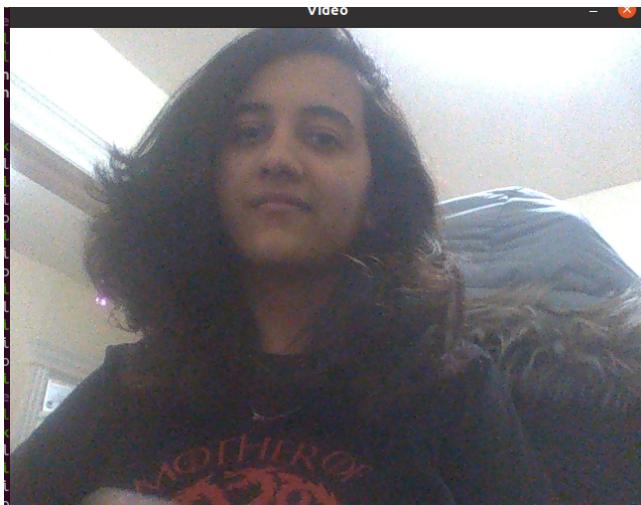


# Real Time Filtering :

**Description :** This project consists of many sub-tasks. Each task consists of a filter which is implemented using C++ and OpenCV. The project's main purpose is to implement filters like Grayscale conversion, Gaussian blur, Sobel X, Sobel Y, and their magnitude, blurred and quantize filter, and cartoonization. User can also save any effect whenever he/she presses the 's' key. The project gets executed on Live Video. The user has a set of keys to switch from one filter to another. Every filter is applied to a normal image without any filtering effect.

As part of an extension, I have implemented: mean Blur filter, effects like - flipping live video, upside down execution of Live video, Pencil Sketch effect, Image sharpening, and edge detection filter - Prewitt X, Prewitt Y, and Prewitt magnitude. I wrote vidDisplay.cpp, filter.cpp, and filter.h files to apply filters on Live Video. imgDisplay.cpp file reads saved image and displays it on a window.

**Input image :**



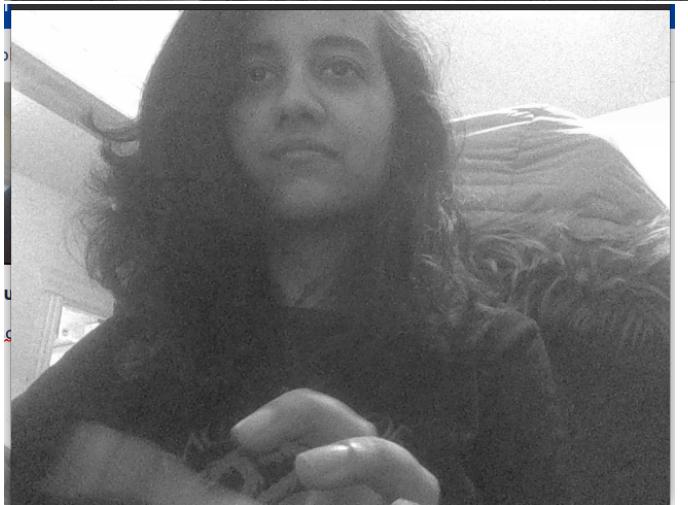
**Input Image for Extension:** Prewitt and sketch effect



**Output Images :** Implemented Prewitt X, Y and Magnitude and Sketch on different image for better output. This is also executed in Live video.

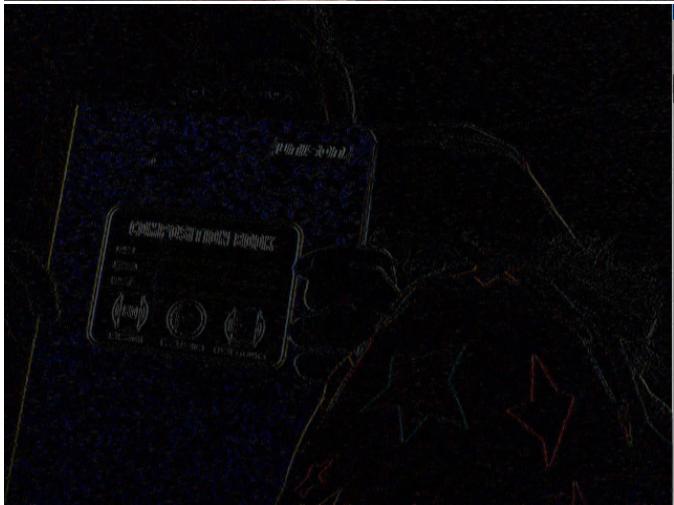
**Grayscale(cvtColor) :**

**Alternative Grayscale :**



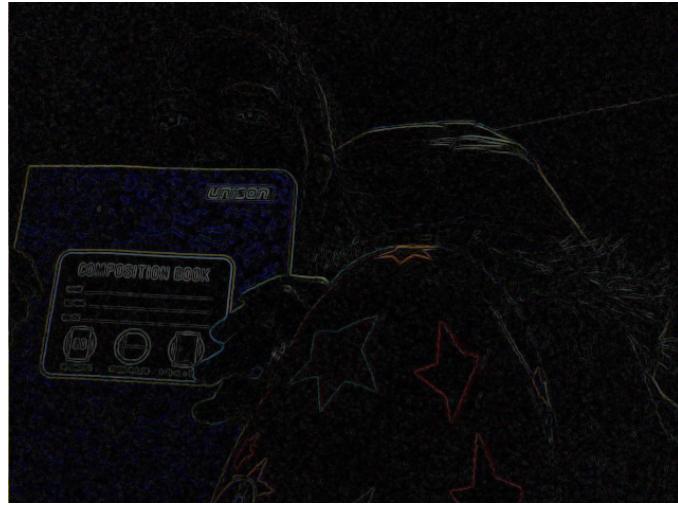
Gaussian blur as separable 1x5 filters :

Sobel X :



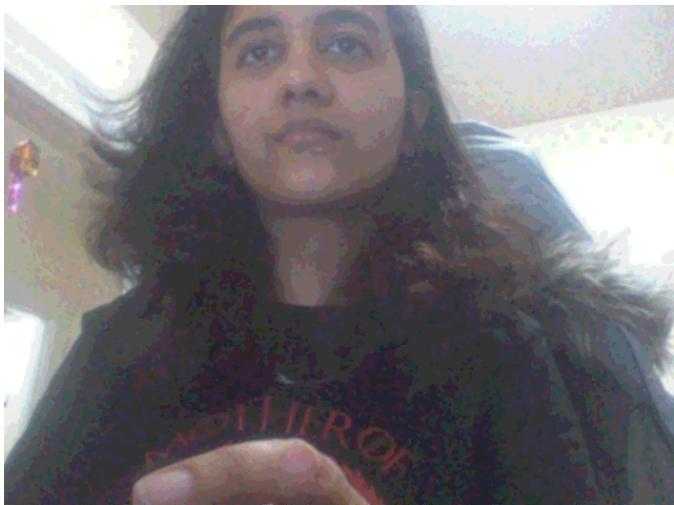
Sobel Y :

Gradient Magnitude :



Blur and Quantize :

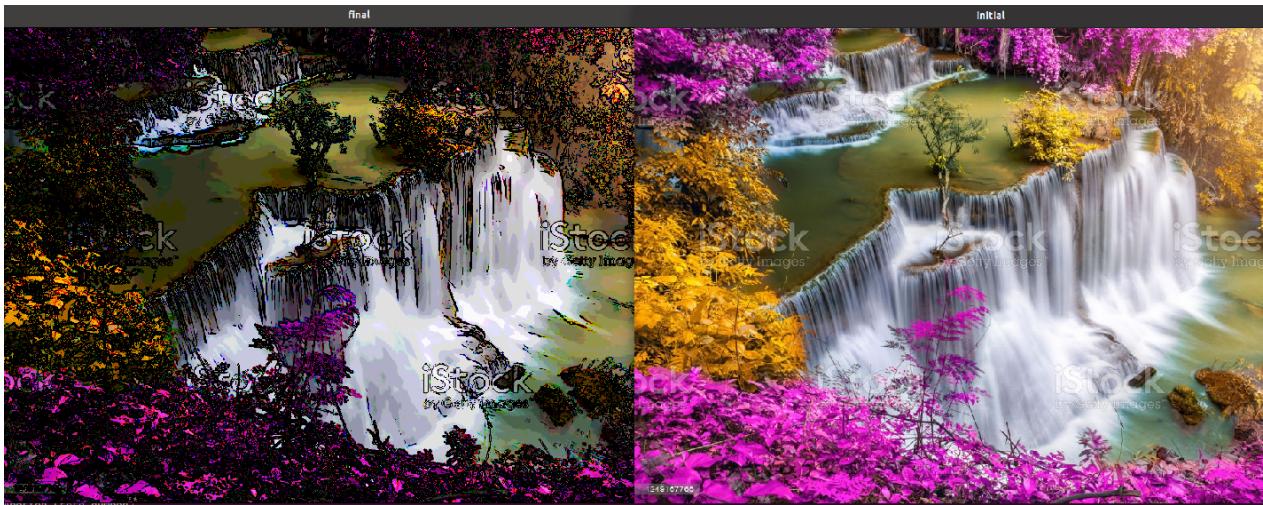
Cartoonization :



Blur and Quantize:



cartoonization on Image:



Brightness :



Negative:



**Extension:** Implemented different effects on Live video and still images where user can saved modified images any time by pressing 's' key. Also, implemented additional blur and edge detection filter from scarch.

Average Blur :

Image sharpening :

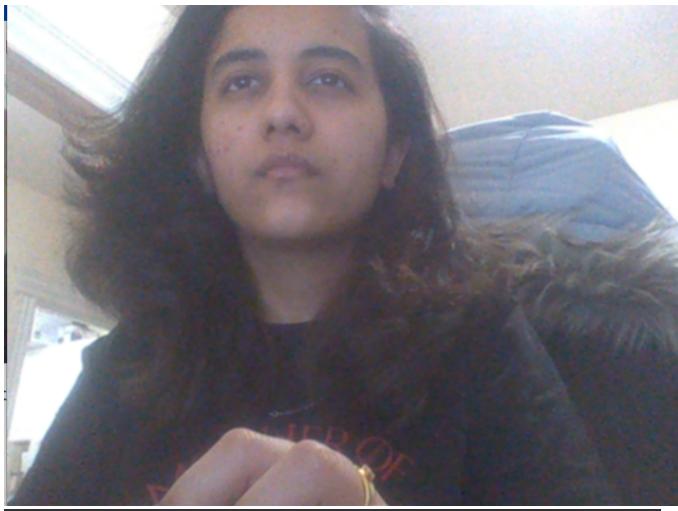
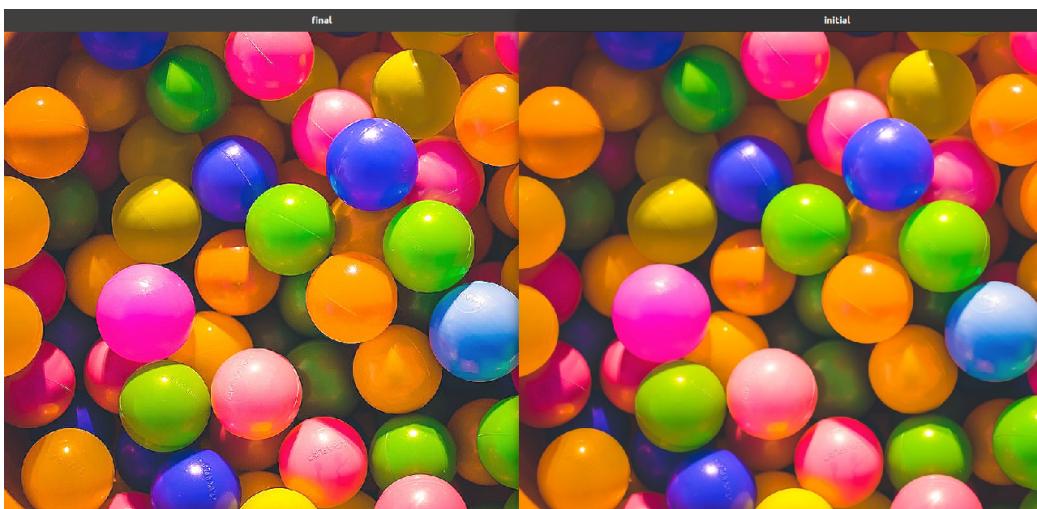


Image Sharpening on Image :



Sketch :

Upside Down Image :



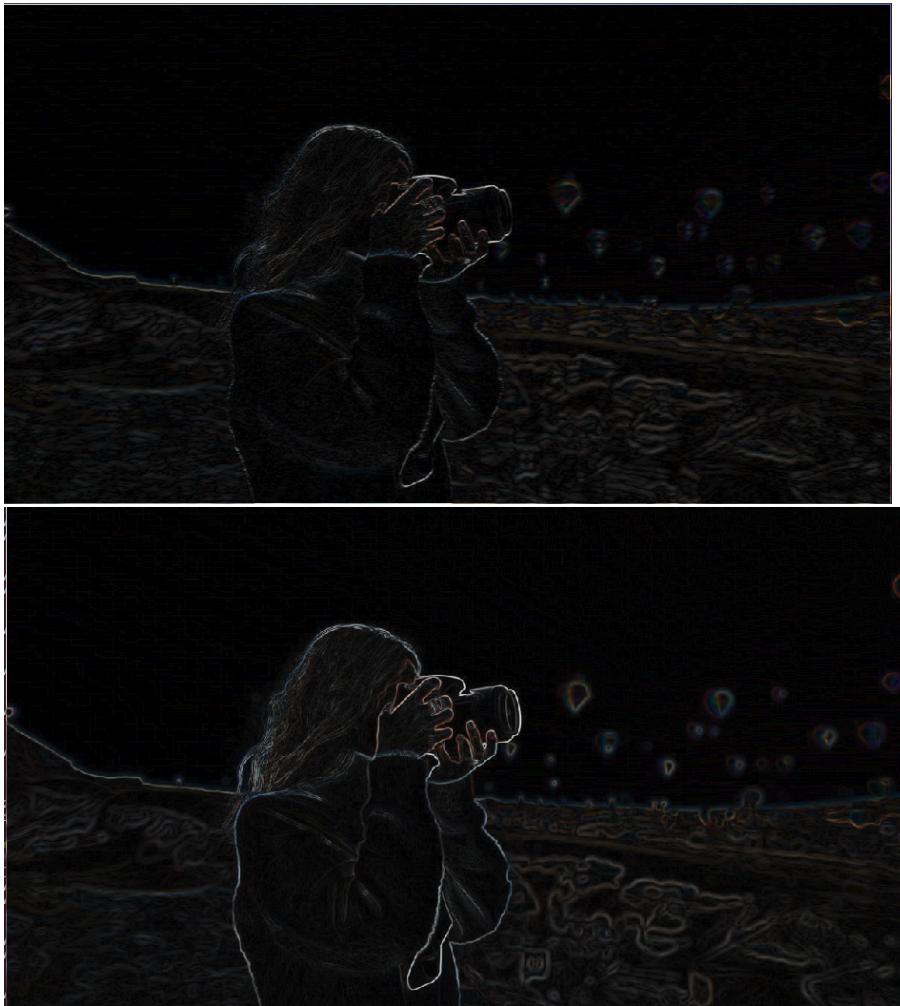
Flip (along X axis) :

Prewitt X :



Prewitt Y :

Prewitt Magnitude :



You can see a screen recording of Live Video : [https://github.com/ravina01/OpenCV\\_Project1/blob/fa62cef72897116e6d74a13f5be4b25e074d2dd4/OpenCV\\_Project1.mp4](https://github.com/ravina01/OpenCV_Project1/blob/fa62cef72897116e6d74a13f5be4b25e074d2dd4/OpenCV_Project1.mp4)

**Reflection :**

I have learned a method to extract data from images, to use filters from scratch through reading each pixel and performing mathematical operations on it, also eliminate high-frequency noises from images. I strongly believe that this project made me ready for future computer vision and deep learning projects as the hidden layers mainly consist of filtered output.

**References:**

I have seen some videos on c++ and OpenCV. I used Stack overflow to resolve bugs in the code. Also, Wikipedia pages to see the process to perform filters.