

CS5330 - Pattern Recognition and Computer Vision
(sec-01 spring 2023)

Northeastern University

Project - 1

Sreehari Premkumar



Description of the Project

Project 1 of Pattern Recognition and Computer Vision, has the usage of OpenCV along with C++ to create an application having different Filters such as gaussian blurring, edge detection etc by applying consecutive convolution of separable kernels with the image. The project required us to Learn OpenCV and C++ and build the mentioned filters from scratch. The filters then would be used on single images as well as a live video feed from our webcam. The live video would just be acquiring single images at high intervals, applying the filter to that frame and displaying it to get the live video with the filter applied. The user would have the selection to save any frame from the video, with the filters applied. Towards the end of the project we are to use different filters in

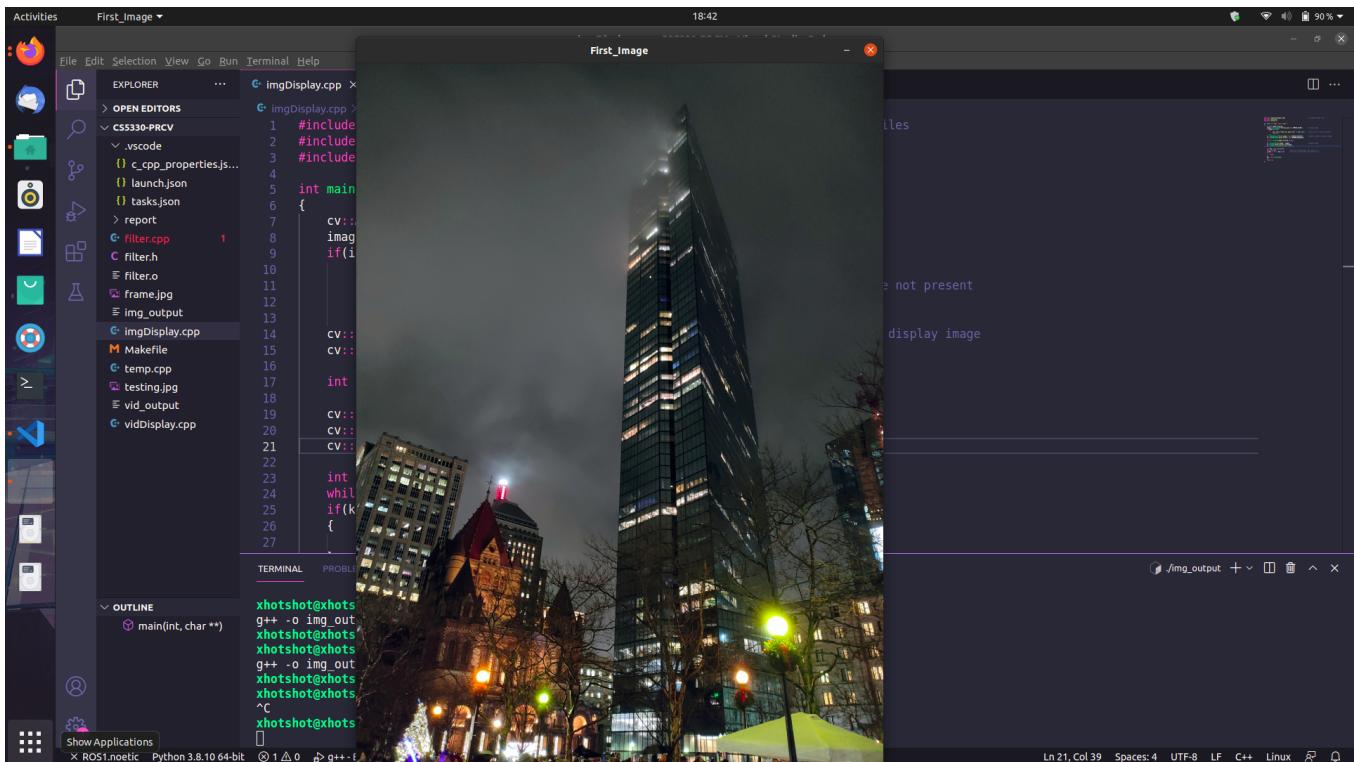
combination to make a cartoonization of the live video and make an additional filter or feature according to our preference.

TASK - 1

Read an image from a file and display it.

Also provide it with the ability to quit when 'q' key is pressed

Output :

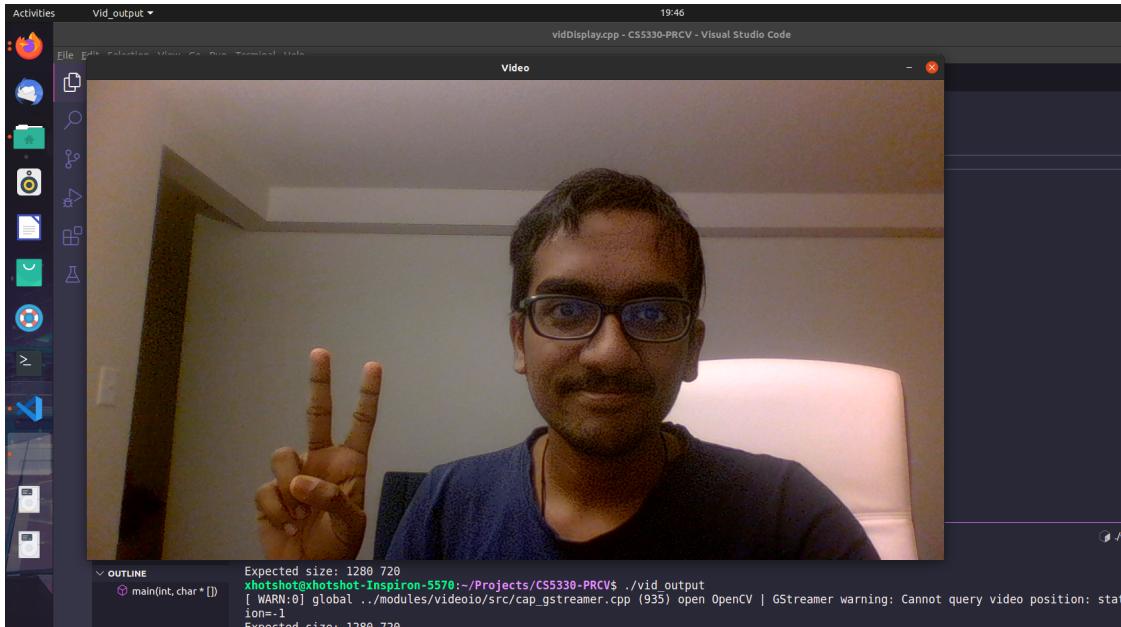


The output is displayed in the above image, in a window named "First_Image"

TASK - 2

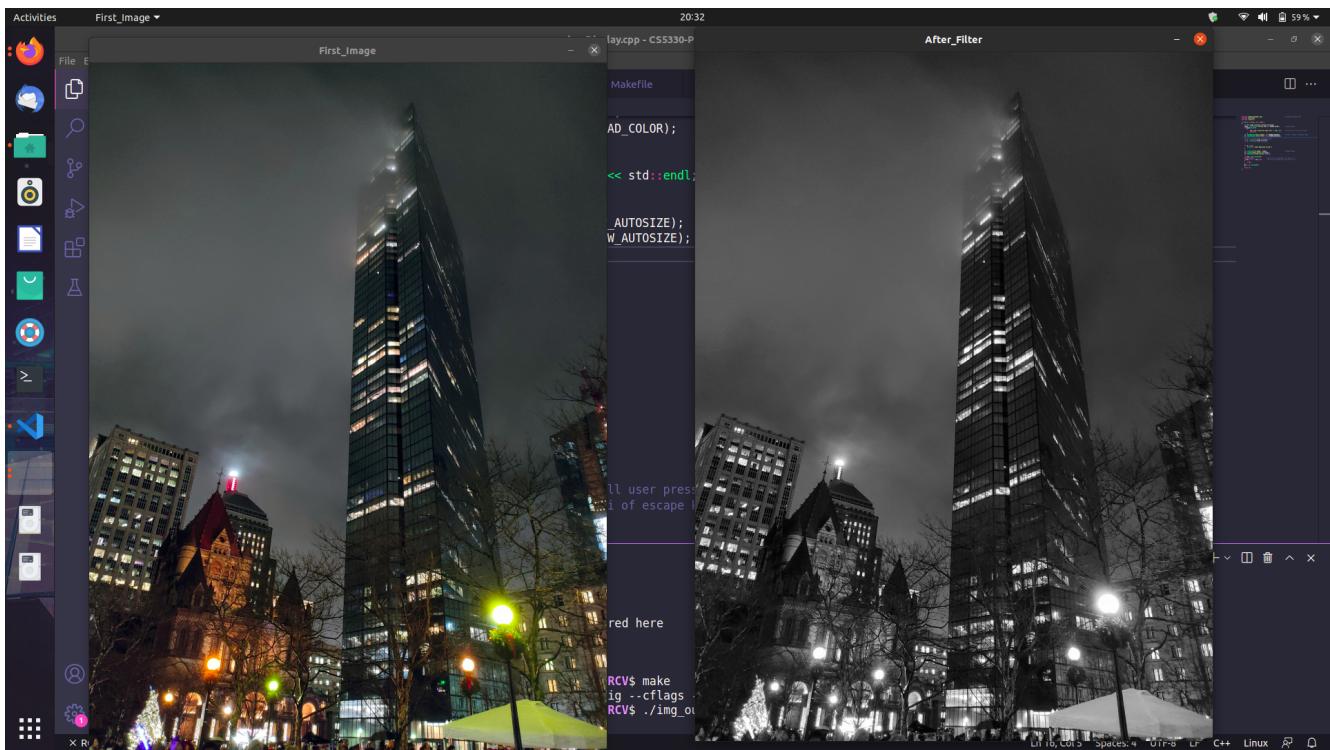
Display live video (press 'q' to quit & 's' to save frame)

Output:



TASK - 3

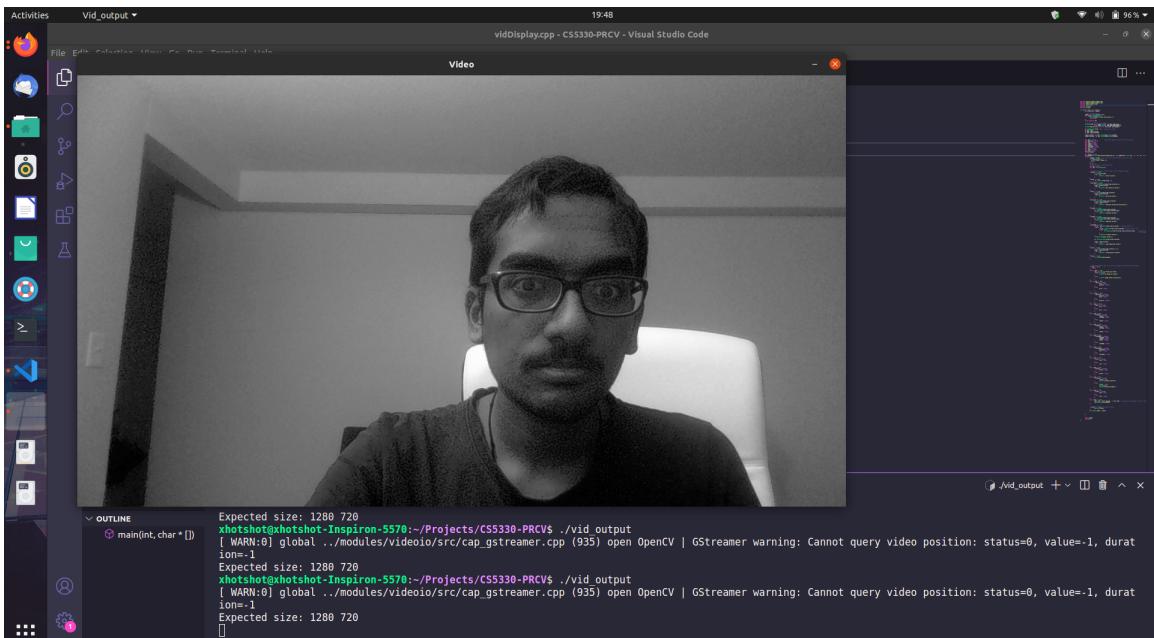
Grayscale image



TASK - 4

Alternative Grayscale Live video (press 'h' to alternate grayscale live video)

Output:



The alternative grayscaling was done pixel by pixel.

To find the final output the following formula was used:

$$\text{RGB}[A] \text{ to Gray: } Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

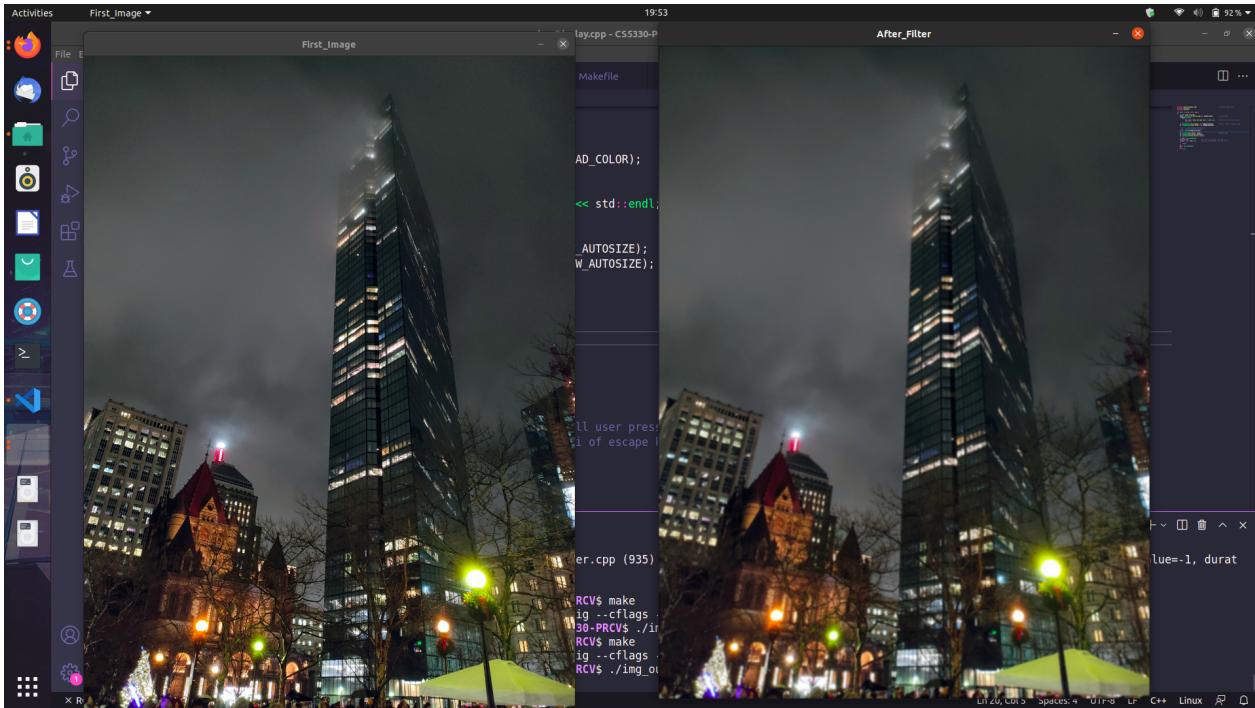
Which was on the official openCV website provided in the project description. (Link below)

https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html#color_convert_rgb_gray

TASK - 5

5x5 Gaussian filter as separable 1x5 filters (press 'b' to blur)

Output:

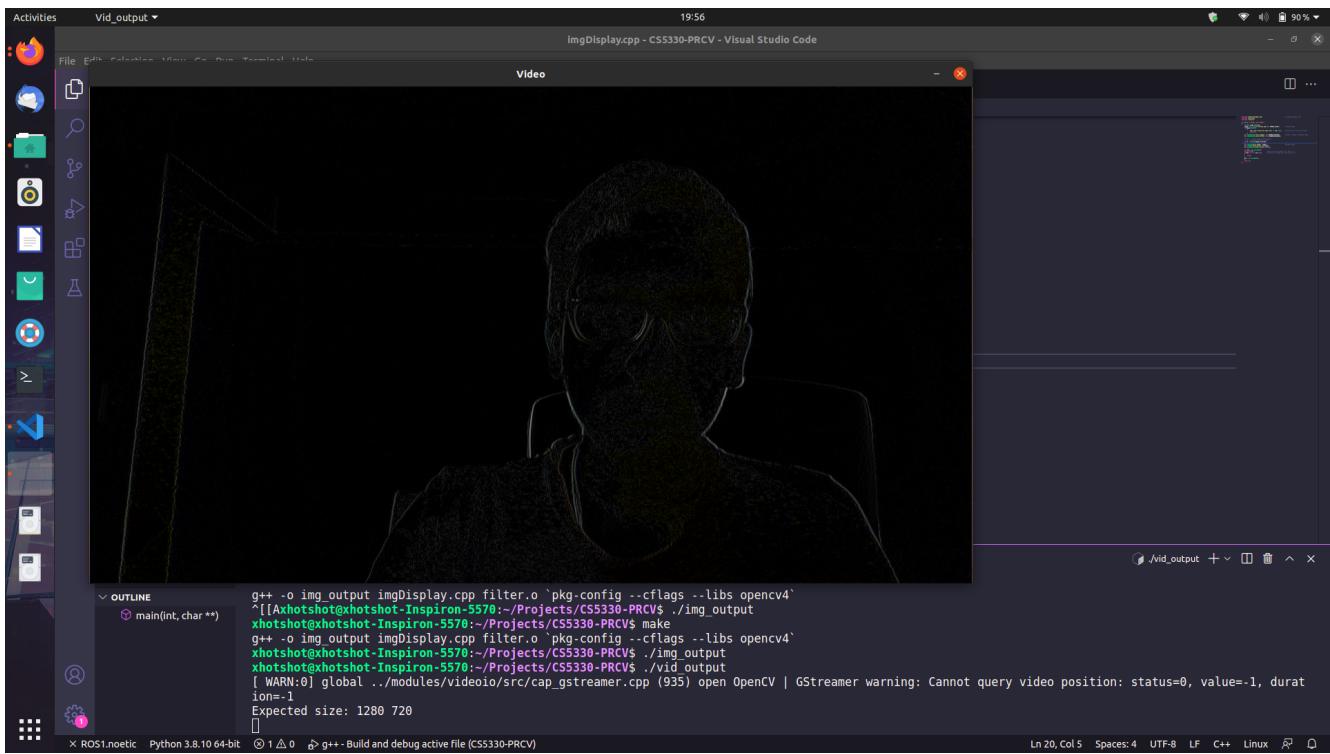


The Left window named "First_Image" shows original image while the right window named "After_Filter" shows gaussian blurred image

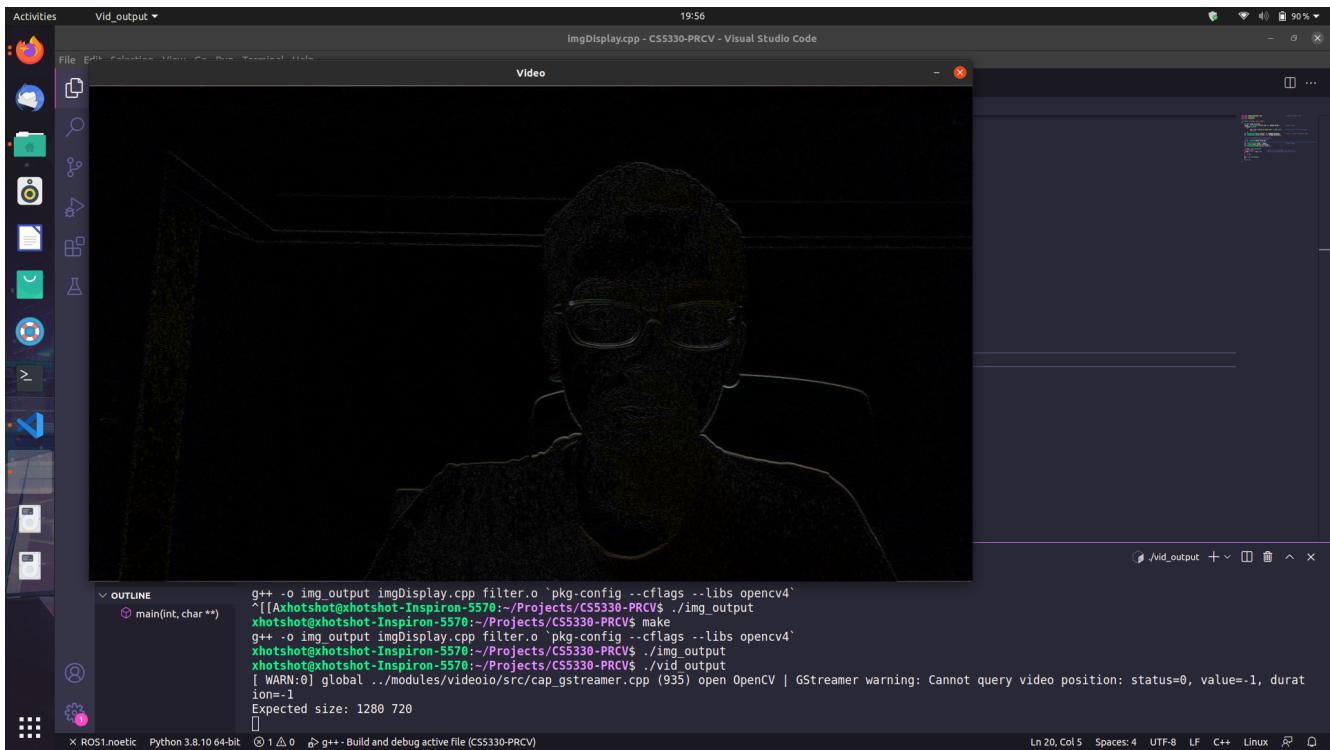
Although the blur was not much visible in the webcam live video, in a high quality image with sharp edges it is actually visible.

TASK - 6

3x3 Sobel X and 3x3 Sobel Y filter as separable 1x3 filters ('x' for sobelX, 'y' for sobelY)



Live video screenshot showing sobelX, showing that it detects vertical edges



Live Video screenshot showing sobelY, showing that it detects horizontal edges

TASK - 7

gradient magnitude image (press 'm' in live video)

Output:

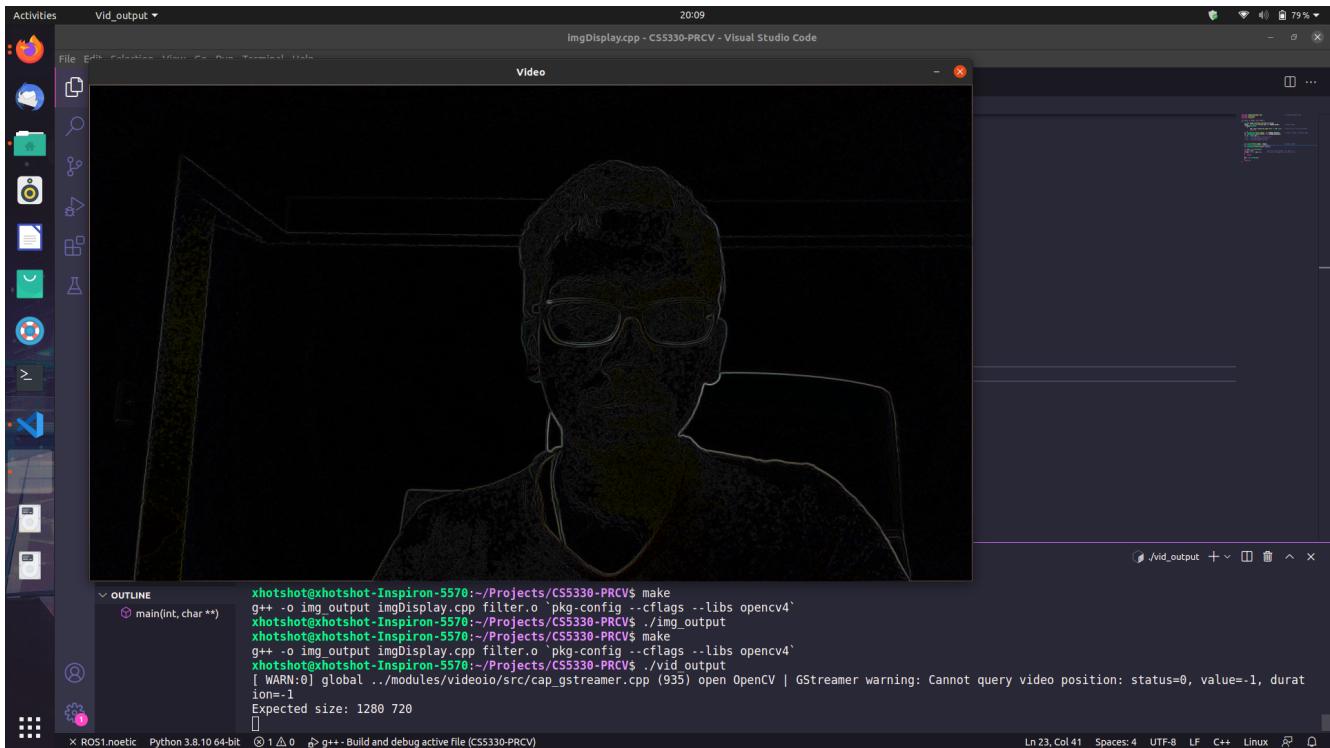


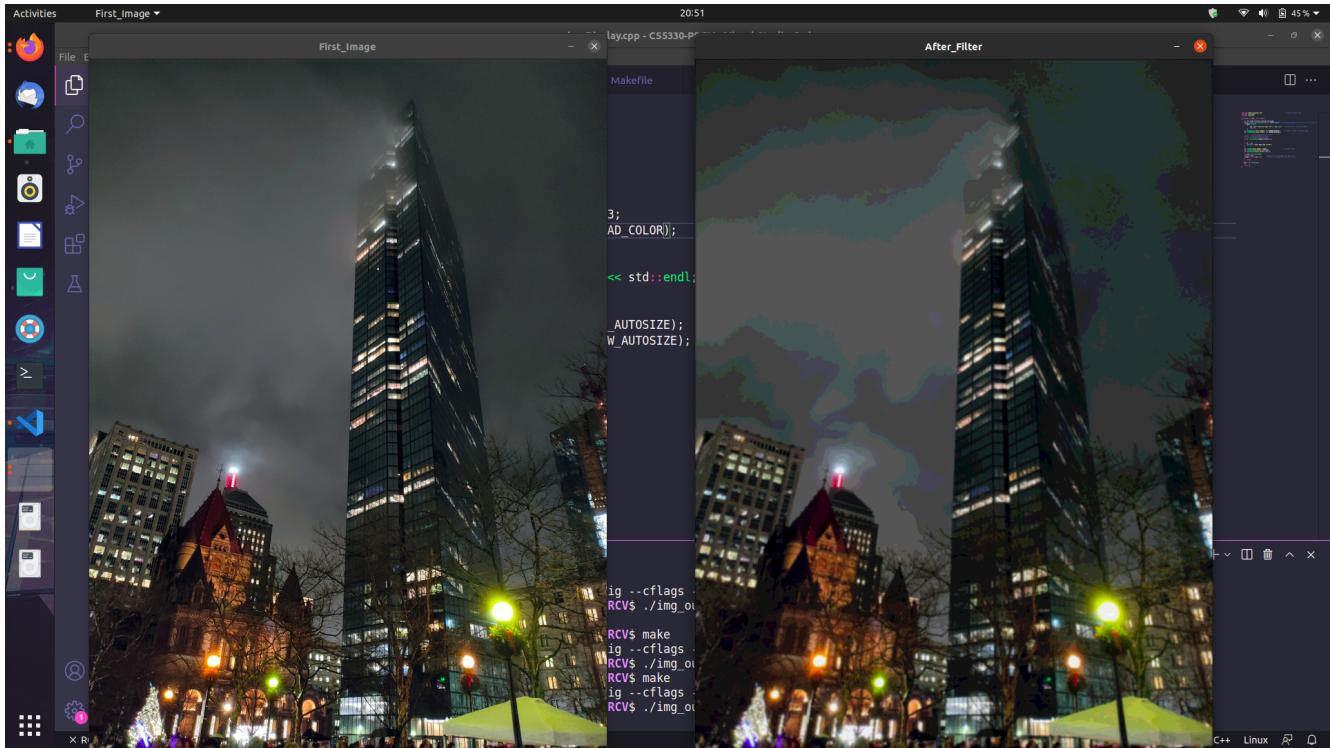
Image showing gradient magnitude of sobelX and sobelY filtered images

The gradient magnitude output is much more crisp compared to sobelX and sobelY filter, and highlights all directional edges

TASK - 8

Blur and Quantize (press 'i' to blur and quantize)

Output:



Left window shows original image, while right shows blurred and quantized image, with level=15

TASK - 9

Cartoonization (press 'c' to cartoonize in live vid)

Output:



Left image is original and right shows after cartoonization

The parameters used for the above images:

Level : 14

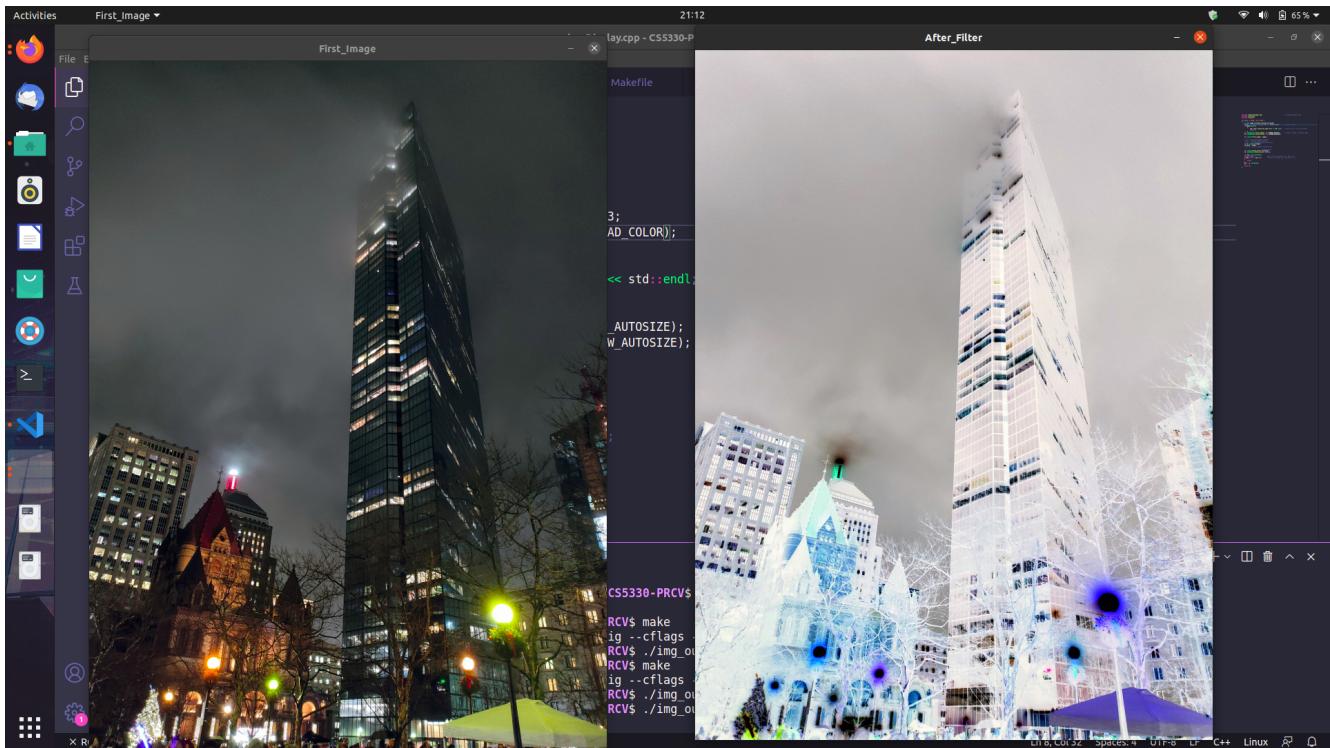
MagThreshold : 14

The cartoonization works well in good quality images. The cartoonization in live video was giving grainy result as my webcam is not of good quality, but it gives a better result when applying the cartoon filter along with an additional blur filter created using gaussian blur (can achieve this by pressing 'c' and then pressing 'b' in the live video)

TASK - 10

Negative/Color Inversion (press 'n' in live video)

Output:



Left side original, right shows negative image

TASK - Extension

Recording feature: press 'r' to start and again to stop recording

Adding Text to live video: press 'p' in live video mode, it will ask to enter text in the terminal, when the text is entered, you can see the entered text by pressing the 't' key.

Most of the filter along with the extension can be seen in the link below:

https://drive.google.com/file/d/1fi_foPieEVRL0OKXBkuolwBHd1xzrjmw/view?usp=sharing

Reflection

I have Learned how to use OpenCV library to read and manipulate images. Also dealing with video as a sequence of images. It was a tremendous experience coding for the first time in c++ and although it was difficult, I had quite the fun. I learned a lot of basic image processing concepts and how to apply them like the filters and visualizing different color channels. I also learned about how to structure different files and the usage of makefile to compile multiple codes simultaneously and with much efficiency.

Acknowledgement

All the Images used were taken by me.

Referenced Stack overflow to help me with debugging the code.

Referred OpenCV official website to write the code.