# Fatemeh Darbehani [redacted]

ENSC474 – SFU – Spring 2017

Assignment 5

# Task 1: Edge Detection

- In this assignment I implemented an algorithm to find edges in my mugshot.
- Since an image is represented as a discrete pixel values, edges can be identified as rapid intensity changes between two neighboring pixels.
- One way to find these rapid changes is to look at the second derivative of the pixels.
- The ***Laplacian*** is an equivalent measure of the second derivative in 2D and can be calculated using the following formula:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Laplacian of a 2D function

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Laplacian of a discrete 2D function

$$\Delta = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Equivalent 3x3 Laplacian mask of a discrete 2D function

# Task 1: Edge Detection

- To apply the Laplacian mask, I implemented my own convolution function that does the following:
  - Flips the mask in both x and y directions to find its 180 degree rotation.
  - Reshapes the mask to a column to make calculation easier
  - Zero-pads the image by half of mask's size at each side.
  - Loops through the image and reshapes the area covered by the mask to a row vector.
  - Multiplies this row vector by the mask vector to find the sum.

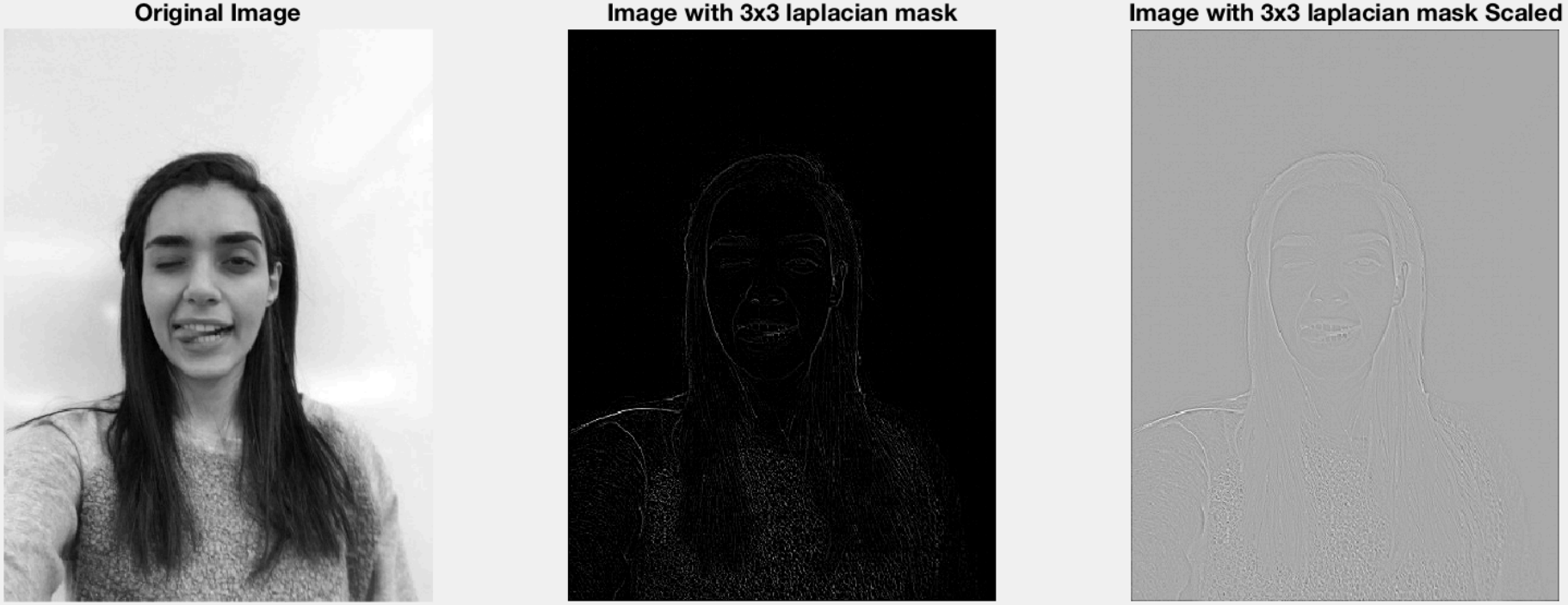
```
%Flip the mask matrix horizontally and vertically for convolution
maskVect = reshape(flip(flip(mask,1),2),[], 1);
% ------------------------ %
% Defining the output image %
% ------------------------ %
Z = zeros(ir, ic);
% ---------------------------------- %
% padding image with zeros           %
% ---------------------------------- %
paddedr = mr + ir - 1; paddedc = mc + ic - 1;
paddedImage = zeros(paddedr, paddedc);
paddedImage(floor(mr/2) + 1:paddedr - floor(mr/2), ...
            floor(mc/2) + 1:paddedc - floor(mc/2)) = image;
% ---------------------------------- %
% Looping through the padded image   %
% to perform convolution in spatial domain %
% ---------------------------------- %
for i = 1:ir
    for j = 1:ic
        %Vectorizing the padded image under mask (Row Vector)
        ImageVect = reshape(paddedImage(i:i+mr - 1, j:j+mc -1), 1, []);
        Z(i, j) = ImageVect * maskVect;
    end
end
```

Figure1: MATLAB Convolution Function

# Task 1: Edge Detection

- The result of convolution of the mask and mugshot image is shown below:
  - Because the Laplacian is a derivative operator, it highlights intensity discontinuities in the image and deemphasizes regions with slowly vary in intensity.
  - Applying this mask will tend to produce images with grayish edge lines and other black background.
  - I scaled the Laplacian image by adding to it its minimum value to bring the new minimum to zero and then scaled the result to the full [0, 255] intensity range.
  - This resulted in the image on the right

Figure2: Original, Laplacian and scaled Laplacian images



# Task 2: Edge Enhancement

- To sharpen the edges, I subtracted the Laplacian image from the original image so the background is preserved and the edges are sharpened.
  - I observed that the sharpened image is quite a bit darker than the original image. This is because the image doesn't have the same dynamic gray-scale range as the original image.
  - To fix that, I found an appropriate range of the values that have the same dynamic gray-scale range as the original image and displayed the enhanced image in that range.
  - In the final image, we can clearly see that the edges are sharpened by this filter.

Figure 3: Original, Enhanced, and Scaled image