Biomedical Image Processing - Image Edge Detection & Enhancement

Paul Vu
301169550
February 24th 2017

SFU

Project Overview

In this project I implemented a program that can successfully identify edges in an image using a Laplacian mask. I developed my own convolution method to apply this mask to pixels across the image and display my results. I also developed a second algorithm to further enhance the edges in an image and investigate ways to improve edge detection. Fig.1 illustrates the results of edge detection using a Laplacian mask.

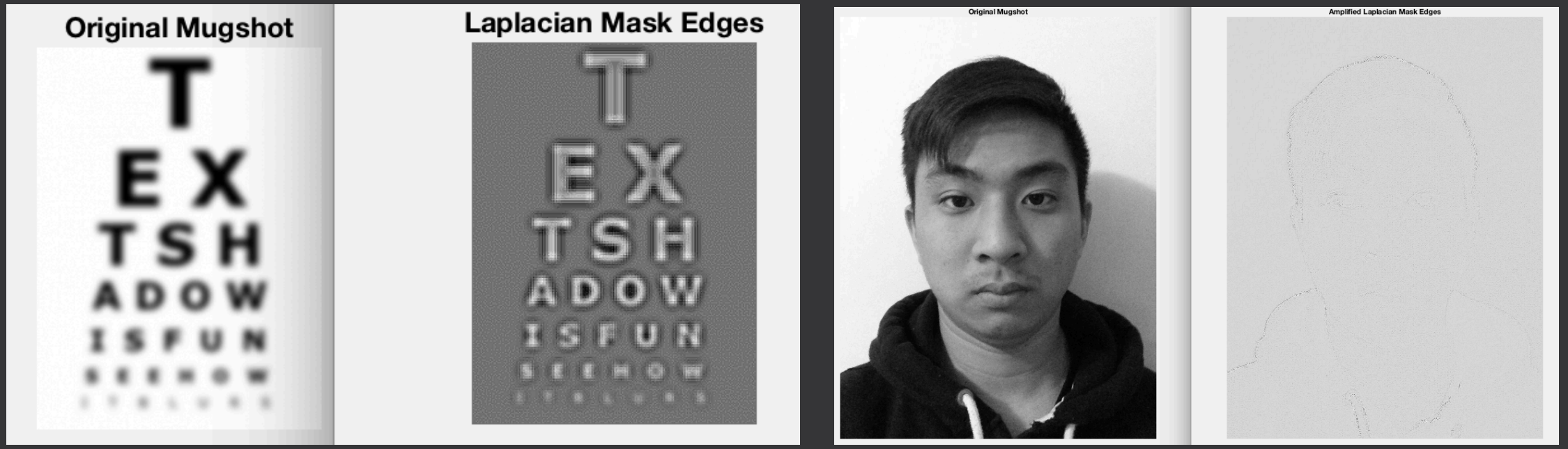

Fig.1 Example output of Laplacian Mask Function

Laplacian Mask

A Laplacian Mask can be approximated to the simple matrix, where the gradient of an image can be approximated to the 3x3 matrix as shown below. Taking this matrix and masking each pixel to it's surrounding neighbor pixels and summing the total of the matrix we will have a newly weighted value for each pixel in the image. As a result, pixels which differ largely from it's neighbor pixels (edges) will be smoothened out. Equation 1 illustrates my methodology futher.

$$\nabla^2 f = \partial^2/\partial x^2 + \partial^2/\partial y^2$$

$$\partial^2/\partial x^2 \approx f(x+1, y) - 2f(x, y) + f(x-1, y)$$
$$\partial^2/\partial x^2 \approx f(x, y+1) - 2f(x, y) + f(x, y-1)$$

0	1	0
1	-4	1
0	1	0

Equation 1: Laplacian Mask Approximation

Algorithm Implementation

Given a gray scaled image, I will only apply the Laplacian mask to an MxN image from rows 2-row_length-1 and columns 2 – col_length-1. This is because the laplacian mask is a 3x3 matrix and it is not possible to apply the mask to pixels which reside on the edges of an image, since pixels x+1, x-1, y+1 and y-1 will be out of bounds. For these pixels I simply ignore them in my calculations which is okay given a large MxN image i.e 1280x960 since the pixels ignored is only 4476 / 1228800 ≈ 0.0036%. Fig.2 shows exact code used in my Laplacian function.

```
%takes in an image and finds the edges of the image, returns 2 images which
%illustrate edges and 2 enhanced versions of that image
function [img_edges, amplified_edges, enhancedImg1, enhanced_Amplified_img] = mask( img )


imgSz = size(img); % read in image size
rowLength = imgSz(1,1); % M length
colLength = imgSz(1,2); %N length
row_boundary = rowLength - 1; % since the laplacian mask is a 3x3 matrix I will only be evaluating the pixels which are not on the outer edges of the MxN images...
col_boundary = colLength -1; %...Those pixels will remain their original values without any masking
laplacian_mask = [0,1,0; 1, -4, 1; 0,1,0];%laplacian mask which will be used to find edges in an image
img_edges = zeros(rowLength, colLength); % initialize output array

for J = 2:row_boundary
    for I = 2: col_boundary
        img_matrix = [img(J-1, I-1:I+1); img(J, I-1: I+1); img(J+1, I-1:I+1);]; %reading in the 3x3 values of the image centered around a pixel at I,J
        temp_matrix = laplacian_mask .* img_matrix; %multiplies the image matrix against the laplacian mask
        sumCol = sum(temp_matrix); % sums the 3x3 matrix into a 1x3 matrix holding the sum of each column
        sumTotal = sum(sumCol); % sums all of the columns into a singl value representing the sum of the entire matrix
        img_edges(J,I) = sumTotal; % update the image value


    end

end
```

Fig. 2 Overview of Laplacian Mask function

Laplacian Mask Results

Given my original portrait image, we can see that the Laplacian mask can successfully pick up some edges of my body and face. However, since every pixel is now averaged out due to the approximations of it's neighbor values most of the image is now light grey aside from the edges which are slightly sharpened and darker. The results are shown in Fig. 3.

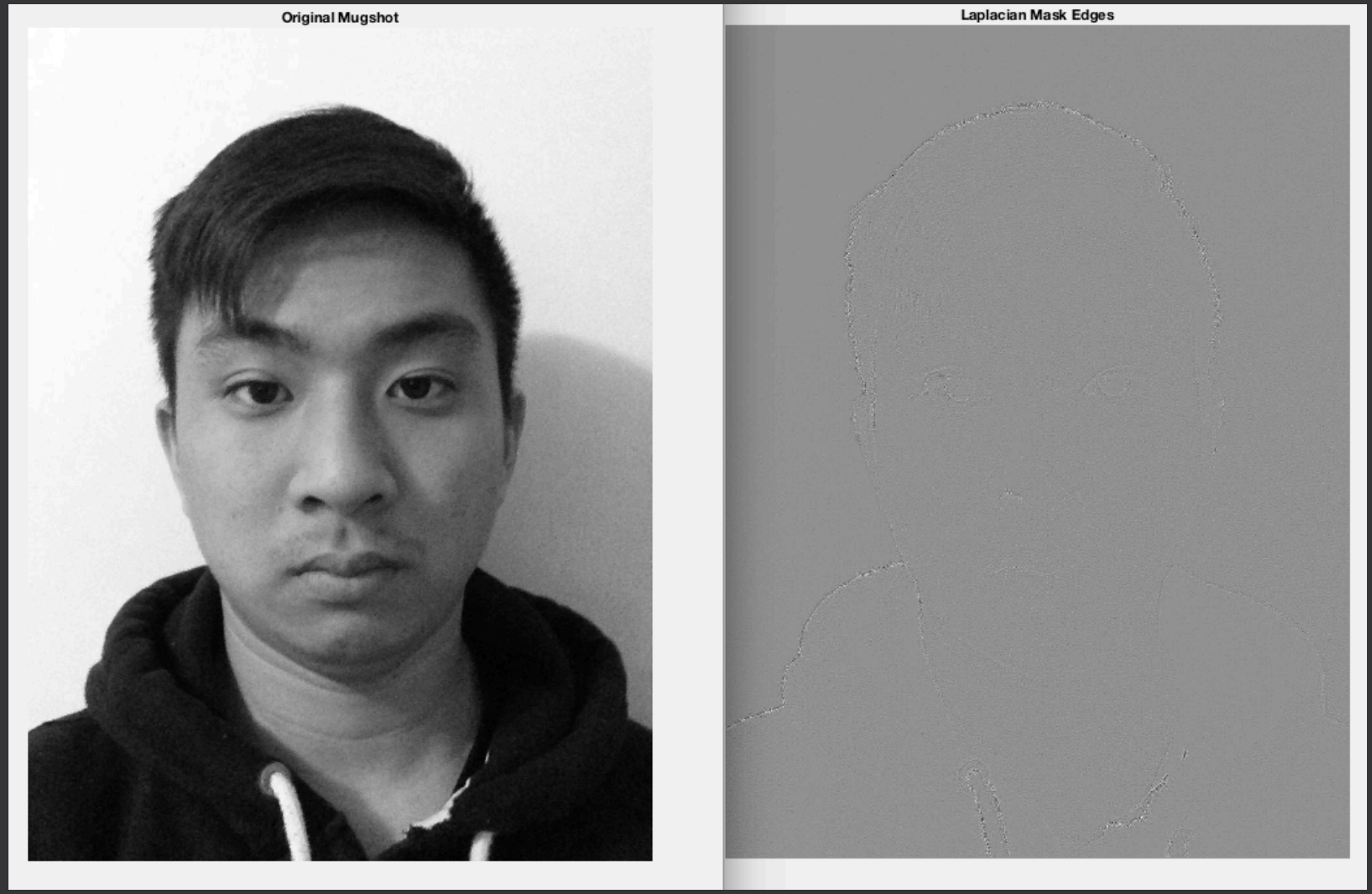

Fig. 3 Edge detection using Laplacian Mask

Amplification of Edges

I attempt to amplify my edges by checking for dark values and amplifying their values. In Matlab grey scale, 0 represents black pixels while 1 represents white. Since the Laplacian mask makes values more negative due to the -4 in the 3x3 matrix, our image when displayed must be scaled such that the minimum value is now 0 while the maximum value is scaled to one. Therefore, in the img_edges matrix dark pixels could be any value less then 0 theoretically, which is why any value less then 0 I will want to amplify to make it even more minimal. When calling imshow and specifying the min and max re-scaling these newly amplified values will greatly contrast the non-edge values, thus making the edges more clear. In my equation I chose 4 arbitrarily, increasing this number will provide more clear edges. Code for this algorithm is provided in Fig. 4.

```
amplified_edges = zeros(rowLength, colLength); % initialize output array which will have an amplified image of the edges
for K = 1: rowLength*colLength
    if img_edges(K) < 0 %if a pixel value is less then 0 then it's a dark pixel so let's enhance it a bit to make it more clear that it's an edge
        amplified_edges(K) = img_edges(K).*4; % enhancing the edges by a scale of 4
    else
        amplified_edges(K) = img_edges(K);
    end

end
```

Fig. 4 Enhanced Edge Algorithm

Edge Amplification Results

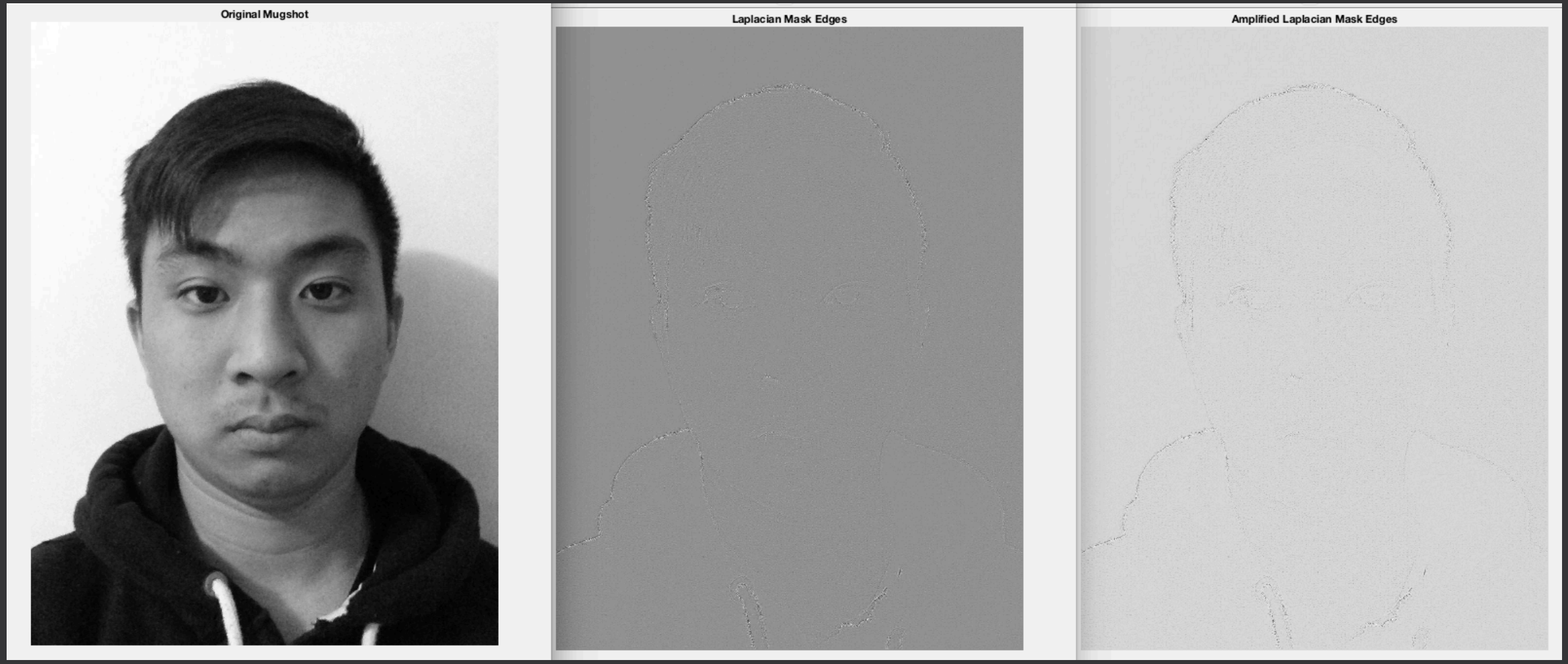

Fig. 5 Results of Edge Detection Amplification

Laplacian Mask Results

As shown below, we can see that the edges have been greatly enhanced in contrast to the non edges in the photo due to the amplification of darker pixels. In addition, when scaling the minimum pixel to 0 and maximum pixel to 1 we can see that the background is less grey since there is a larger difference between edge values and non edge values. The results are shown in Fig. 5.

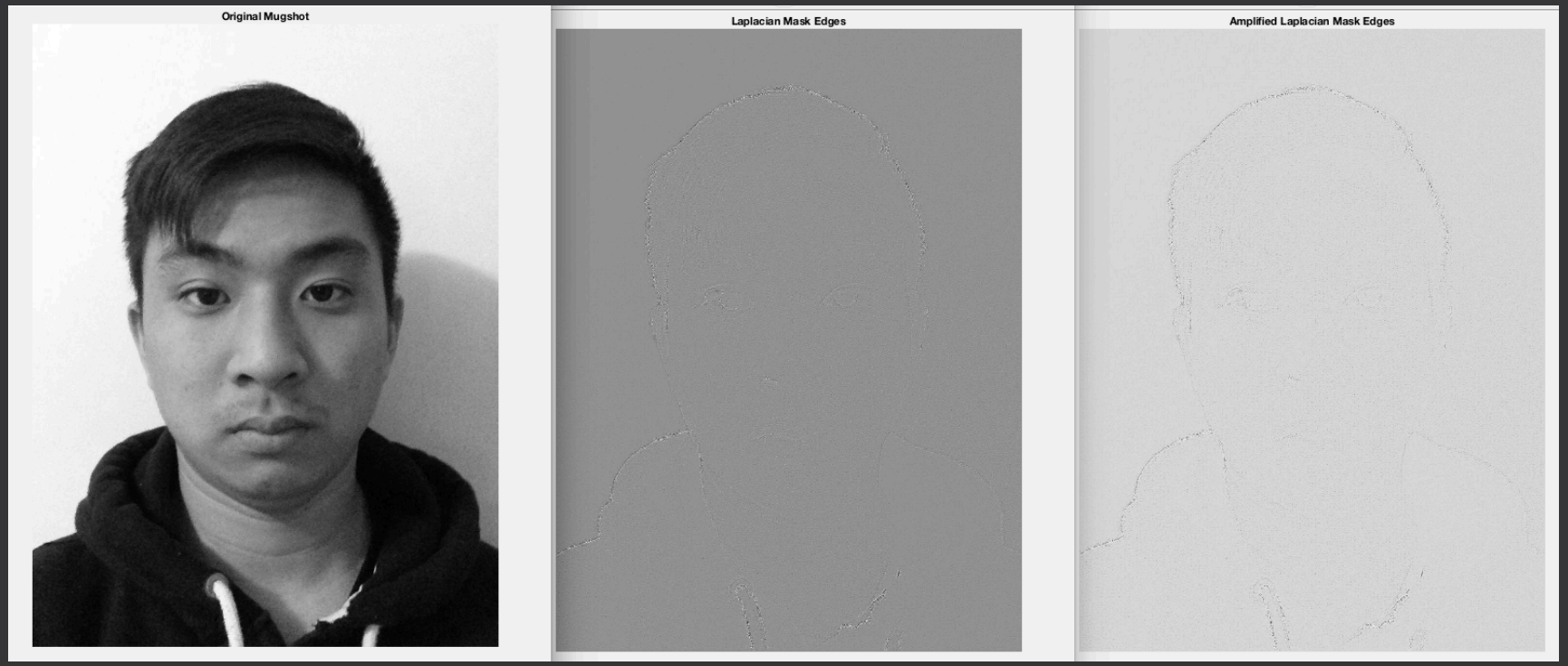

Fig. 5 Results of Edge Detection Amplification

Enhanced Image w/ Regular Edge Detection

To show an enhanced image I subtract the original image with the results of the edge detection image. Since the edge detection image is more grey (dark) due to the negative scaling at the center pixel by -4, the image looks naturally darker. By subtracting the edge values against the image we can also see the enhanced image is sharper around the edges. Fig. 6 illustrates the results found below.

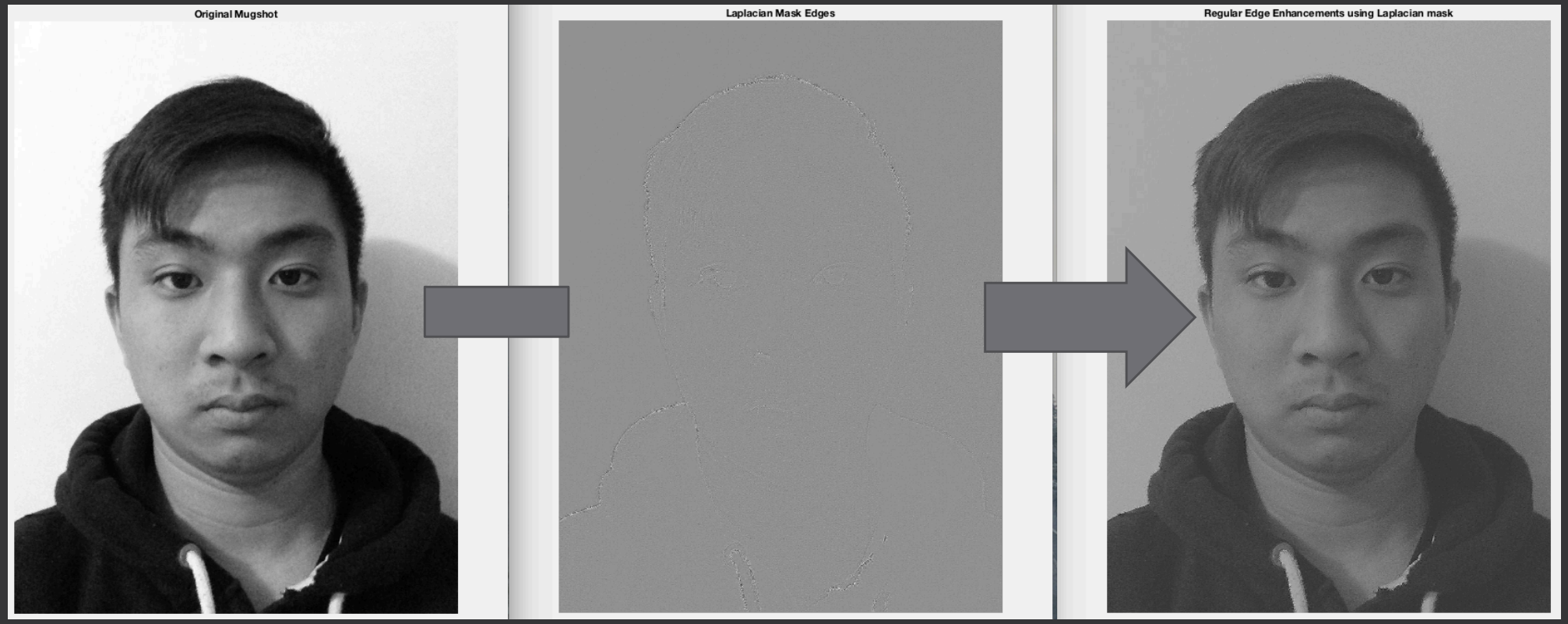

Fig. 6 Enhanced Image using Laplacian Mask subtraction

Enhanced Image w/ Amplified Edge Detection

Subtracting the more amplified edge detection results in a darker image. This is intuitive since I amplified the edge values to be even more negative then they already were, so when MATLAB scales the minimum and maximum values in the “imshow” function a darker image is shown. In this image we can clearly see the edges from the subtracted image in comparison to the regular Laplacian mask. Results are provided in Fig. 7.

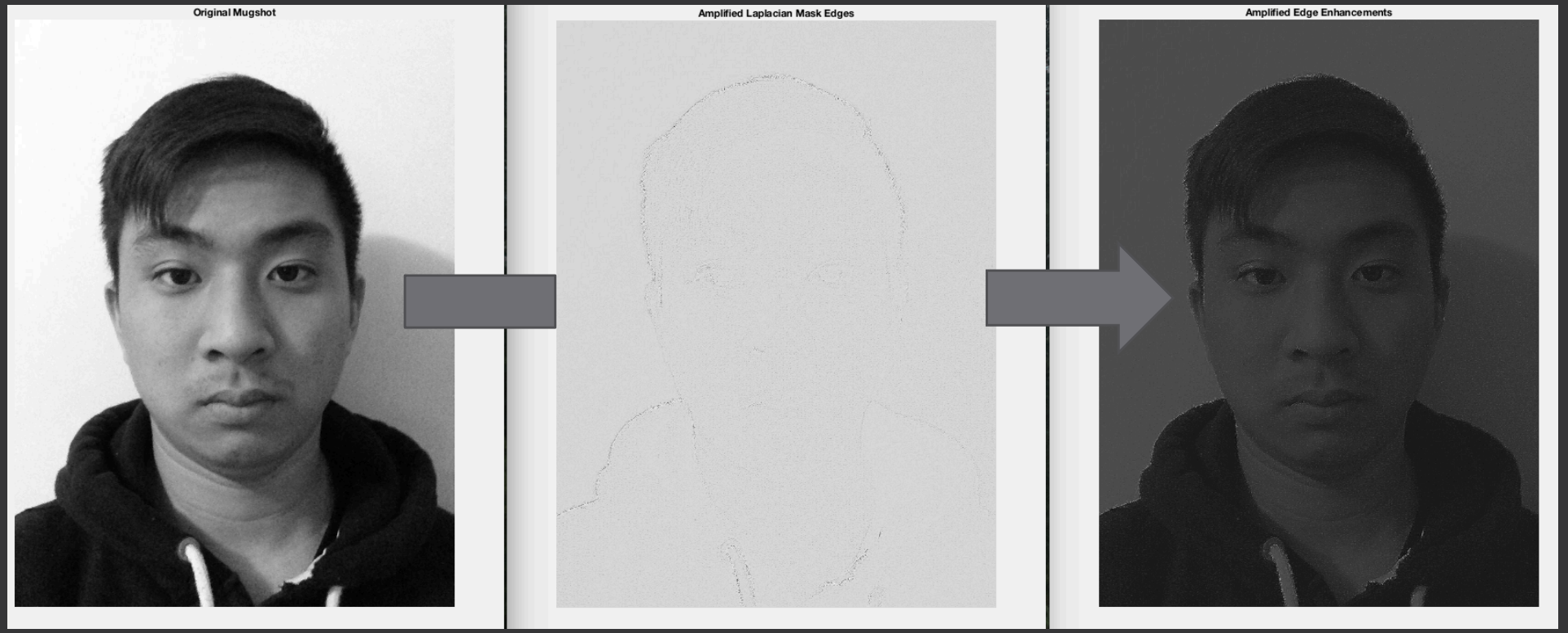

Fig. 7 Image enhancement using edge detection amplification