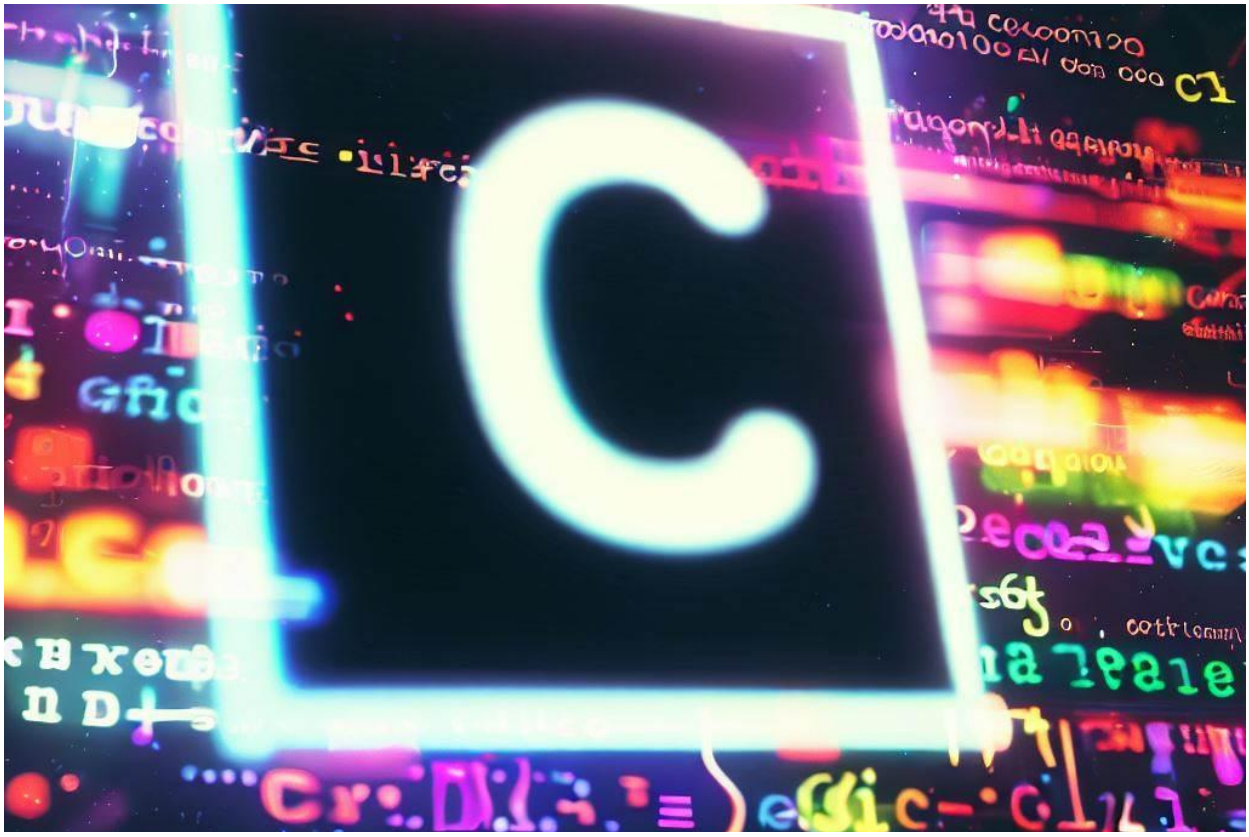


+

IMAGE FILTERING PROGRAM

Implemented by C programming language

1402

—

Negar ChahardahCherick & Setayesh Saeidi
Shiraz University

Overview

The project aims to develop a program that allows users to upload an image from three ways including :

1. From the address of the image in PC
2. Form URL of the image
3. From user's email and giving the user's UID

And then apply various filters to it, and save the filtered image using the C programming language; the saving formats include :1.png 2.jpeg 3.bmp.

The program will provide a user-friendly interface for selecting and applying filters, changing the visual appearance of the image.

Inputs:

The program will support user image input through various methods, such as browsing the local file system to select an image file or accepting image URL directly from the user or getting it from user's gmail and downloading it to PC. The chosen input method will depend on user interface preferences. By the way the supported formats for images are bit map , png and jpeg.

Image filtering:

Once the user provides the input image, the program will offer a range of filter options to apply. Filters include , color overlay ,grain ,brightness , posterize ,solarize , mirror , darkness , glitch , grayscale conversion, sepia tone, blur, sharpen and edge-detection. The user can select one filter to apply to the image.

Image processing:

To apply the selected filters, the program will utilize image processing techniques and algorithms. These algorithms will operate on the pixel data of the image, manipulating color channels and intensities according to the selected filter. The program will ensure efficient memory handling and error handling to prevent crashes or memory leaks.

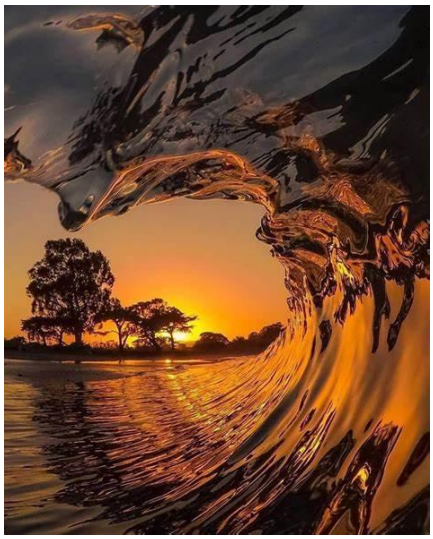
Filter customization:

The program may incorporate a feature for users to customize filter parameters, such as adjusting brightness, contrast, saturation, or specific filter settings. This functionality adds flexibility and allows users to achieve their desired image transformations.

Summary about how each filter works:

(with picture examples)

Sample images:

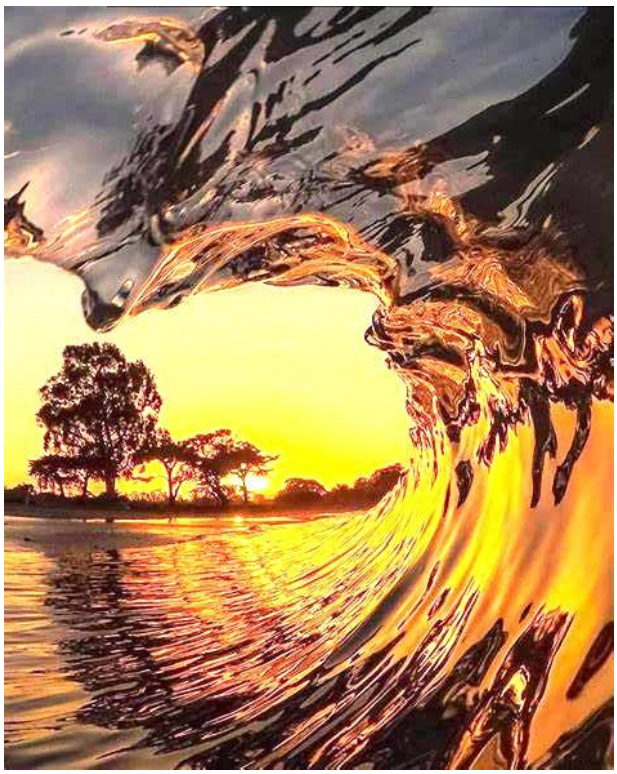


grayscale:

Getting average of RGB values then assigning it to each of them.

Brightness:

By multiplying the value of the pixel channel by the given amount we increase the intensity of the channel then assigning the new value to the pixel channel in the image data array.
The range of given amount to make it brighter must be between (1-10)
Increasing the intensity of the channels makes the picture brighter!



Darken:

By multiplying the value of the pixel channel by a given amount we reduce the intensity of the channel then assigning the new value to the pixel channel in the image data array.
The range of given amount to make it brighter must be between (0.0-1.0)
Reducing the intensity of the channels makes the picture darker!

Grain:

Adds random values to pixels to create a noisy appearance.
Noise is generated by random value between (0-intensity) then its value is added to the original value of pixels.

(pictures above are generated from different grain amount)

Blur:

(Box Blur)

Average value of pixels in the neighborhood around each pixel.
Size and shape of neighborhood is determined by blur kernel.(kernel is filled with ones and it's size is 2* radius + 1).

Kernel : matrix of weights , multiplied by pixel value and summed up to get a blurred value.
Radius (positive int) => Controls the size of the blur kernel.
normalize matrix => scale values the way so range of row or column value is between (0 -1)

Function: allocates memory for a temporary array to store blurred image data and for a blur kernel. The math done by the kernel creates a uniform kernel that gives equal weight to all pixels within the radius.
Then assigns blur value to pixels in the temporary array and copies the temporary array back to the original image.

Sepia tone:

Adds a warm brown tone (using formula) applied to each RGB value.

Solarize:

solarize effect is a photo effect that creates a reversed or negative look for some parts of an image. It manipulates the pixel values by inverting them above a certain threshold. This creates a contrast between the normal and inverted parts of the image.

Mirror:

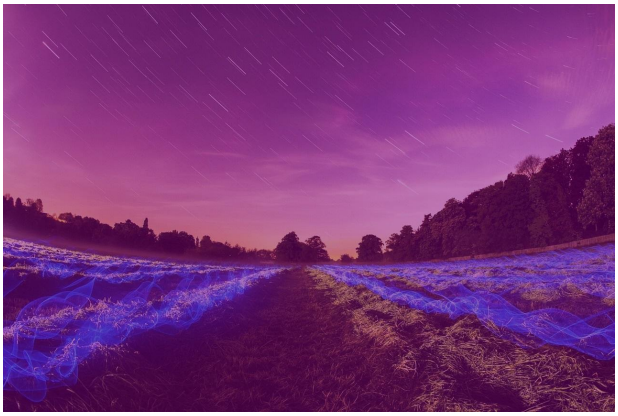
creates a mirror image of the input image by reversing the order of the pixels along a specified axis.A mirror filter is implemented by swapping the pixel values of the input image according to the desired axis of reflection.

(x,y) => (width - x,y)

(x,y) => (x,height - y)

Color overlay:

This filter changes the image by blending it with a single color.
We have an alpha variable which controls the visibility of the original image, and how much of the overlay color is applied.
Alpha range is between (0.0 - 1.0) => The higher the alpha, the more overlay color dominates.
The lower the alpha, the more the original image shows through.



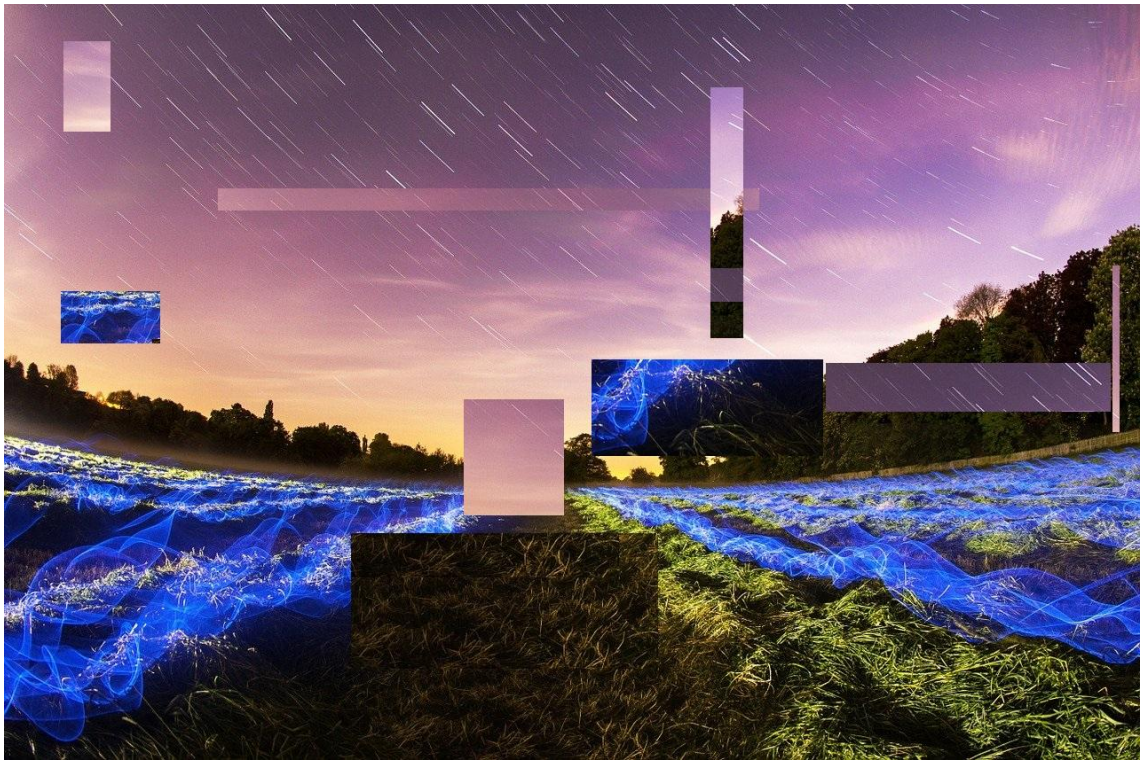
Glitch:

Glitch effect creates a distorted or corrupted appearance of an image by randomly copying and pasting parts of the image to different locations. The effect can create a sense of chaos, error, or malfunction in the image.
The function loops 10 times.

- It chooses a random source rectangle within the image by generating random coordinates for its top-left corner (src_x, src_y) and its width (src_w) and height (src_h).

- It chooses a random destination rectangle within the image by generating random coordinates for its top-left corner (dst_x, dst_y) and its width (dst_w) and height (dst_h).

- It copies and pastes the source rectangle to the destination rectangle by looping through every pixel in both rectangles and copying the color values from the source pixel to the destination pixel.

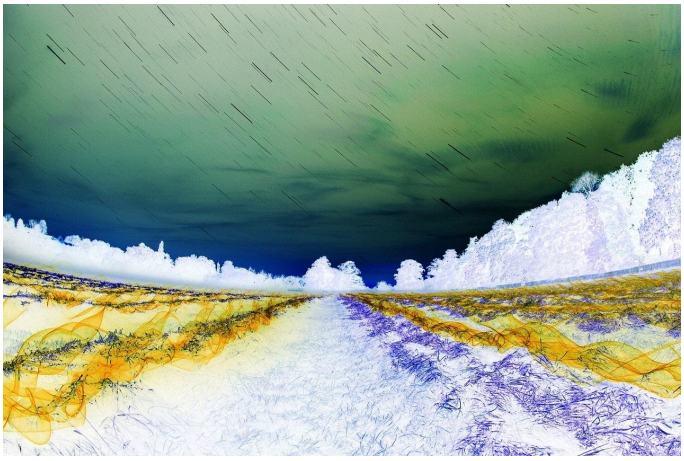


Posterized:

A posterize filter reduces the number of colors in an image, creating a stylized effect.
takes level to posterize image (smaller the number => the fewer colors in the image)
function calculates the step size for each color channel based on the desired number of levels.
It loops through every pixel in the image and every color channel in each pixel and applies a filter to each color channel.
For each pixel value, it rounds it down to the nearest step.

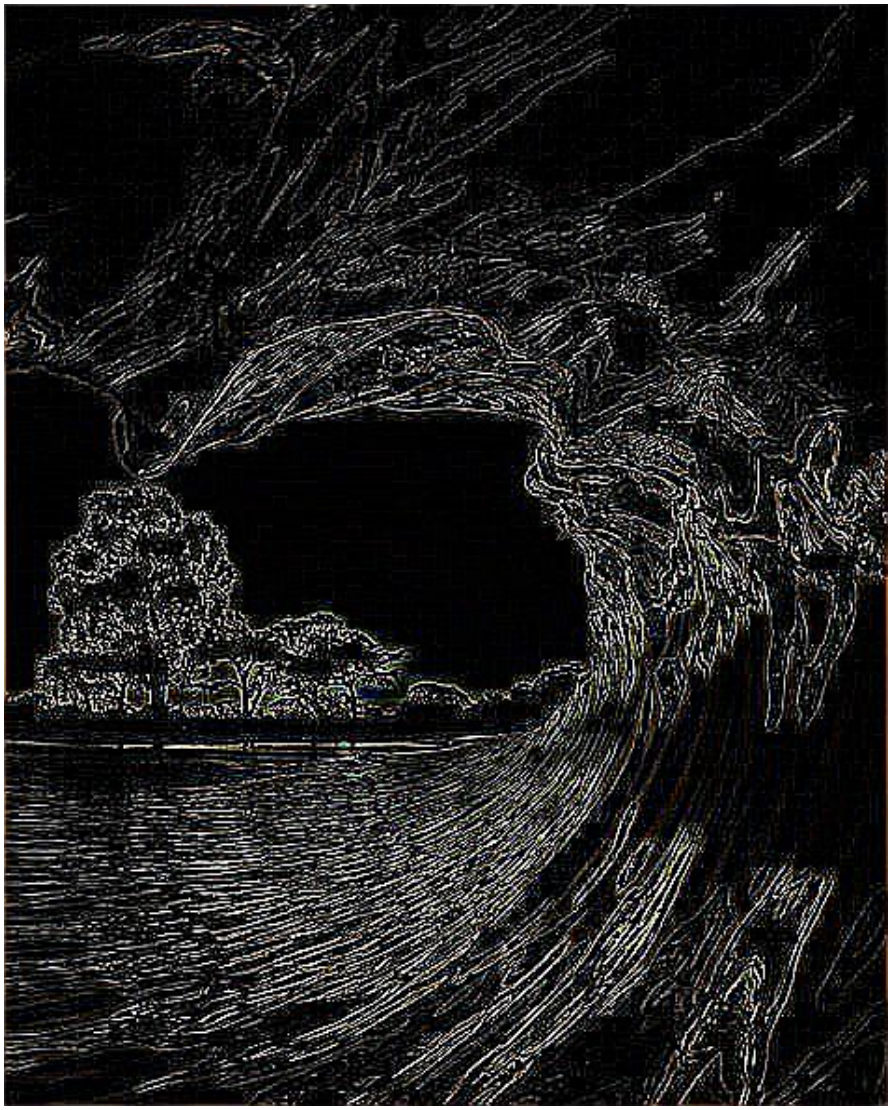
Invert:

Invert colors of image by subtracting pixel values from 255.



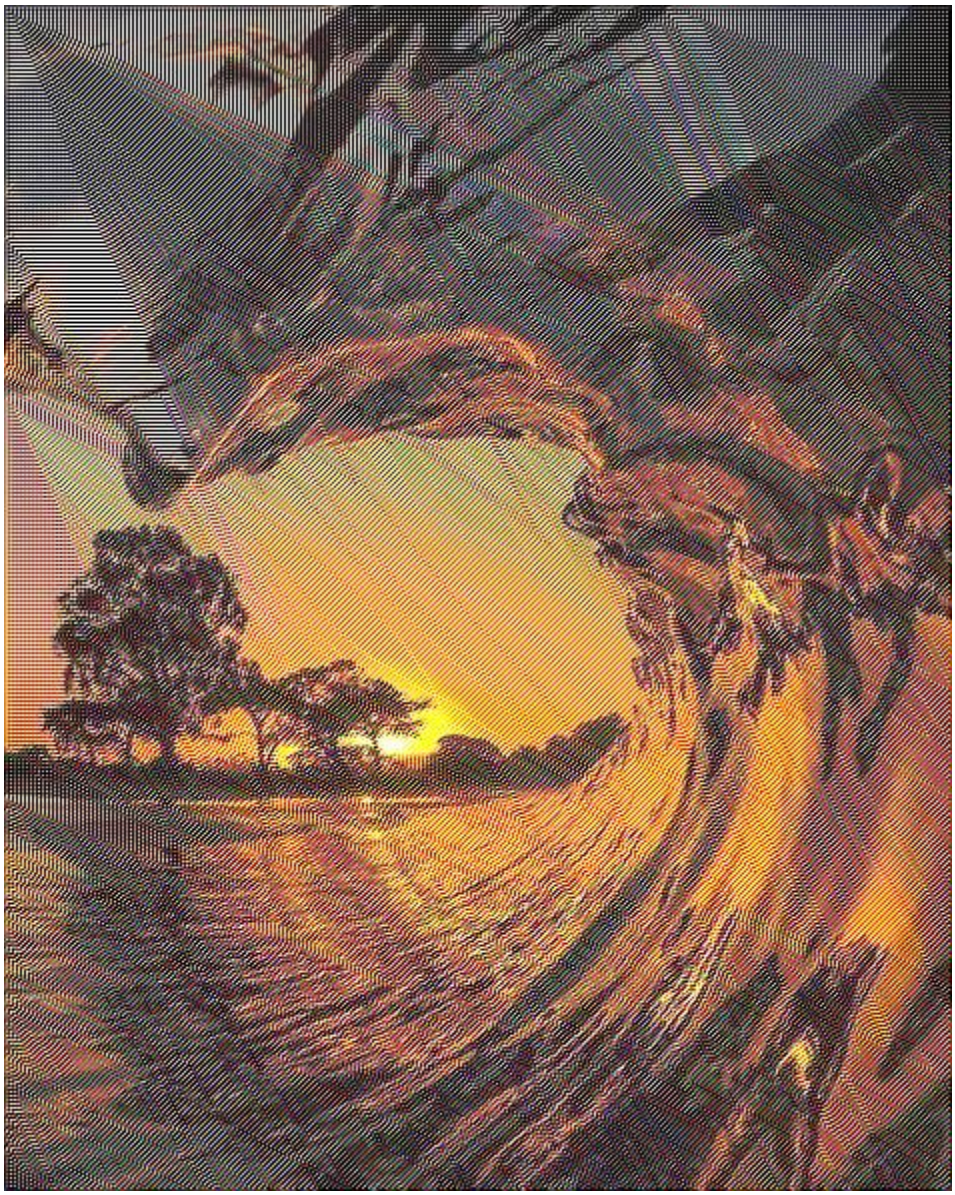
Edge-detection:

This filter highlights the edge or boundaries of objects in image.
function creates a temporary memory copy of image data using malloc and memcpy to avoid modifying
the original image data while editing.
The Kernel 3x3 has high value for edges and low value For smooth regions.
It loops through every pixel in the image, except border pixels then applies the filter to the color channel. multiplies each pixel and its neighbors by corresponding kernel weight and adds them up then it gives sum that represents intensity of the edge at that pixel , then clamps the sum range between (0-255) then frees temporary image data.



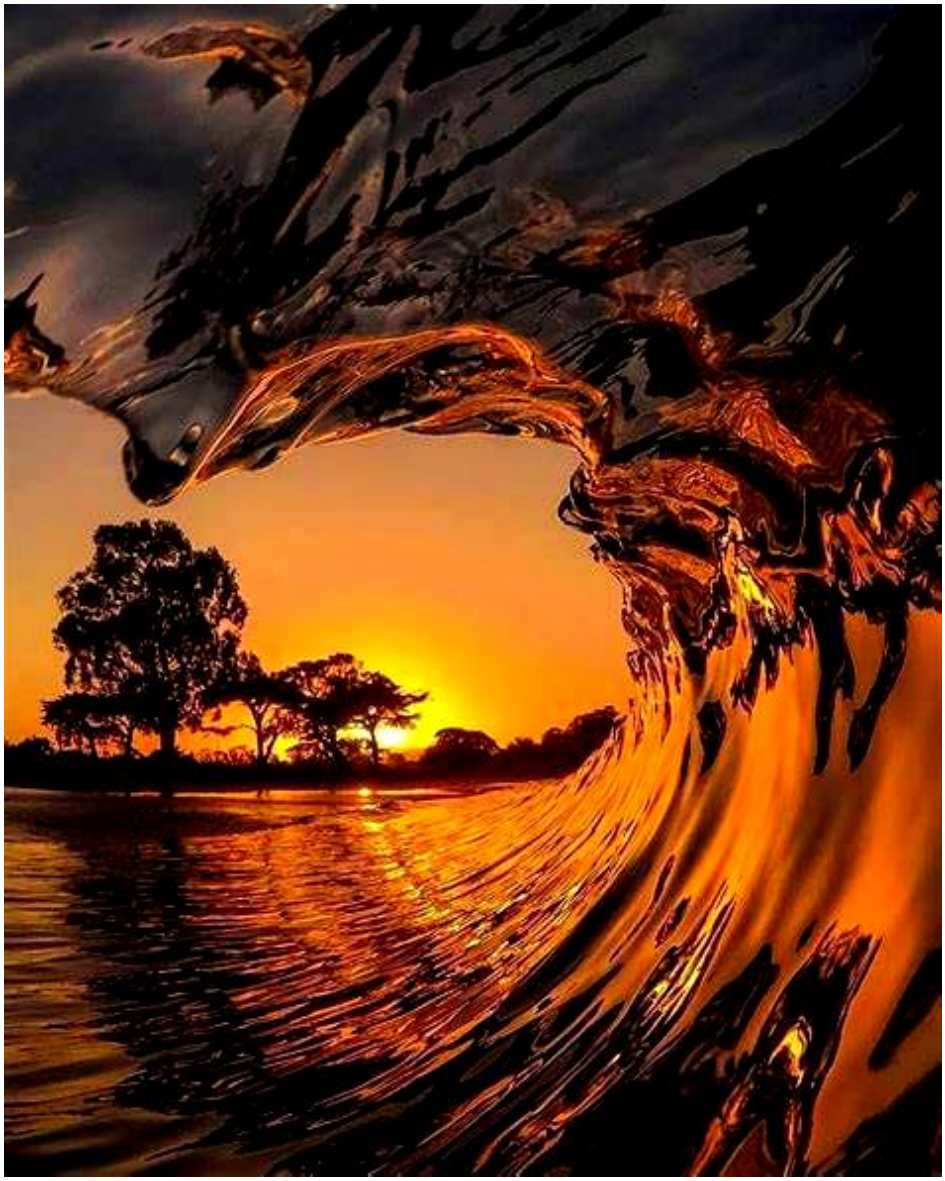
Emboss:

This filter creates a realistic embossed effect on the image, making it look like it has been stamped or carved on a paper or metal surface. An emboss filter replaces each pixel of an image by a highlight or a shadow, depending on the light/dark boundaries on the original image. Low contrast areas are replaced by a gray background. It uses a specific kernel or matrix that defines the direction and strength of the embossing effect.



Contrast:

Contrast in image processing refers to the difference in luminance or color that makes an object distinguishable from other objects within the same field of view. It is a measure of the difference between the maximum and minimum pixel intensities in an image.Contrast effect is the process of changing the contrast of an image to improve its appearance or highlight some features.



Saving image :

After the user applies the desired filters and achieves the desired image appearance, the program will provide an option to save the filtered image. The program will prompt the user for a file name and location to save the image. It will utilize appropriate file handling techniques to ensure a seamless saving process without overwriting existing files.

Error handling :

To ensure robustness and stability, the program will implement error handling mechanisms. It will validate user inputs, such as file formats, to prevent crashes or erroneous operations. The program will display clear error messages to guide users in rectifying any issues encountered during the process.

Conclusion :

By developing this image filtering and saving program in C, users will have the ability to enhance their images with a variety of filters. The project aims to provide an efficient and user-friendly solution for applying filters, previewing changes, and saving the filtered images, contributing to an enhanced visual experience for users.

Thank you for your time, have a good day.