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

Contents

[Summary 4](#_Toc1280113)

[Input 4](#_Toc1280114)

[Convolution Filter Kernel Derivation 5](#_Toc1280115)

[Identity 5](#_Toc1280116)

[Kernel Derivation 5](#_Toc1280117)

[Output 5](#_Toc1280118)

[Average (Low pass) 5](#_Toc1280119)

[Kernel Derivation 5](#_Toc1280120)

[Output 5](#_Toc1280121)

[High Pass 5](#_Toc1280122)

[Kernel Derivation 5](#_Toc1280123)

[Output 6](#_Toc1280124)

[Sharpening 6](#_Toc1280125)

[Kernel Derivation 6](#_Toc1280126)

[Output 6](#_Toc1280127)

[Gaussian 6](#_Toc1280128)

[Kernel Derivation 6](#_Toc1280129)

[Output 6](#_Toc1280130)

[First Derivative (Gradient) 6](#_Toc1280131)

[Kernel Derivation 6](#_Toc1280132)

[Output 6](#_Toc1280133)

[Second Derivative (Laplacian) 6](#_Toc1280134)

[Kernel Derivation 6](#_Toc1280135)

[Output 6](#_Toc1280136)

[Difference of Gaussians (σ1=0.5, σ2=2.5) 6](#_Toc1280137)

[Kernel Derivation 6](#_Toc1280138)

[Output 6](#_Toc1280139)

[Figure 1 - Input 1: Black and White Husky 4](#_Toc1278428)

[Figure 2 - Input 2: Landscape 4](#_Toc1278429)

# 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 - Input 1: Black and White Husky



Figure - 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

## 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

## High Pass

### Kernel Derivation

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

### Output

## 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

## 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

## 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.

### Output

## Second Derivative (Laplacian)

### Kernel Derivation

### Output

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

### Kernel Derivation

### Output