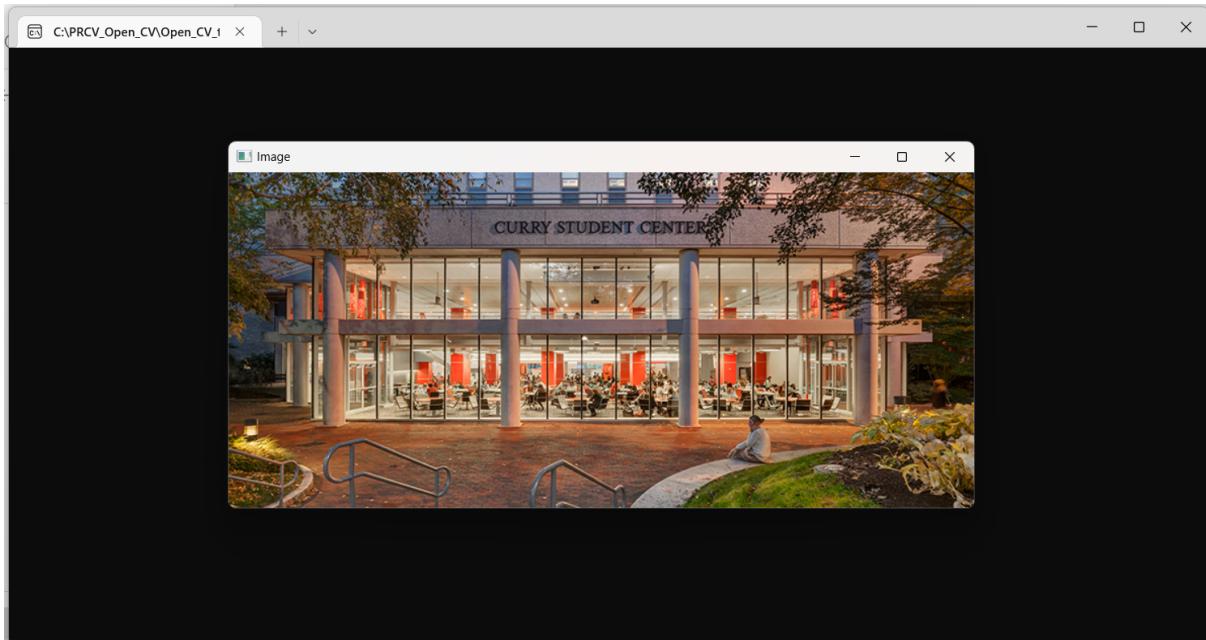


PATTERN RECOGNITION AND COMPUTER VISION - ASSIGNMENT 1

DONE BY: ADVAITH KANDIRAJU (002743436)

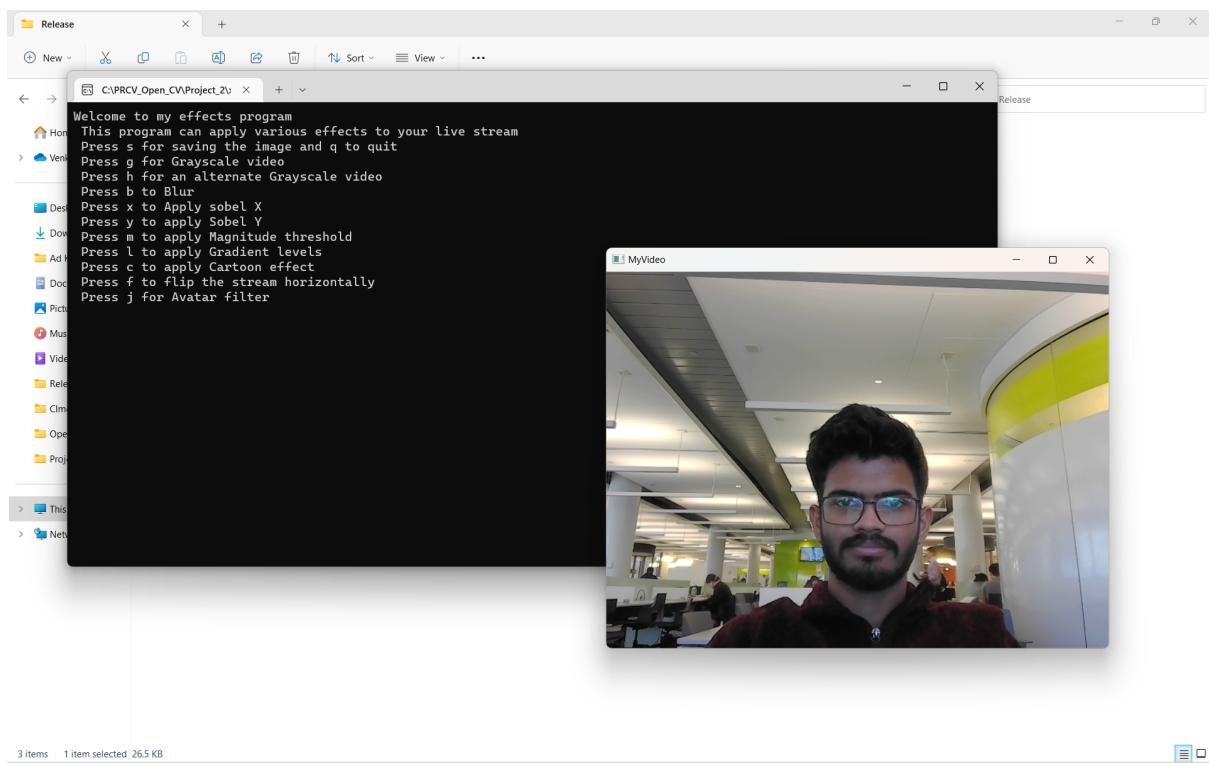
TASK-1: Read an image from a file and display it

imread function is used to read the image and imshow function helps in displaying the image.



I have allocated the location of the image in the same location as the executable file of the program. I have predefined it to read and display 1.jpg image. It is the image which you can see above.

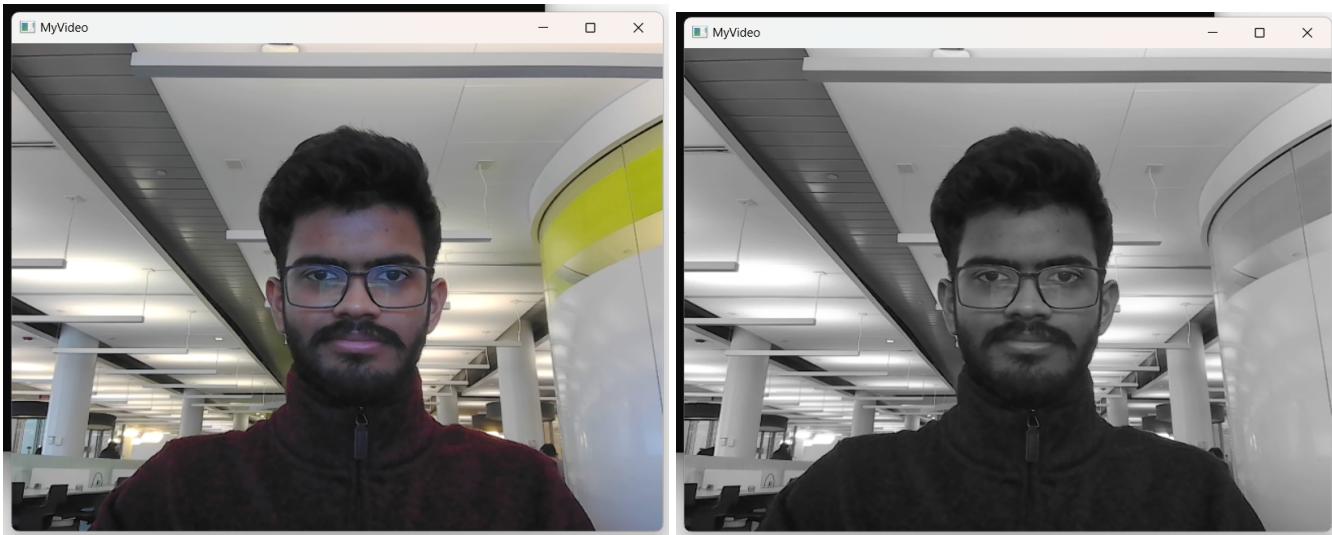
TASK-2 : Display live video



This is the live video that is displayed after executing the program. Above are all the functions that the program can perform.

TASK-3 : Display greyscale live video

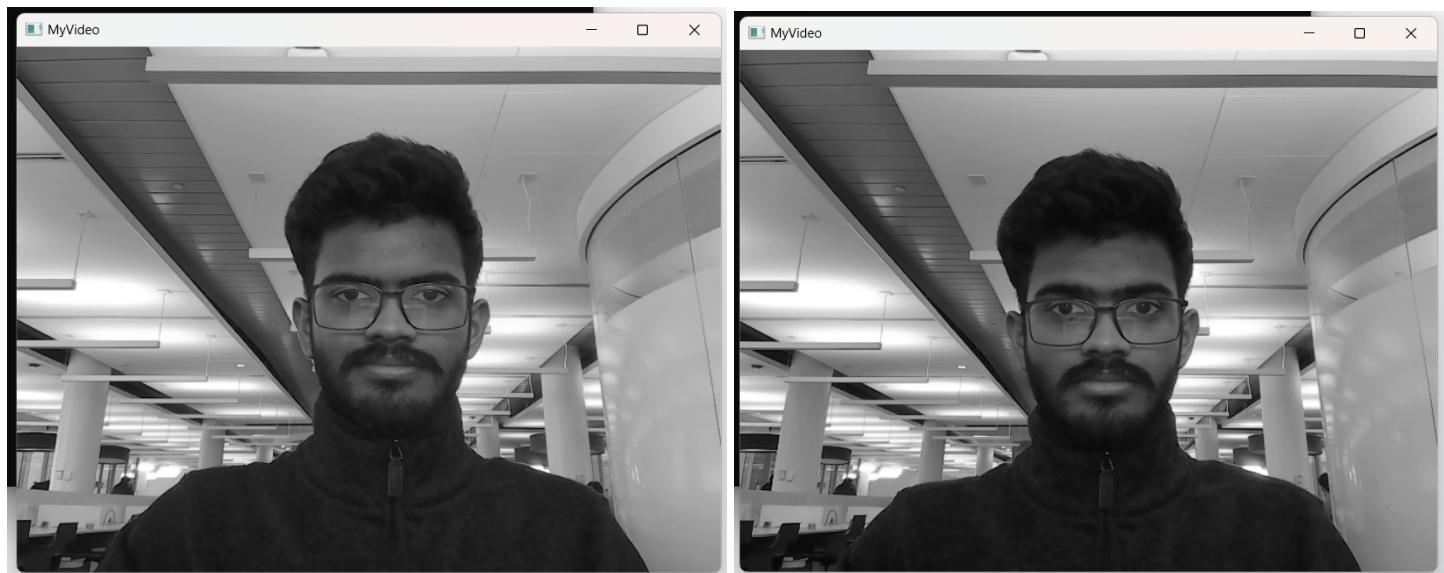
The program will display a grayscale live video upon pressing 'g' and it will save the image as Captured_image.jpg upon pressing 's'.



Required image 1: original and cvtColor version of the greyscale image

TASK-4 : Display alternative greyscale live video

An alternative grayscale live video will be displayed upon pressing 'h' and saves the image as Captured_image.jpg upon pressing 's'.

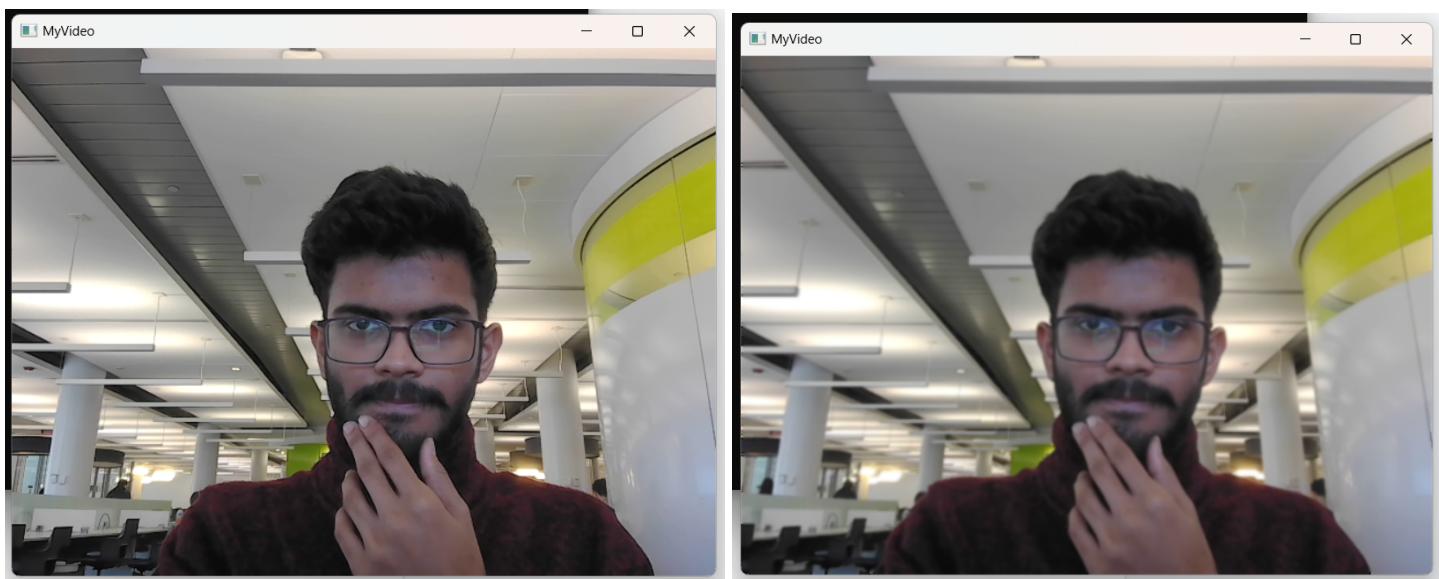


Required image 2: customized greyscale image

The previous grayscale image is shown here to display the disparity between both grayscale filters.

TASK-5 : Implement a 5x5 Gaussian filter as separable 1x5 filters

This effect will apply a gaussian blur filter to the live stream. The stream will turn blurry upon pressing 'b'. Here are the original & the blurred images before & after applying the filter. It saves the image as Captured_image.jpg upon pressing 's'.



Required image 3: original and the blurred image

TASK-6 : Implement a 3x3 Sobel X and 3x3 Sobel Y filter as separable 1x3 filters

Implemented a Sobel X and Sobel Y effect on the video. It displays the X sobel upon pressing x and it displays the Y sobel upon pressing y. It saves the image as Captured_image.jpg upon pressing 's'.

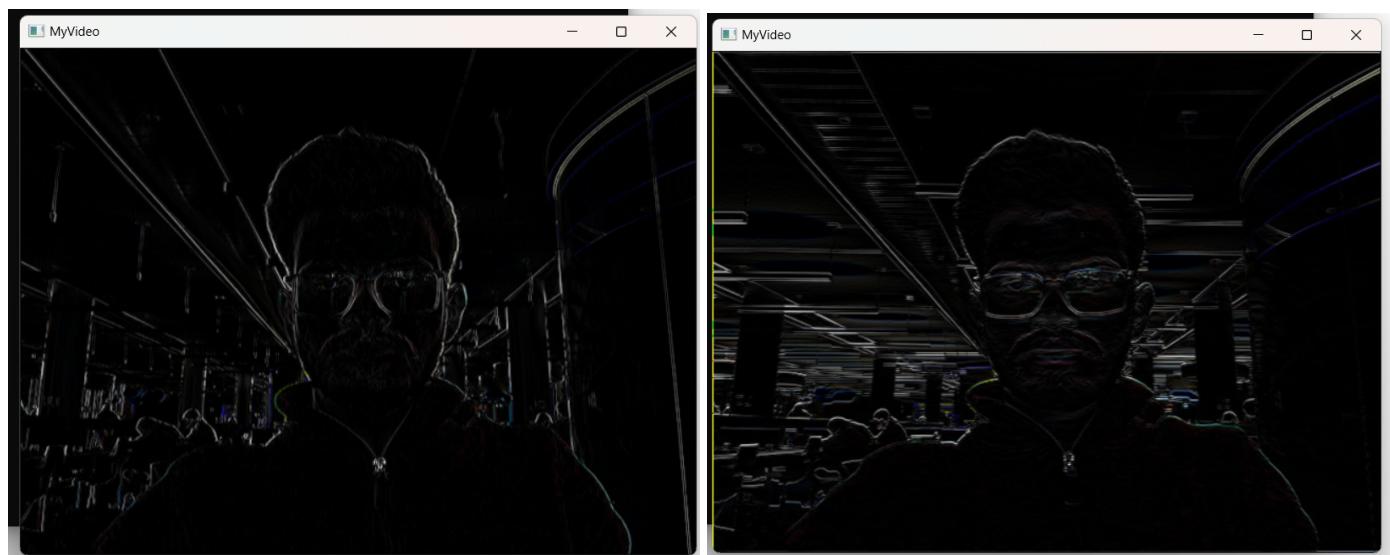
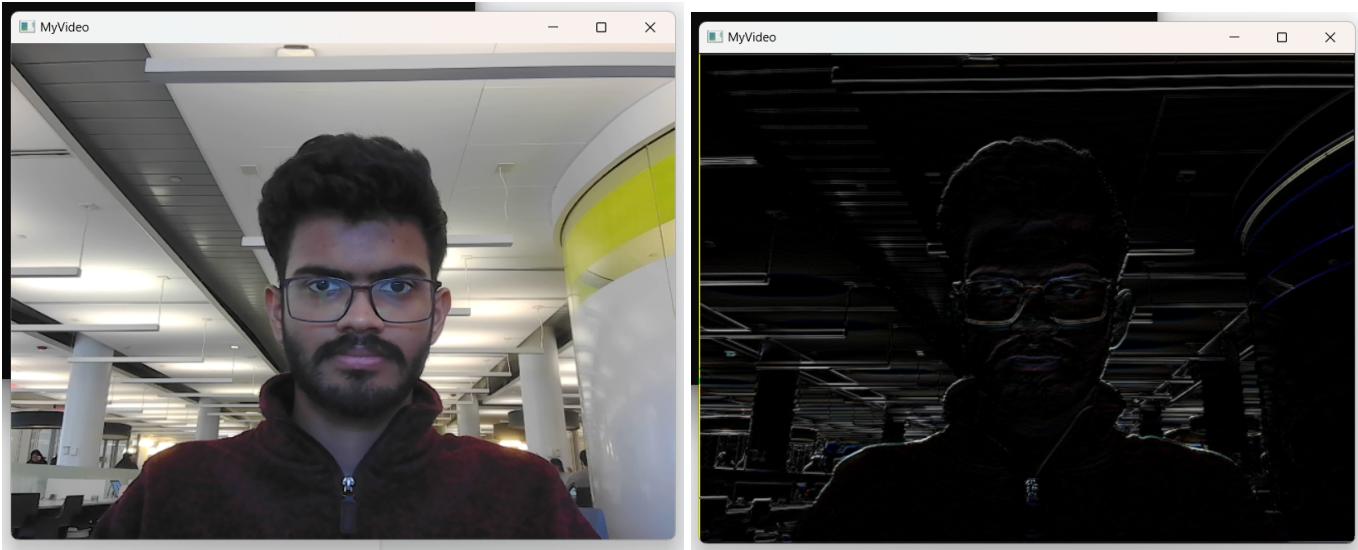


Fig: Sobel-X filter and Sobel-Y filter

TASK-7 : Implement a function that generates a gradient magnitude image from the X and Y Sobel images

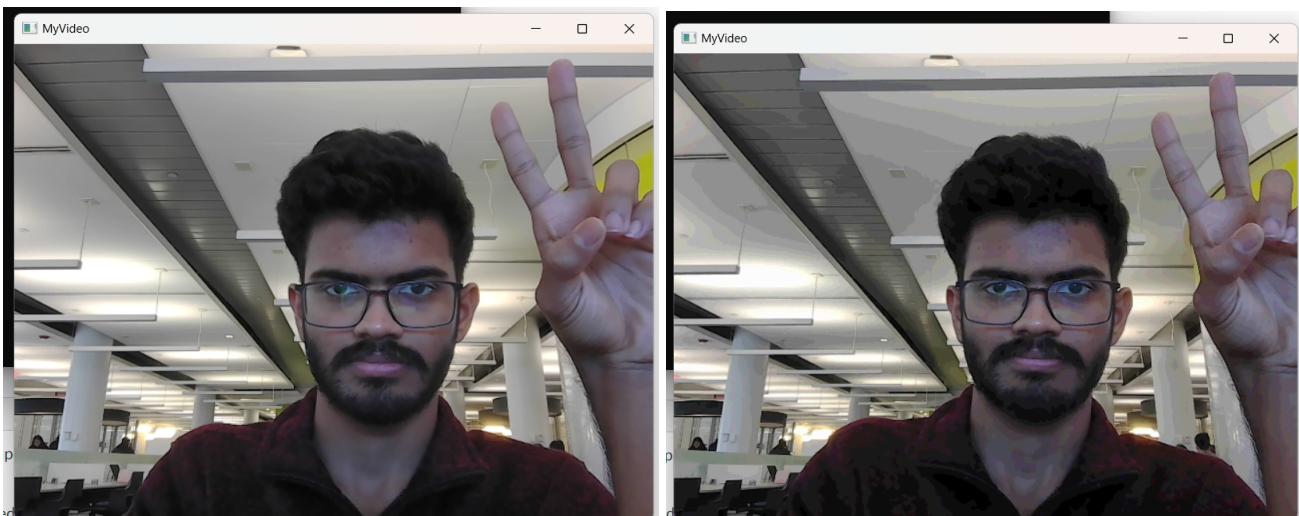
The gradient Magnitude filter is applied to the live video. The live video will turn into color gradient magnitude image upon pressing 'm'. It saves the image as Captured_image.jpg upon pressing 's'.



Required image 4: original and the gradient magnitude image

TASK-8 : Implement a function that blurs and quantizes a color image

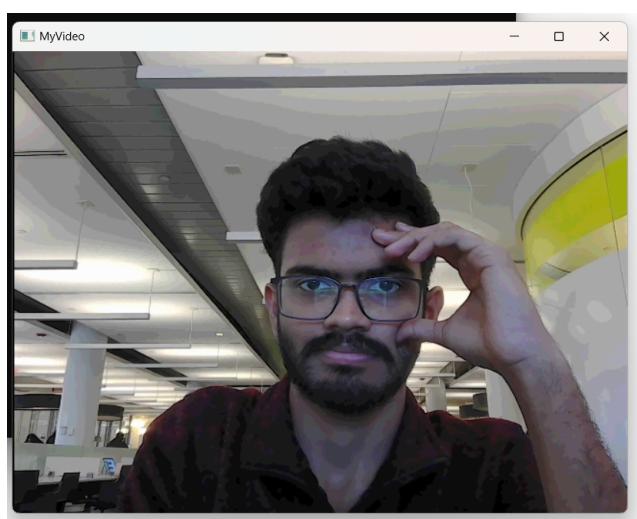
The video will be blurred and quantized in this task. The program will ask the user to input the number of levels and it will display the desired image. Users can press 'l' and input the levels. Here is an example where I have given an input of 20 levels. It saves the image as Captured_image.jpg upon pressing 's'.



Required image 5: original and the blurred/quantized image

TASK-9 : Implement a live video cartoonization function using the gradient magnitude and blur/quantize filters

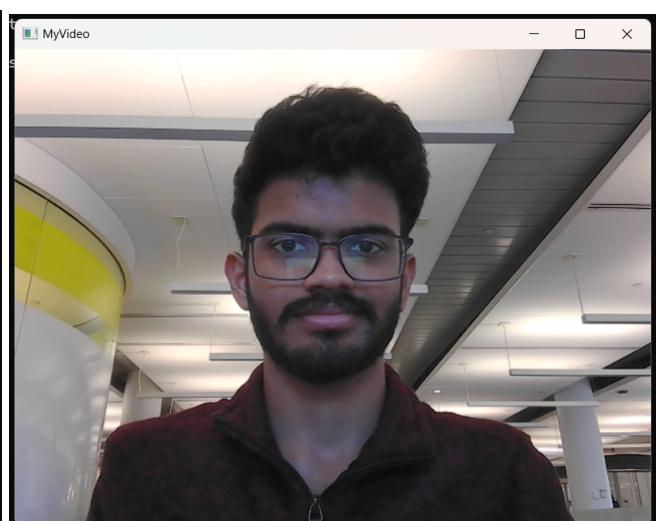
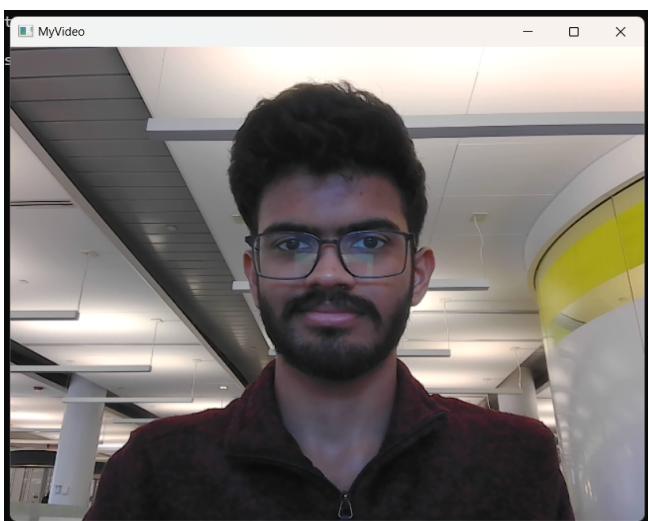
The cartoon filter is applied to the live video using gradient magnitude and blur quantize filters. Users can access this feature by pressing 'c'. Then the program will ask for a Magnitude threshold between 10 and 20. The program also asks for the number of levels. After allocating Magnitude Threshold to 15 and the number of levels to 20, here is the output. It saves the image as Captured_image.jpg upon pressing 's'.



TASK-10 : Pick another effect to implement on your video

I have used a flip function from the inbuilt opencv functions and executed a horizontal flip feature which mirrors the live stream. Users can access this feature by hitting 'f' key. It saves the image as Captured_image.jpg upon pressing 's'.

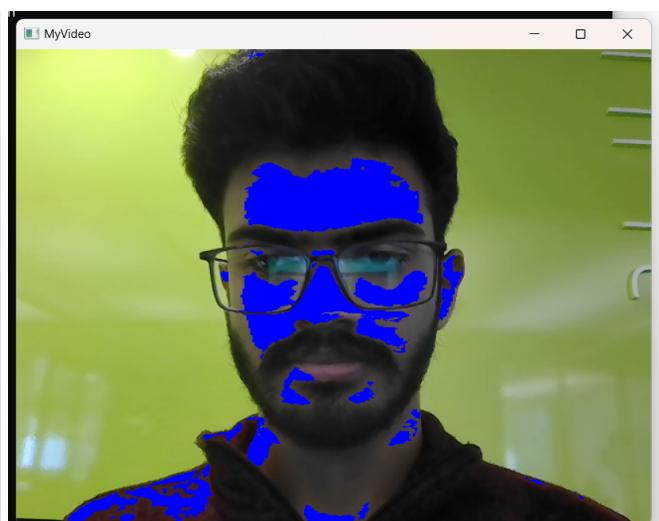
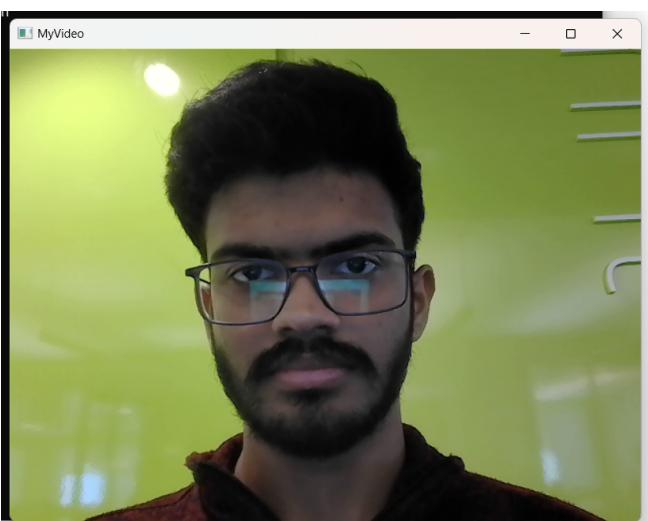
Here is the output:



Extra task for higher grade:

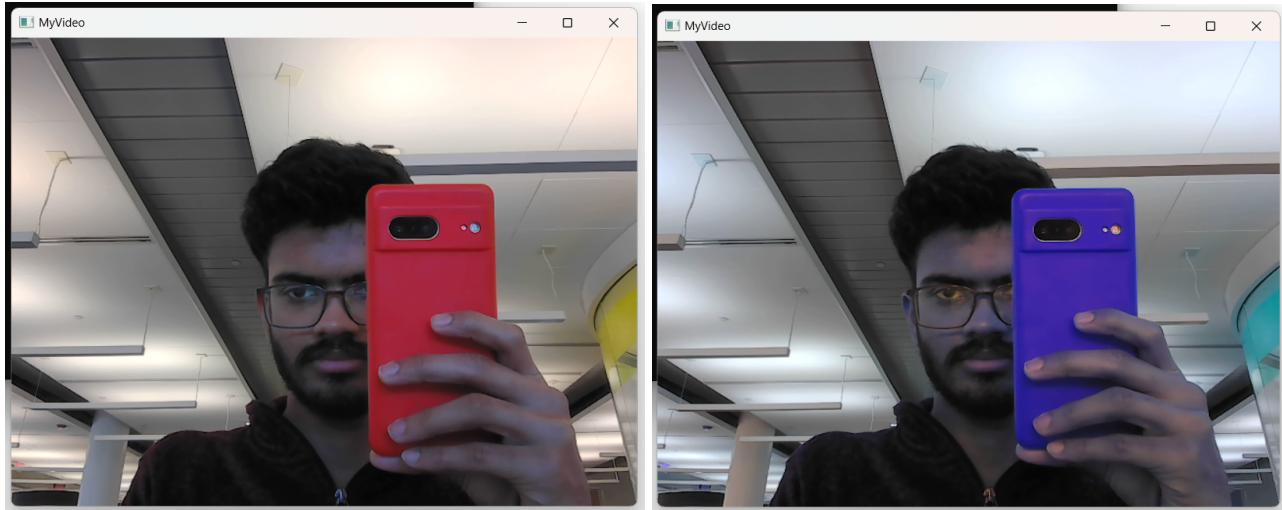
SPECIAL TASK-1: AVATAR FILTER

This feature was built from scratch without using any existing opencv functions. It provides the user with a special avatar filter where the human skin is detected and converted to avatar-blue color. Users can press 'j' to unlock this feature. It saves the image as Captured_image.jpg upon pressing 's'. The output looks like this:



SPECIAL TASK-2: RED TO BLUE CONVERTOR

This special feature was developed by altering pixel intensity values of red, green and blue colours in the image. It will convert any red object to blue colour on the live stream. It saves the image as Captured_image.jpg upon pressing 's'. Users can use this feature by pressing 'k' key. The output:



SPECIAL TASK-3: COLORMAP EFFECT

This feature was developed using an existing opencv function called COLORMAP_JET. Users can use this feature by pressing 'u' key. It saves the image as Captured_image.jpg upon pressing 's'. Here is the output:

