HO CHI MINH UNIVERSITY OF SCIENCE

COMPUTER VISION - LAB01

March 3, 2024

Author

Nguyen Viet Kim - 21127333

Contents

1 Introduction

This report provides an overview of the C++ program `main.cpp` and its functionalities. The program is designed to perform various image processing operations using OpenCV library.

2 File Management

The source code include these following folders:

```
21127333.zip
├── .vscode
│   ├── c_cpp_properties.json
│   ├── launch.json
│   └── tasks.json
└── Sources
    ├── bin
    │   └── 21127333.exec
    ├── data
    │   └── Lena.png
    ├── include
    │   ├── color.hpp
    │   ├── edge.hpp
    │   └── filter.hpp
    └── src
        ├── color.cpp
        ├── edge.hpp
        ├── filter.cpp
        └── main.cpp
```

- .vscode: Folder for configure the Visual Studio Code CC++ for running the program
- bin: Folder for saving execution files for the application
- include: Folder for storing required C++ header files
- src: Includes the C++ files.

3 Code Overview

3.1 Main.cpp

The *main()* function first checks the number of command-line arguments, ensuring proper usage. Next, it parses the arguments to identify the desired operation and input/output files. Based on the identified mode, it selects the corresponding image processing function.
The program accepts command-line arguments to specify the mode of operation and input/output files. Below are the supported modes along with their usage:

- **-rgb2gray <input_file> <output_file>**: Convert input image to grayscale
- **-brightness <input_file> <output_file> <brightness_value>**: Adjust brightness of input image
- **-contrast <input_file> <output_file> <contrast_value>**: Adjust contrast of input image
- **-med <input_file> <output_file> <kernel_size>**: Apply median filter to input image

- **-avg <input_file> <output_file> <kernel_size>**: Apply average filter to input image
- **-gau <input_file> <output_file> <kernel_size>**: Apply Gaussian filter to input image
- **-sobel <input_file> <output_file>**: Apply Sobel edge detection to input image with kernel size of 3x3
- **-laplace <input_file> <output_file>**: Apply Laplace edge detection to input image with kernel size of 3x3

3.2 Color.cpp

3.2.1 Convert RGB To Grayscale Image

The *rgb2gray()* function converts a color image to grayscale using the average of RGB channels for each pixel. The function iterates through each pixel of the input image, calculates the average intensity of the RGB channels, and assigns it to the corresponding pixel in the grayscale image.

$$gray\_intensity = \frac{R+G+B}{3}$$

3.2.2 Adjust Brightness of The Image

The *brightness()* function adjusts the brightness of an image by adding a constant value (c) to each pixel's intensity. The function it iterates through each pixel of the input image, adds the constant value to each channel's intensity (clipped to the range [0, 255]), and assigns the result to the corresponding pixel in the new image.

$$new\_channel = channel + brightness\_constant$$

3.2.3 Adjust Contrast of The Image

The `contrast` function adjusts the contrast of an image by multiplying each pixel's intensity by a constant value (`c`). It follows a similar process to the `brightness` function, but instead of adding a constant value, it multiplies each intensity value by the specified constant. The result is then clipped to the range [0, 255] and assigned to the corresponding pixel in the new image.

$$new\_channel = channel \times brightness\_constant$$

3.3 Edge.cpp

3.3.1 Sobel Edge Detection

The Sobel function computes the gradient magnitude of an image to detect edges. The Sobel operator consists of two 3x3 convolution kernels, G_x and G_y, for detecting edges in the horizontal and vertical directions, respectively.
This function applies the Sobel edge detection algorithm to the input grayscale image specified by user. It computes the gradient magnitude of the image and writes the result to the output file specified by `outputFile`.
The gradient magnitude ∇f of an image f at each pixel (x, y) is calculated using the formula:

$$\nabla f = \sqrt{(G_x * f)^2 + (G_y * f)^2}$$

The Sobel kernels are defined as follows:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

3.3.2 Laplace Edge Detection

The Laplace function computes the second derivative of an image intensity.
This function applies the Laplace edge detection algorithm to the input grayscale image specified by user. It computes the Laplacian of the image and writes the result to the result output image.
The Laplacian $\nabla^2 f$ of an image f at each pixel (x, y) is calculated using the formula:

$$\nabla^2 f = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+j) \times Laplacian\_Kernel(i, j)$$

The Laplacian kernel is defined as:

$$Laplacian\_Kernel = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

3.4 Filter.cpp

3.4.1 Median Filter

The median filter is a nonlinear digital filtering technique that is often used to remove noise from an image. It replaces each pixel's value with the median value of the pixel values in its neighborhood.
For each pixel in the image, the median filter calculates the median value of the pixel values within a kernel neighborhood. The resulting pixel value is then set to the median value.

$$\text{Output}(x, y) = \text{median}(\{I(x+i, y+j) \mid i, j \text{ within the neighborhood defined by the kernel}\})$$

Where:

- $\text{Output}(x, y)$ is the value of the pixel at coordinates (x, y) in the output image.
- $\text{median}(\{I(x+i, y+j) \mid i, j \text{ within the neighborhood defined by the kernel}\})$ represents the median value of the set of pixel intensities within the neighborhood.
- $I(x+i, y+j)$ is the value of the pixel at coordinates $(x+i, y+j)$ in the input image.
- $N \times N$ is the size of the kernel (width and height of the neighborhood).

3.4.2 Average Filter

The average filter, also known as the mean filter, is a linear image filtering technique used to reduce noise and smooth images. It replaces each pixel's value with the average value of the pixel values in its neighborhood.
For each pixel in the image, the average filter calculates the average value of the pixel values within a kernel neighborhood. The resulting pixel value is then set to the average value:

$$\text{Output}(x, y) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \text{Input}(x+i, y+j)$$

Where:

- Output(x, y) is the value of the pixel at coordinates (x, y) in the output image.
- Input$(x + i, y + j)$ is the value of the pixel at coordinates $(x + i, y + j)$ in the input image.
- N is the size of the kernel (width and height).

3.4.3 Gaussian Filter

The Gaussian filter is a linear image filtering technique used to blur images and reduce detail and noise. It applies a convolution operation with a Gaussian kernel to the input image.
The Gaussian filter convolves the input image with a Gaussian kernel, which is a two-dimensional Gaussian distribution. The kernel is defined by its standard deviation (σ) and size ($n \times n$). The Gaussian function is given by:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The resulting pixel value is calculated as the weighted sum of the pixel values in the kernel neighborhood, with weights determined by the Gaussian function.

4 Program Result

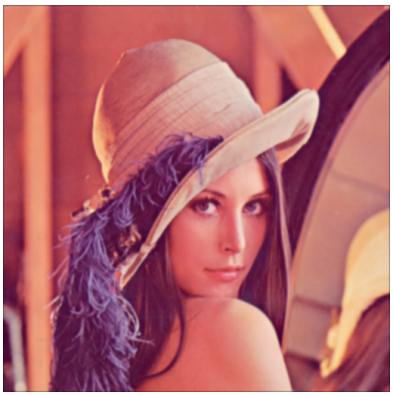
Original image

Brightness 50

Contrast 3

Average filter 5x5

Median filter 5x5



Gaussian filter 5x5


Convert RGB to Grayscale


Sobel edge detection


Laplace edge detection

5 User guide

Unzip the file 21127333.zip. Once ready, open the folder in the Visual Studio Code. Press *Ctrl* + ‘ to open the terminal from the VSCode. We need to locate to the bin directory and execute the program by running these commands:

```
$ cd /path/to/directory/bin
$ ./21127333 <mode> <file_input> <file_output> <factor>
```

User needs to select option for image processing mode. User can select the image for image processing and the output file name for image after the changes. The factor is optional depending on the number arguments required to execute the option. These arguments are described in the Code Overview section.