|  |  |
| --- | --- |
| Name: | Edward Eisenberger |
| ID# | 1066164 |
| Assignment 2 | |
| Due Date | February 18, 2019 |
| Date of Submission | February 17, 2019 |

Contents

[Summary 4](#_Toc1301185)

[Input 4](#_Toc1301186)

[Convolution Filter Kernel Derivation 5](#_Toc1301187)

[Identity 5](#_Toc1301188)

[Kernel Derivation 5](#_Toc1301189)

[Output 5](#_Toc1301190)

[Average (Low pass) 6](#_Toc1301191)

[Kernel Derivation 6](#_Toc1301192)

[Output 6](#_Toc1301193)

[High Pass 7](#_Toc1301194)

[Kernel Derivation 7](#_Toc1301195)

[Output 8](#_Toc1301196)

[Sharpening 8](#_Toc1301197)

[Kernel Derivation 8](#_Toc1301198)

[Output 9](#_Toc1301199)

[Gaussian 9](#_Toc1301200)

[Kernel Derivation 9](#_Toc1301201)

[Output 10](#_Toc1301202)

[First Derivative (Gradient) 11](#_Toc1301203)

[Kernel Derivation 11](#_Toc1301204)

[Output 12](#_Toc1301205)

[Second Derivative (Laplacian) 13](#_Toc1301206)

[Kernel Derivation 13](#_Toc1301207)

[Output 14](#_Toc1301208)

[Difference of Gaussians (σ1=0.5, σ2=2.5) 15](#_Toc1301209)

[Kernel Derivation 15](#_Toc1301210)

[Output 15](#_Toc1301211)

[Figure 1 - Input 1: Black and White Husky 4](#_Toc1301212)

[Figure 2 - Input 2: Landscape 4](#_Toc1301213)

[Figure 3 - Identity Example 1 5](#_Toc1301214)

[Figure 4 - Identity Example 2 6](#_Toc1301215)

[Figure 5 - Average (Low Pass) Example 1 6](#_Toc1301216)

[Figure 6 - Average (Low Pass) Example 2 7](#_Toc1301217)

[Figure 7 - High Pass Example 1 8](#_Toc1301218)

[Figure 8 - High Pass Example 2 8](#_Toc1301219)

[Figure 9 - Sharpening Example 1 9](#_Toc1301220)

[Figure 10 - Sharpening Example 2 9](#_Toc1301221)

[Figure 11 - Gaussian Example 1 10](#_Toc1301222)

[Figure 12 - Gaussian Example 2 11](#_Toc1301223)

[Figure 13 - First Derivative (Gradient) Example 1 12](#_Toc1301224)

[Figure 14 - First Derivative (Gradient) Example 2 13](#_Toc1301225)

[Figure 15 - Second Derivative (Laplacian) Example 1 14](#_Toc1301226)

[Figure 16 - Second Derivative (Laplacian) Example 2 14](#_Toc1301227)

[Figure 17 - Difference of Gaussians Example 1 15](#_Toc1301228)

[Figure 18 - Difference of Gaussians Example 2 16](#_Toc1301229)

# Summary

Assignment 2 consists of deriving kernels for eight (8) convolution filters and implementing the derived kernels in a C# application.

# Input

The following two (2) images were used as inputs to each convolution filter.



Figure 1 - Input 1: Black and White Husky



Figure 2 - Input 2: Landscape

# Convolution Filter Kernel Derivation

Kernels column length is equal to row length and the length must be odd, so that a center cell exists.

The sum of all values in a kernel must equal one (1).

Kernels are applied by multiplying each cell by a corresponding pixel cell, summing the results, then storing the resultant summation in location of the center pixel.

Kernels were derived for the following convolution filters.

## Identity

### Kernel Derivation

The Identity Kernel must produce an image identical to the original image, the center cell must be one (1) and all other cells must be zero (0):

### Output

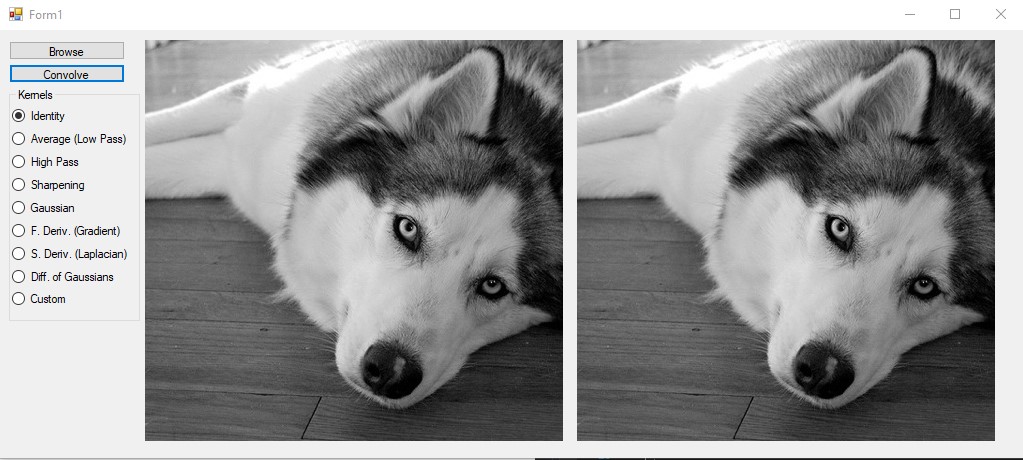


Figure - Identity Example 1



Figure - Identity Example 2

## Average (Low pass)

### Kernel Derivation

The Average (Low pass) kernel uses the neighboring pixels to determine the average pixel value. Therefore, all cells must be 1 divided by the sum of all values.

For a 3x3 kernel:

### Output



Figure - Average (Low Pass) Example 1

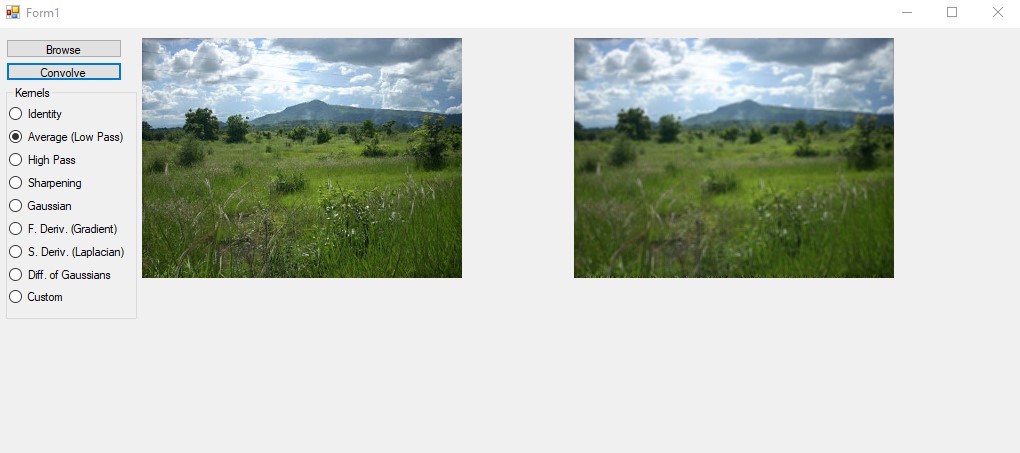


Figure - Average (Low Pass) Example 2

## High Pass

### Kernel Derivation

The High Pass Kernel is equivalent to an All Pass minus a Low Pass. The Identity Kernel is an All Pass; therefore, the High Pass Kernel would equal:

### Output

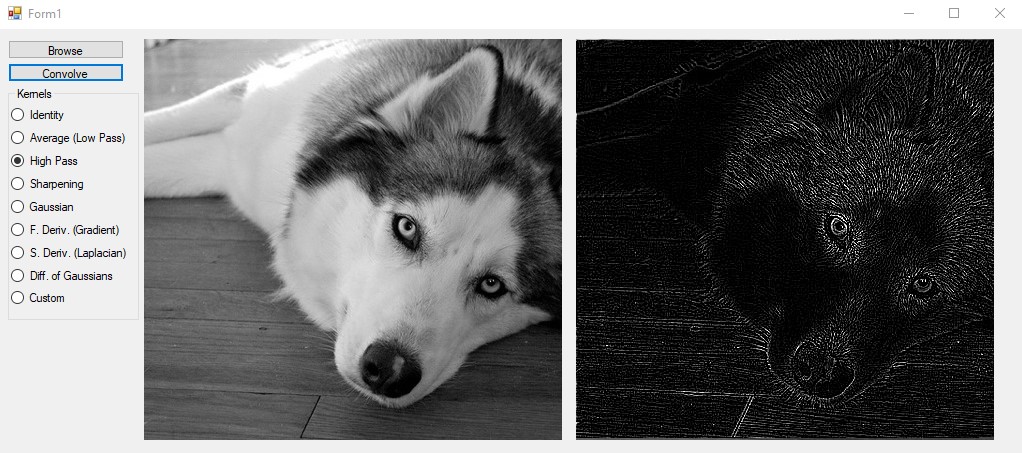


Figure - High Pass Example 1

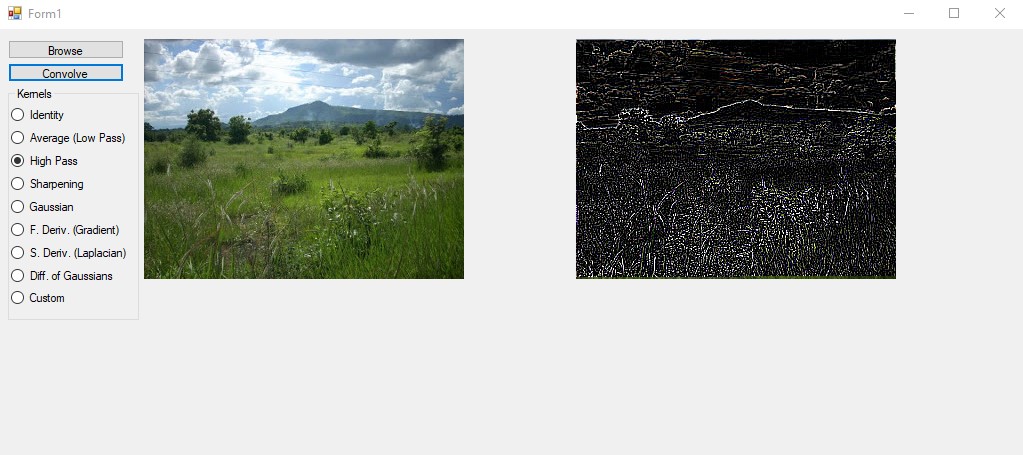


Figure - High Pass Example 2

## Sharpening

### Kernel Derivation

The Sharpening Kernel is equivalent to some percentage of the original image plus some percentage of the high pass filtered image:

### Output



Figure - Sharpening Example 1



Figure - Sharpening Example 2

## Gaussian

### Kernel Derivation

The Gaussian Kernel is equivalent to the two-dimensional Gaussian filter, represented by:

The following output uses a value of 0.5, which produces the kernel:

### Output



Figure - Gaussian Example 1

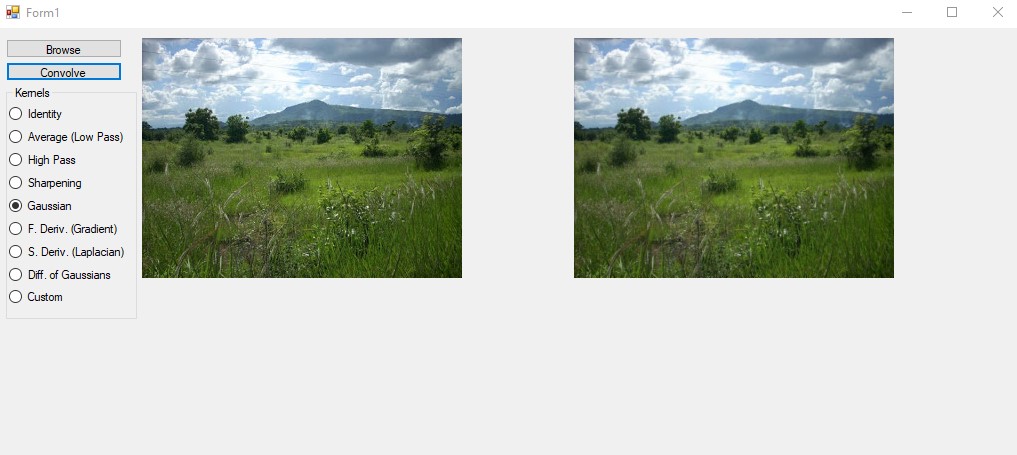


Figure - Gaussian Example 2

## First Derivative (Gradient)

### Kernel Derivation

The First Derivative (Gradient) Kernel uses Taylor Series to approximate the first derivative:

This shows only the Right and Left columns contribute to and only the Top and Bottom rows contribute to .

The Gradient Kernel X Component, therefore, must have a center column of zero and left and right columns that are equal magnitude, but opposite sign.

The Gradient Kernel Y Component, therefore, must have a center row of zero and top and bottom rows that are equal magnitude, but opposite sign.

The complete Gradient Kernel is

Alternatively, the component kernels may be:

### Output

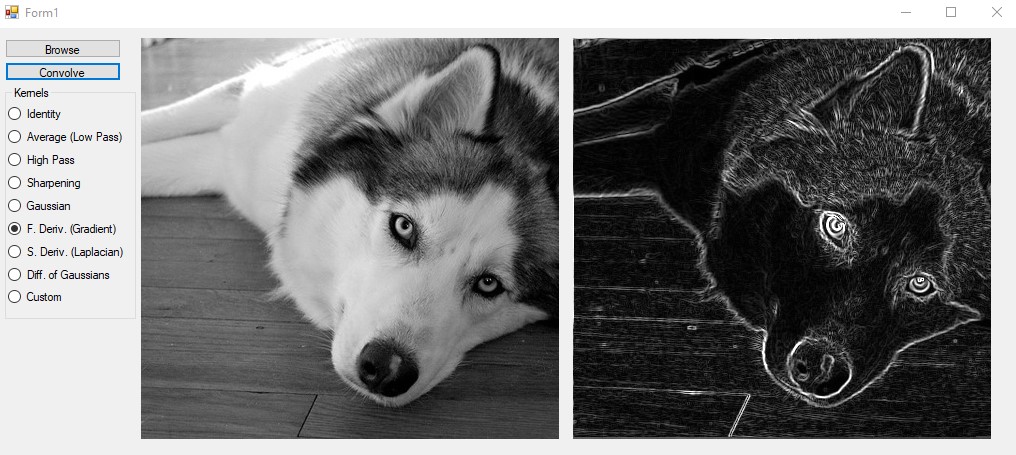


Figure - First Derivative (Gradient) Example 1

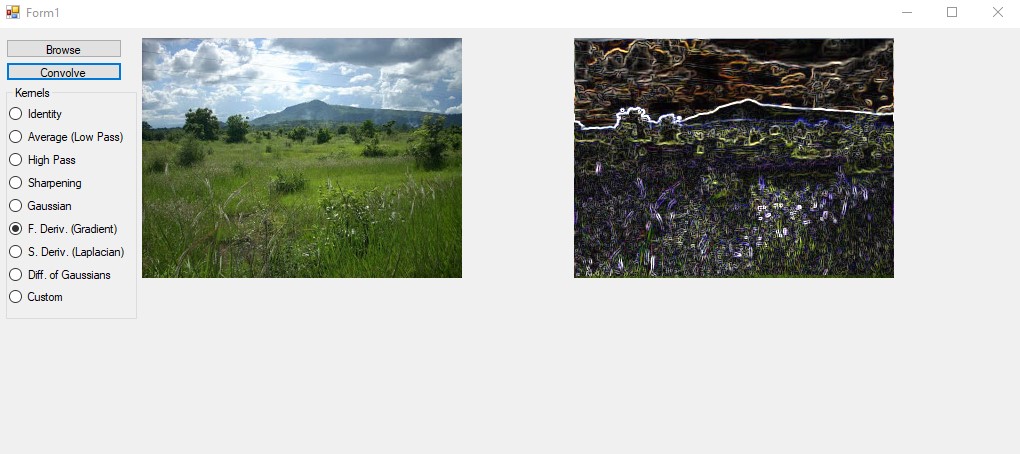


Figure - First Derivative (Gradient) Example 2

## Second Derivative (Laplacian)

### Kernel Derivation

The Second Derivative (Laplacian) Kernel uses Taylor Series to approximate the second derivative

Considering four (4) of the central point’s neighbors:

Considering all eight (8) of the central point’s neighbors:

### Output

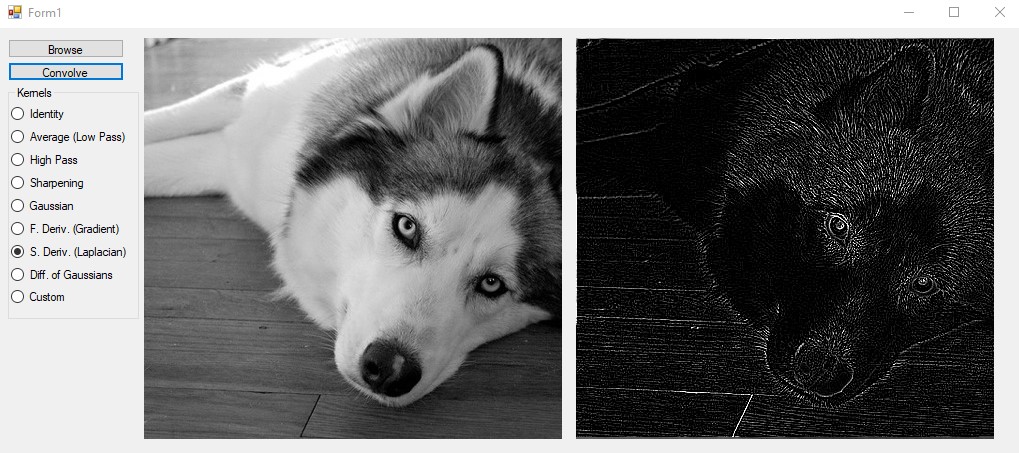


Figure - Second Derivative (Laplacian) Example 1

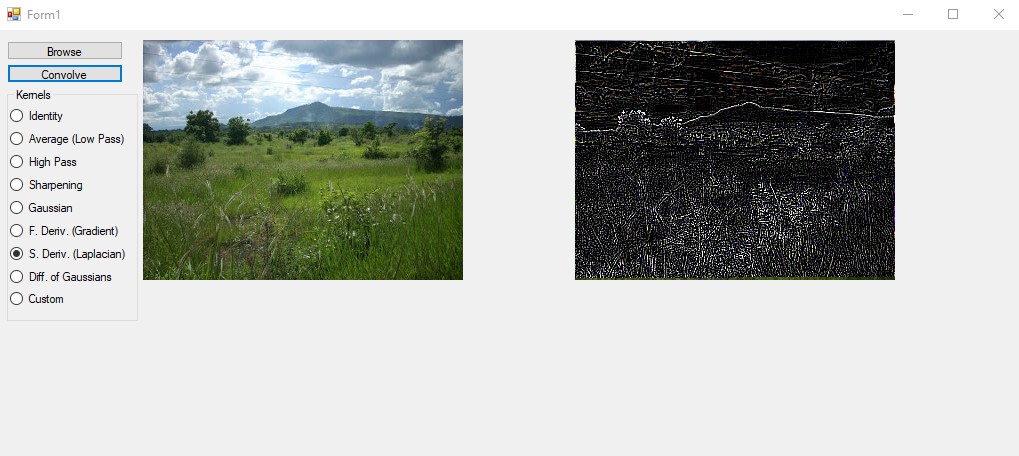


Figure - Second Derivative (Laplacian) Example 2

## Difference of Gaussians (σ­1=0.5, σ2=2.5)

### Kernel Derivation

The Difference of Gaussians Kernel is equivalent to the difference between two two-dimensional Gaussian filters, represented by:

A value of σ­1=0.5 produces and a value of σ2=2 produces :

The resulting normalized Difference of Gaussians Kernel is:

### Output



Figure - Difference of Gaussians Example 1



Figure - Difference of Gaussians Example 2