**Project 1 Report**

**Name: Xianhe Zhang**

**Email:** [**zhang.xianh@northeastern.edu**](mailto:zhang.xianh@northeastern.edu)

**[Description]**

This project comprises two main files: **imgDisplay** and **vidDisplay**.

**imgDisplay** is primarily tasked with reading and displaying images. On the other hand, **vidDisplay** is slightly more intricate, handling live video feeds. With this, users can apply different filters to the video feed by pressing specific keys. The available filters include two additional greyscale images, a 5x5 Gaussian filter, a 3x3 Sobel X filter, a 3x3 Sobel Y filter, a function to generate gradient magnitude images, another that blurs and quantizes a color image, and a live video carton function**. Notably, for this project, pressing these keys won't alter the live video in real-time; instead, users will observe varied effects applied to the same images.**

**[Required Images]**

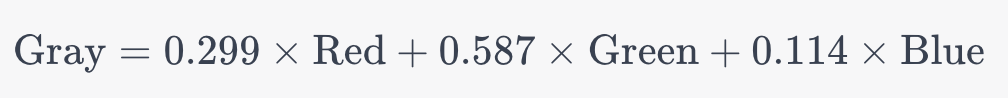
Original Image

A puppy sitting in the grass

Description automatically generated

Image 1 – by using cv::cvtColor.

Normally, this is typically computed using the following weighted sum of the channels:



**A dog lying in the grass

Description automatically generated**

Image 2 - customized greyscale image.

Now, all three channels (B, G, R) now contain the values of the original blue channel.

**A dog lying in the grass

Description automatically generated**

Image 3 - blurred image

A puppy sitting in the grass

Description automatically generated

Image 4 - the gradient magnitude image

A dog sitting in the grass

Description automatically generated

Image 5 - blurred/quantized image

A dog sitting in the grass

Description automatically generated

Image 6 – cartoonization image.

This is achieved exactly by following steps on the project description.

A dog sitting in the grass

Description automatically generated

Image 7 – reverse image. This is implemented by simply inversion operation.

A dog sitting in the grass

Description automatically generated

**[Extension Images]**

As for the extension, I choose to add Laplacian filter.

It is a 2D filter used in image processing, primarily for edge detection and enhancing edges in images. It is a derivative filter that calculates the second derivative of an image, highlighting regions of rapid intensity change.

**A black and white image of a dog

Description automatically generated**

**[Take-away]**

This project primarily taught me how to obtain video data and apply various effects to it. Furthermore, it shed light on the collaboration of files within a C++ project, showcasing the interdependencies and integrations essential for the smooth operation of a program. In addition, the significance of a makefile became apparent. A makefile streamlines the build process, automating the compilation and linking of multiple source files. It not only ensures consistency in the build process but also simplifies complex project structures, allowing developers to efficiently handle and maintain large codebases.