Image Processing Using C/C++

1 - Introduction

In this project, you will be doing some image processing using C/C++. You will be required to implement the following image processing operations.

- *Negation*
- *Image Thresholding*
- *Histogram Stretching*
- *Median Filter*
- *Gaussian Filter*

Images used in this project will be provided in PGM (Portable Gray Map) because of their ease of reading and writing. Code is provided to handle reading and writing of the PGM files where all image processing procedures will be done on images that have been loaded in memory of your program. The provided functions for reading/writing are written in a C-style format since more complex C++ methods have yet to be presented in the course.

2 - Background

2.1 - PGM Files

A PGM file is in grayscale image format. Its file format provides encoding information, file comments, width/height, and max pixel value. Each file contains a header specifying information about the file followed by the image data itself. Below is a PGM file example (reference: http://netpbm.sourceforge.net/doc/pgm.html).

```
P2
# feep.pgm
24 3
15
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  3  3  3  3  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15 15 15 15  0
0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0 15  0
```

One of the easiest ways to open and view a PGM file is to use an online PGM viewer, such as https://bytes.usc.edu/~saty/tools/PGMViewer/viewer.html.

2.2 - Negation

Image negation makes lighter areas appear darker and darker areas appear lighter. This is done by subtracting each pixel value in the original image from a constant, which is the maximum value that can be represented using given bits. In this project, each pixel is represented by 8 bits, so it can have a value from 0 (black) to 255 (white). Therefore, the negation formula is:

output_pixel_value = 255 - original_pixel_value.

2.3 - Thresholding

Image thresholding is a technique used to segment an image into distinct regions based on pixel intensity values. The process involves converting a grayscale image into a binary image, where each pixel is classified as either black or white depending on whether its pixel value is above or below a certain threshold. In this project, a threshold value of 127 is used. Therefore, the output pixel value is set to 255 (white) if the value of an input pixel is greater than or equal to 128; it is set to 0 (black) otherwise.

2.4 - Histogram Stretching

Histogram stretching improves the contrast of an image by expanding the range of pixel values. It detects the minimum pixel value (min) and the maximum pixel value (max) of an original image and converts the pixel range from [min, max] to [0, 255]. The formula for histogram stretching is:

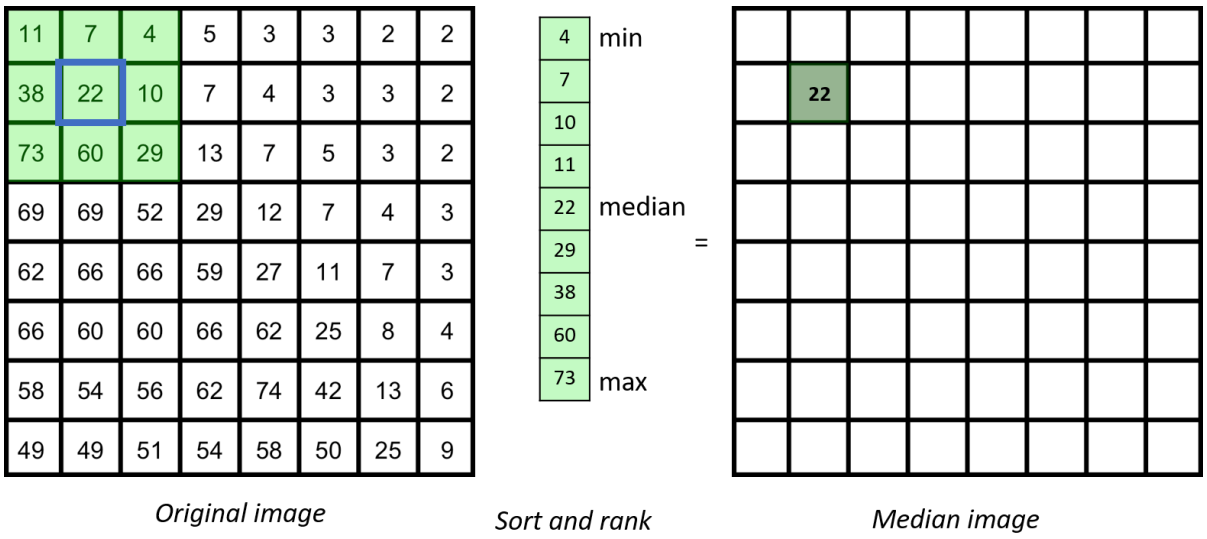
output_pixel_value = 255 x (original_pixel_value - min) / (max - min).

2.5 - Median Filter

The median filter is a non-linear digital filtering technique. It is commonly used for noise reduction and is highly effective at the removal of “salt and pepper” noise. The main concept of the median filter is to create a window of neighbors. Within a single window, the middle neighbor is replaced with the median value of all the neighbors in the window. The median filter is executed by sliding the window across the entire image.

For this project, you will be implementing a 3x3 median filter and applying it to an image. In particular, the filter window slides not only from left to right but also from top to bottom. **The edges of the resulting image (i.e., the first row, the last row, the first column, and the last column) are to be filled with the original values for simplicity.**

Figure 1 (image source: https://neubias.github.io/training-resources/median_filter/index.html) shows a 3x3 median filter example. A method for determining the median value in the window is to flatten the window into a 1D array, sort the values, and pick the middle value.

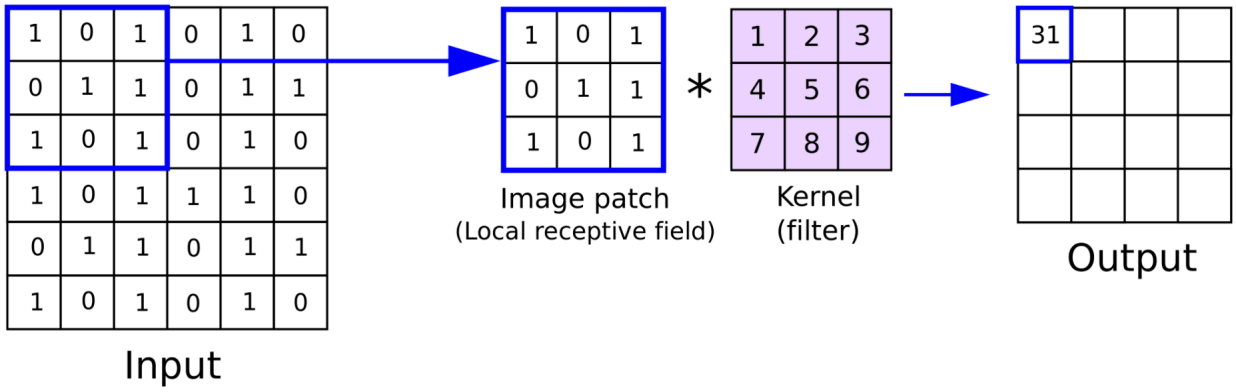


[**Figure 1**. Median filter example. The 3x3 window slides over the original image **by one pixel in a Z-order (from left to right; from top to bottom)**.]

2.6 - Gaussian Filter

The Gaussian filter is commonly used for signal processing. There are many uses of Gaussian filtering within the realm of image processing ranging from blurring an image to detecting the edges based on how much they contrast their surroundings depending on what filter, or distribution you use.

For this project, you will be implementing a 7x7 Gaussian filter and applying it to an image. The filter is applied using 2D convolution of the filter window with the input image as shown in Figure 2 (image source: https://www.superannotate.com/blog/guide-to-convolutional-neural-networks).



[**Figure 2**. 2D convolution example using a 3x3 filter (also called kernel) window. **This figure is just an example of convolution. In this project, the filter window size is 7x7, and the size of an input image and the size of an output image must be the same.**]

For this project, the edges of the resulting image (i.e., the first three rows, the last three rows, the first three columns, and the last three columns) are to be filled with the original values for simplicity. The 7x7 filter window slides not only from left to right but also from top to bottom. The filter window you will be using in this project is shown in Figure 3. This filter is already present in the gaussianFilter() function.

0	0	1	2	1	0	0
0	3	13	22	13	3	0
1	13	59	97	59	13	1
2	22	97	159	97	22	2
1	13	59	97	59	13	1
0	3	13	22	13	3	0
0	0	1	2	1	0	0

× 1/1003

[**Figure 3**. 7x7 Gaussian filter window used in this project. **You must divide the convolution result by 1003 "at last."**]

3 - Tasks

Implement the **imgNegation**(), **thresholding**(), **histogramStretching**(), **medianFilter**(), and **gaussianFilter**() functions in the ImageProcessing.cpp file.

- You can find information about the parameters and return data of these functions in the ImageProcessing.h file.
- **To match the precision of golden images with your resulting images, do not use the *double* and *float* data types (use the *int* data type).**
- PGM.cpp and PGM.h are used to read and write PGM files. There is nothing to do with these files.
- main.cpp receives the name of an input image, the type of operation (negation, thresholding, histogram stretching, median filter, or Gaussian filter), and the name of an output image. There is nothing to do with this file.

4 - Compile and Test

```
g++ -o main *.cpp *.h
./main
```

- ***Negation***
    - Inputs to the program

```
Enter Original File Name: input_images/cameraman.pgm
…
Enter Selection: 0
…
Enter Save File Name: output_images/negation.pgm
```

    - Original and resulting images

- ***Thresholding***
    - Inputs to the program

```
Enter Original File Name: input_images/cameraman.pgm
…
Enter Selection: 1
…
Enter Save File Name: output_images/thresholding.pgm
```

    - Original and resulting images

- ***Histogram Stretching***
    - Inputs to the program

```
Enter Original File Name: input_images/balloons.pgm
…
Enter Selection: 2
…
Enter Save File Name: output_images/stretching.pgm
```

    - Original and resulting images

- ***Median Filter***
    - Inputs to the program

```
Enter Original File Name: input_images/noisy.pgm
…
Enter Selection: 3
…
Enter Save File Name: output_images/median.pgm
```

    - Original and resulting images

- ***Gaussian Filter***
    - Inputs to the program

```
Enter Original File Name: input_images/cameraman.pgm
…
Enter Selection: 4
…
Enter Save File Name: output_images/gaussian.pgm
```

    - Original and resulting images

- ***Check all together***

```
chmod 777 run.sh
./run.sh
```

You must see the following messages:

```
IMAGE NEGATION - PASS
IMAGE THRESHOLDING - PASS
HISTOGRAM STRETCHING - PASS
MEDIAN FILTER - PASS
GAUSSIAN FILTER - PASS
```

5 - Submit

- Please upload your **ImageProcessing.cpp** file to **myCourses > Assignments > Project1**. If you have made modifications to other files or created new files, upload them together.
- Do not submit *.docx, *.pdf, *.txt, or *.zip file.

6 - Grading

- [90 pts] - Functions to be implemented
  - [10 pts] imgNegation()
  - [15 pts] thresholding()
  - [15 pts] histogramStretching()
  - [25 pts] medianFilter()
  - [25 pts] gaussianFilter()
- [10 pts] - Program execution
  - **The program execution portion of the grade is ALL or nothing.**