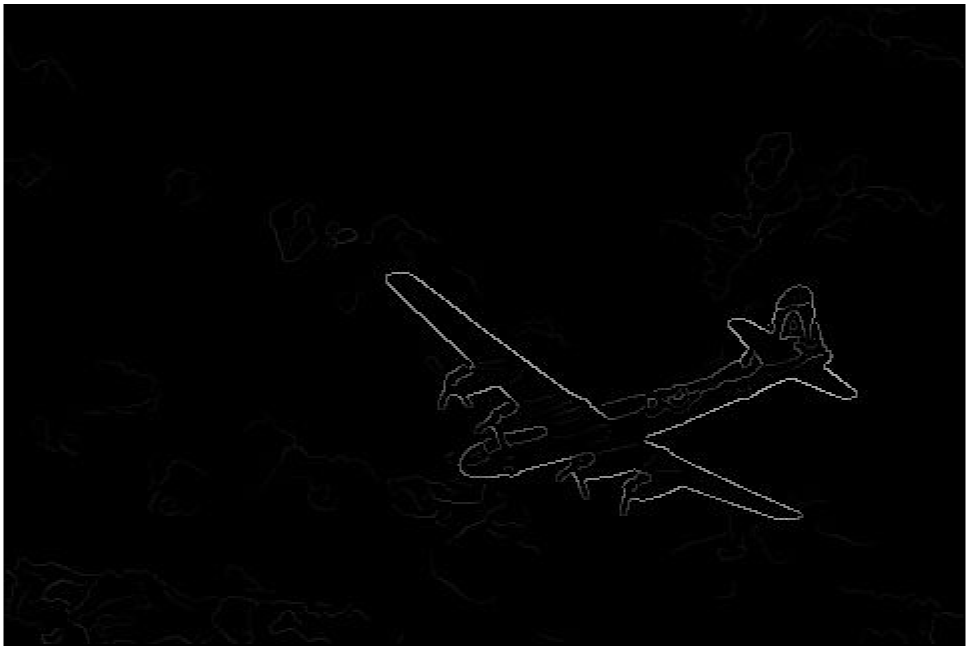
4.2

The way this code works is that it blurs the image with a gaussian filter, then splits the resulting image up into 3 channels, red, blue and green. It then calculates the magnitude for each channel, and the max of those three channels is then used to calculate the theta.

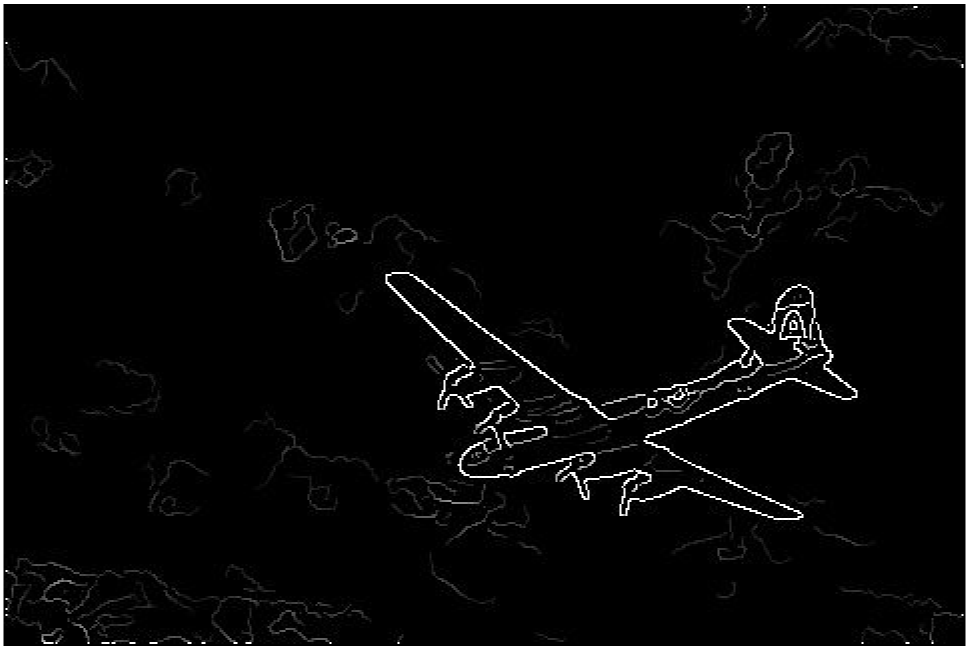
I used 20 orientations linearly spaced from 0 to 2pi. I calculated the best pixel value for each and used that as the magnitude.



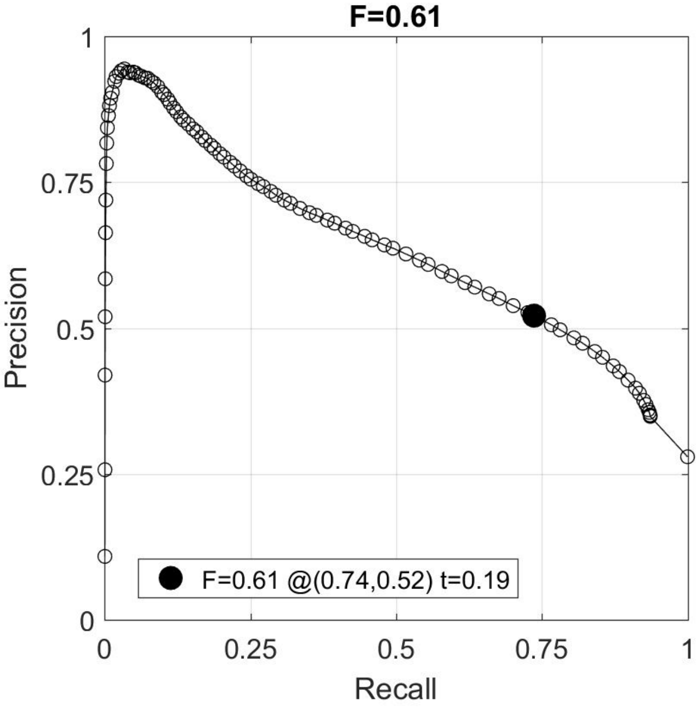
Original Image



Edge Gradient Image

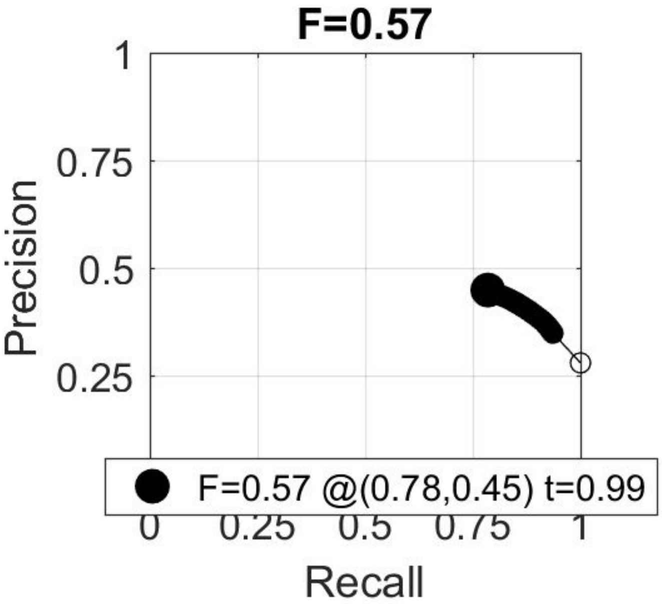


Edge Oriented Image



PR score for gradient method

Method gradient: overall F-score = 0.610 average F-score = 0.648



PR value for oriented method

Method oriented: overall F-score = 0.571 average F-score = 0.599