

PATTERN RECOGNITION & COMPUTER VISION
VIDEO VISUAL EFFECTS ASSIGNMENT REPORT
BASIL REJI & KEVIN SANI

Project Description

This project is designed to give us a hands-on experience with C++ programming, specifically using the OpenCV library for image processing tasks. Most of us have used OpenCV library with python so using C++ was an entirely different approach. This project was a great introduction to the realm of image and video manipulation using filters.

Using multiple filters in real time to manipulate images and videos was one of the major tasks in this project. Writing custom functions for filters and pixel manipulation was a great learning experience.

Other tasks involved face detection and implementing unique video effects. The project's extensions involved delving deep into realm of computer vision and encourages us students to explore beyond such as creating more sophisticated filters, creating, and enhancing a user interface and much more. This allows students to be more creative.

The project also stretched the importance of proper documentation and organization. Working on this project will improve and solidify our understanding of image and video processing.

Google Drive link for all videos of the assignment is in the following link:

https://drive.google.com/drive/folders/12jw3SlCW0z768X9te65ukVmmEDm7csVU?usp=drive_link

1. Reading an Image file and displaying it



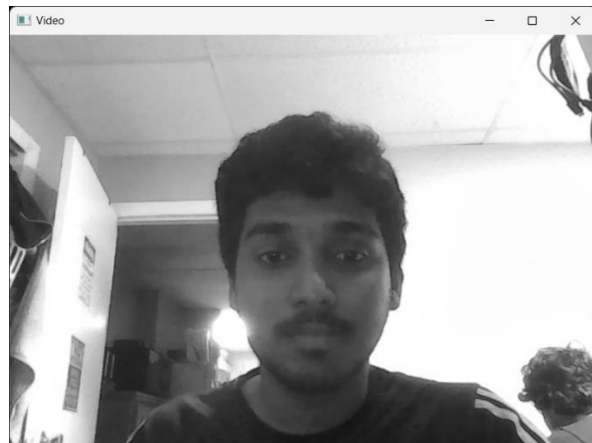
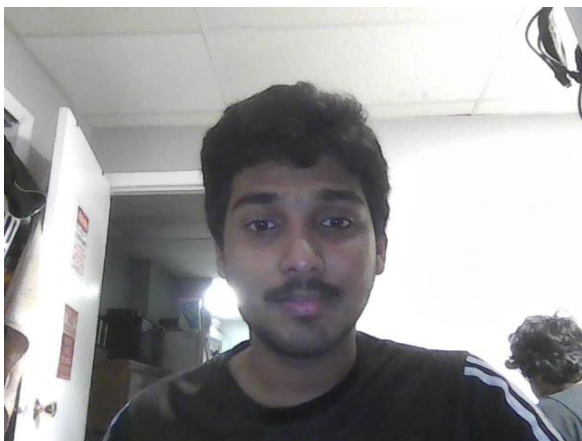
2. Live Video Display

Live Video display was successfully implemented and the keypress loop was added to quit the live feed when we press 'q' and save the image when we press 's'. The link for the video of working live feed is as follows:

<https://drive.google.com/file/d/1jF8MHjNNxiYp1smvxBJasJv7J1mqyn7C/view?usp=sharing>

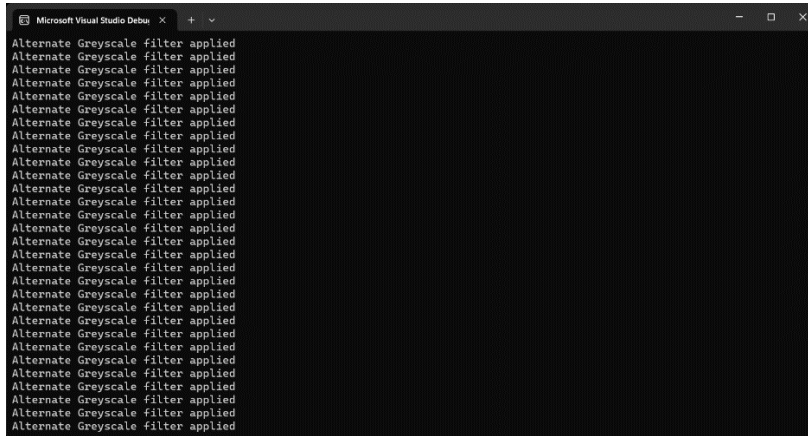
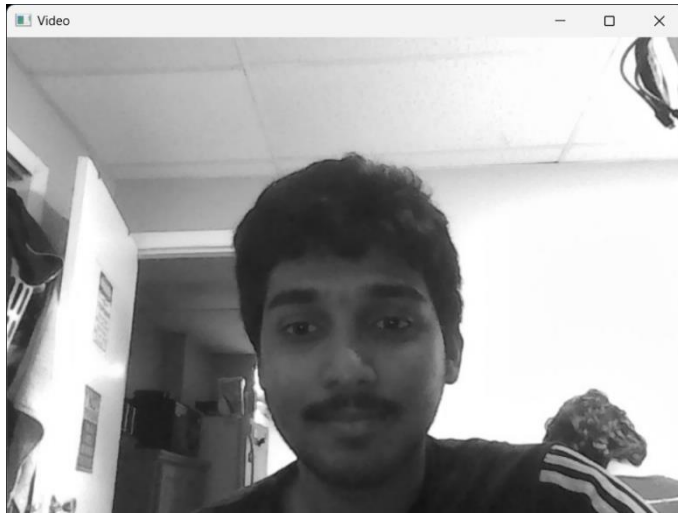
3. Display Grey Scale Live Video

The code was successfully executed where if we press the key 'g', the image will turn greyscale. The



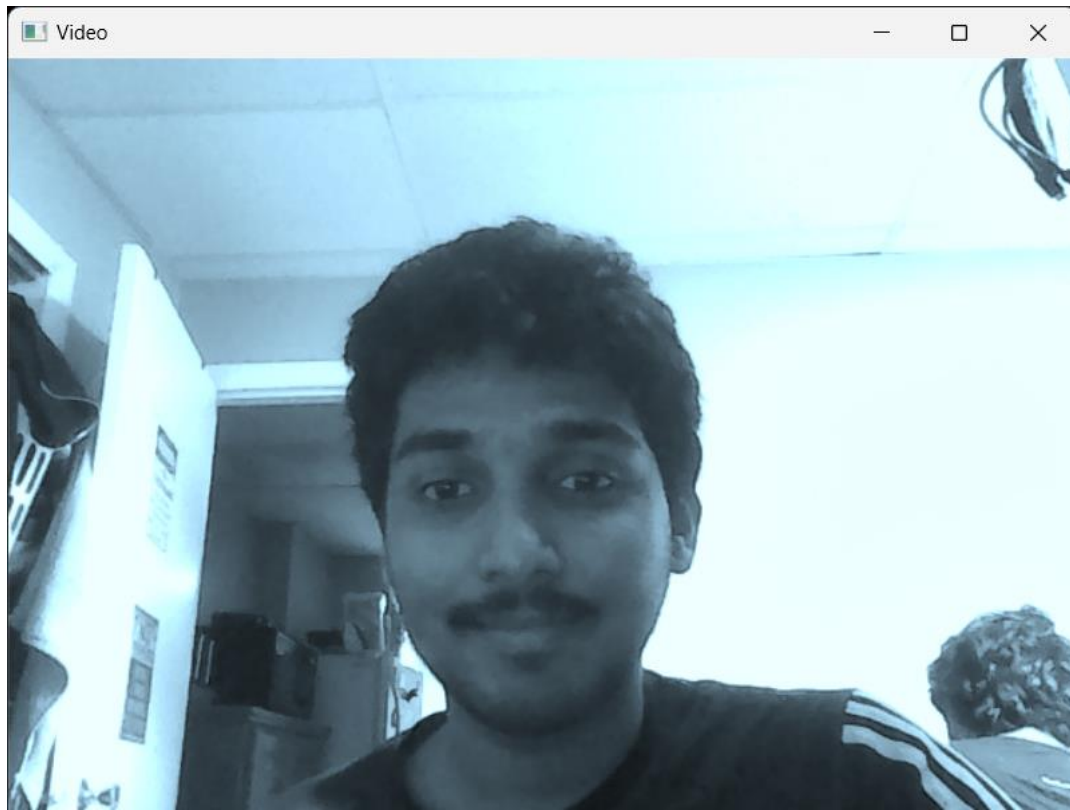
4. Display alternative greyscale live video

Display Alternative Greyscale when we press the keypress 'h'.



5. Sepia Filter:

Sepia Filter will be implemented when we press the letter



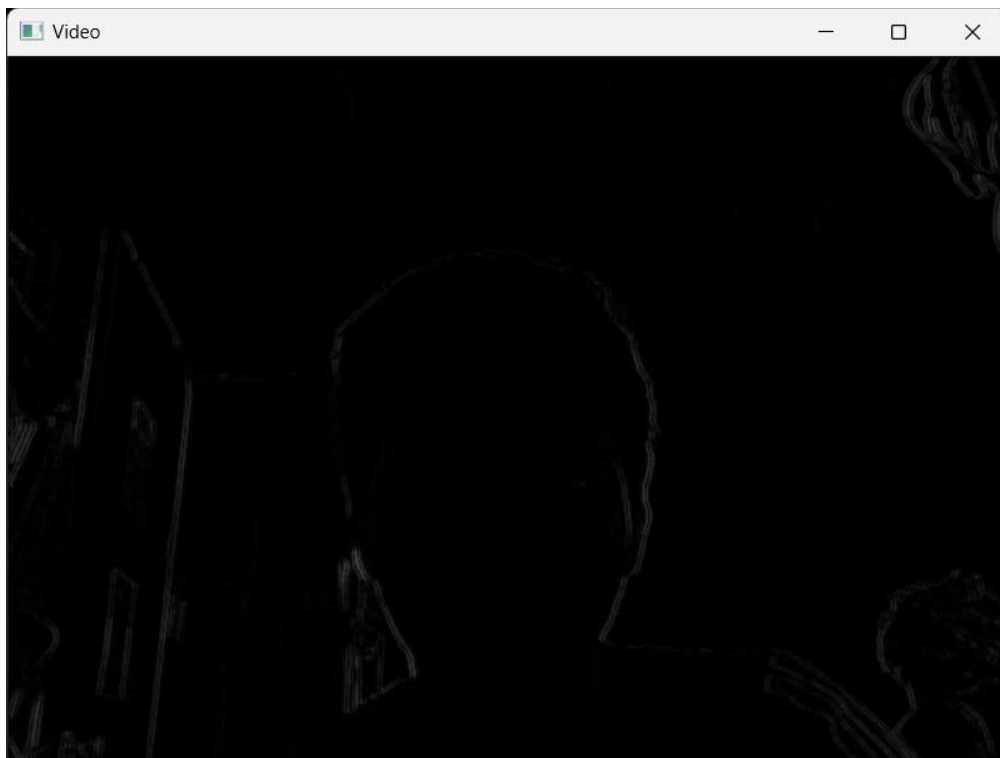
6. 5x5 Blur Filter

Implemented two different blur filters. First function will be called when we press the key 'b' and the second function will be called when we press 'n'.

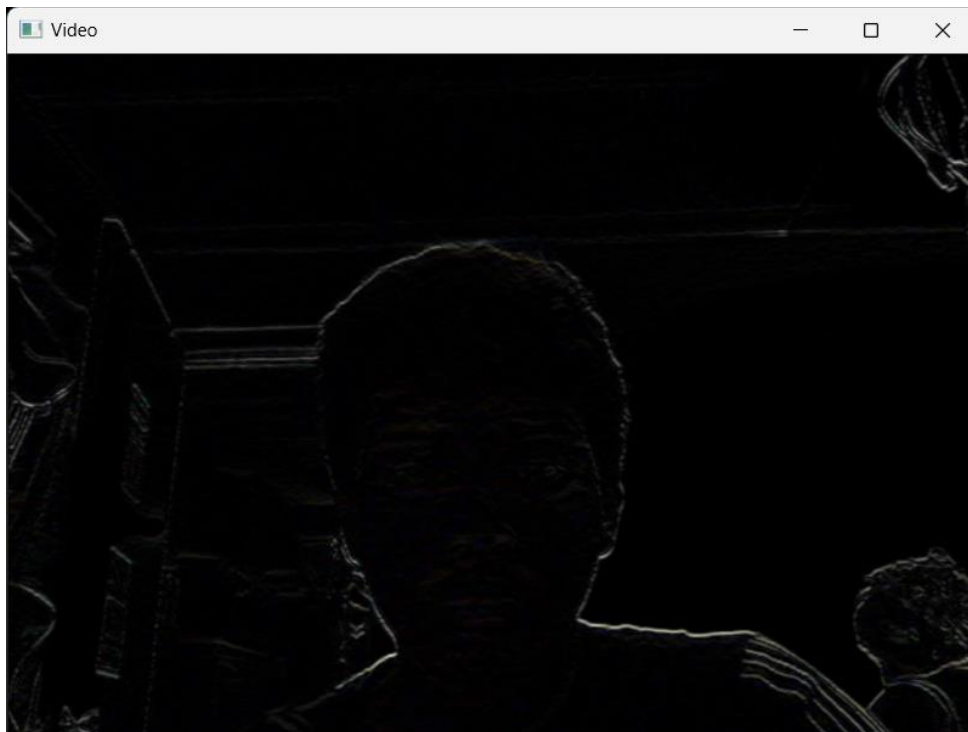


7. 3x3 Sobel X and Y filters:

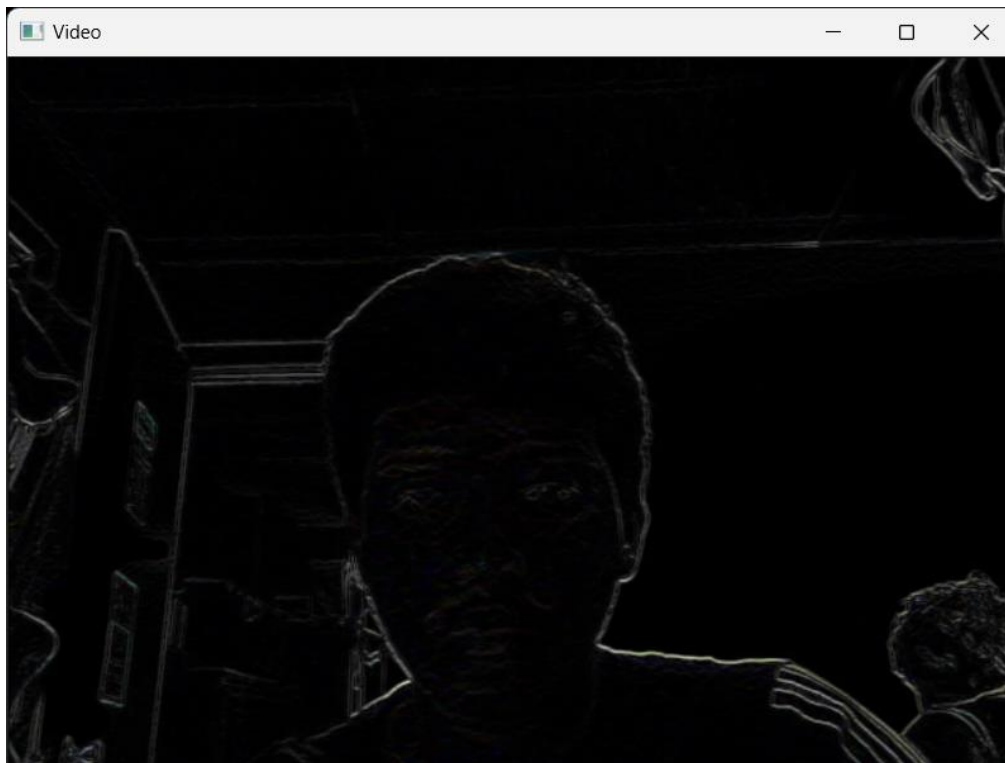
Sobel X filter



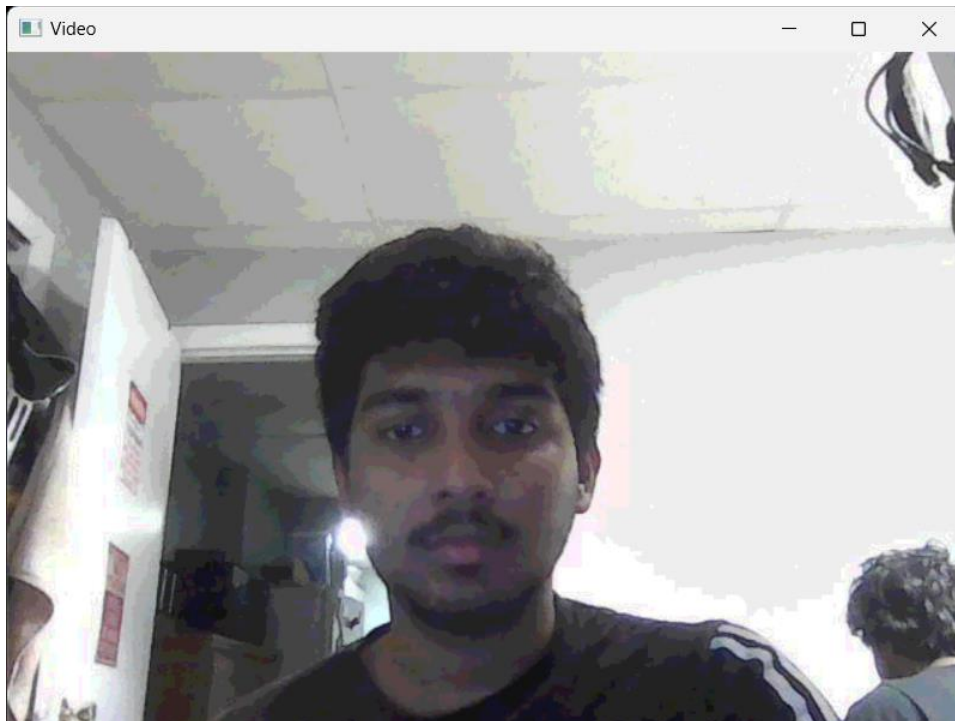
Sobel Y Filter



8. Gradient Magnitude of Sobel X and Y Filter

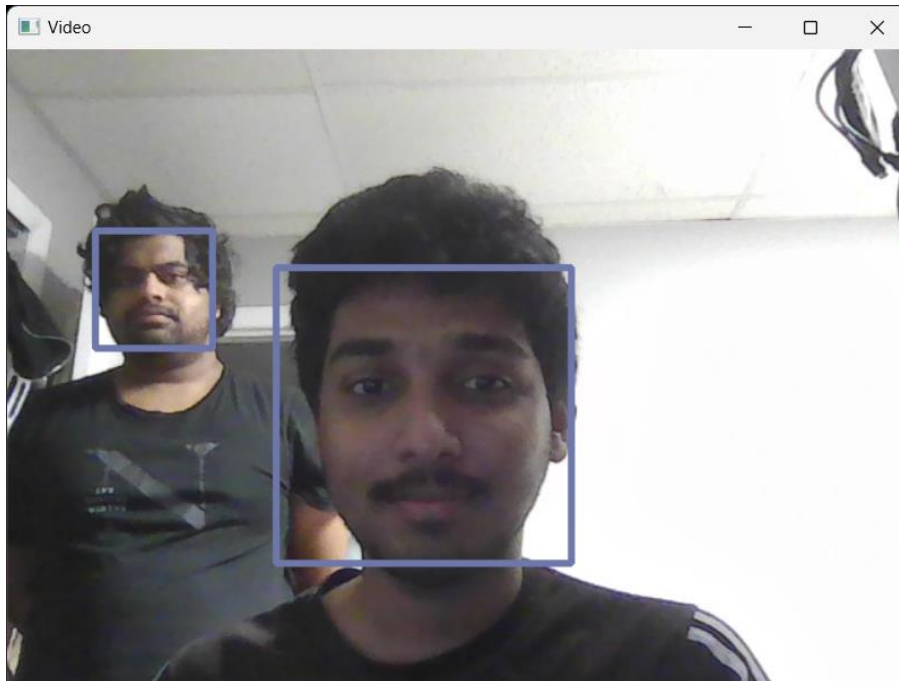


9. Blur & Quantization



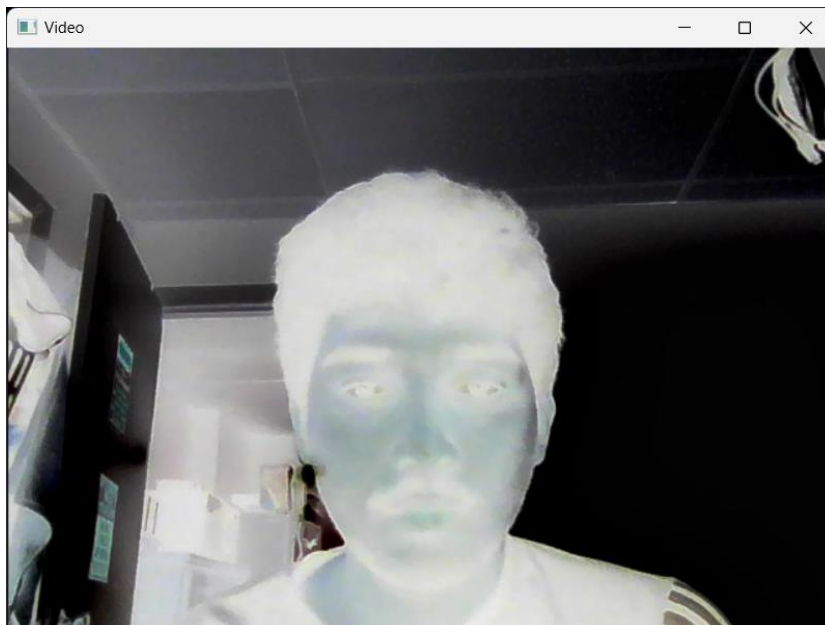
10. Face Detection:

Face Detection of both teammates was done using Haar Cascase function and the code generates squares on all recognised faces



11. Special Effects and Project Extensions:

a) Negative of an Image



b) Histogram Generation:

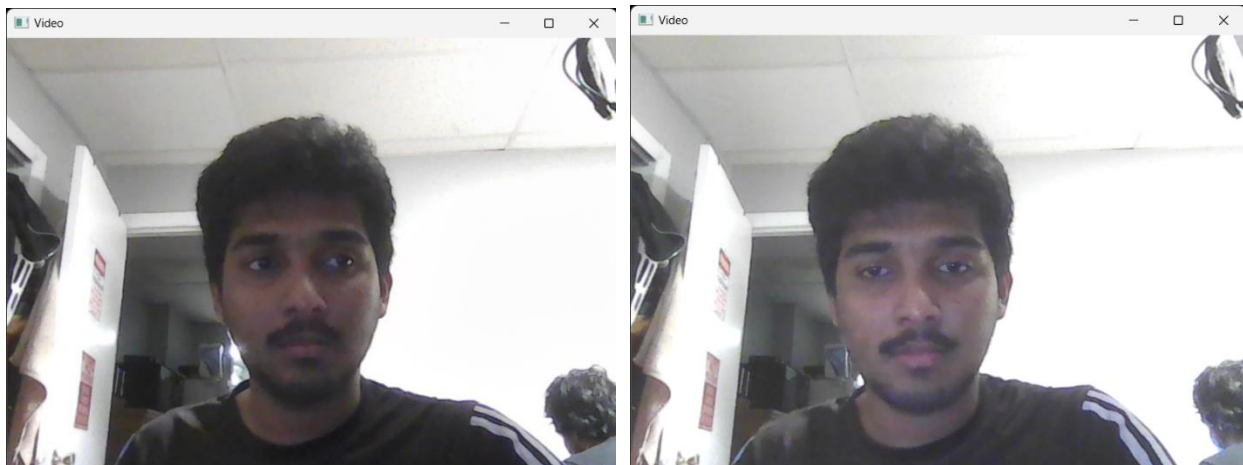
This histogram created is a graphical representation of the distribution of pixel intensities in the image. Its a useful tool for understanding the contrast, brightness, and intensity distribution of the image

Live histogram that tracks the movement was generated and the video of the working histogram is in the following link:

<https://drive.google.com/file/d/1ykw1jpyXw8GID9CbtpR1ksgt6TR6EEj/view?usp=sharing>

c) Increase Decrease Brightness:

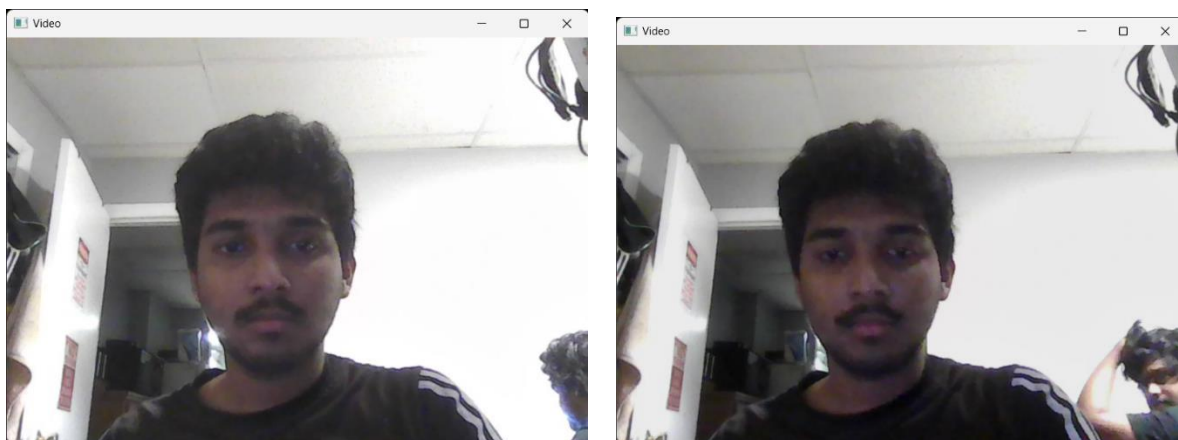
Keypress 'j' will increase the brightness. The screenshot of normal and increased brightness is given below:



Extensions:

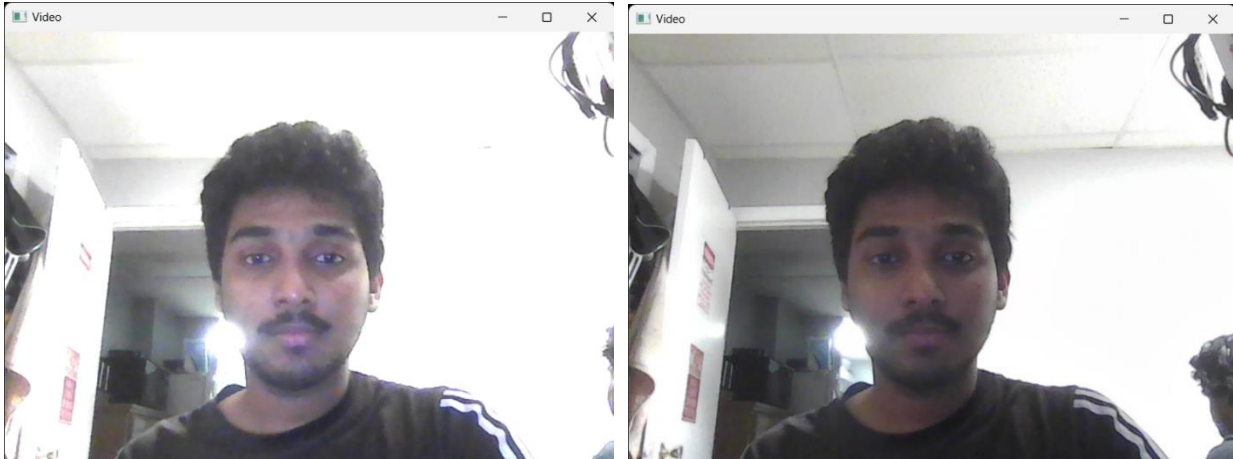
a) Decrease Brightness:

Keypress 'k' will decrease the brightness. The screenshot of normal and decreased brightness is given below:



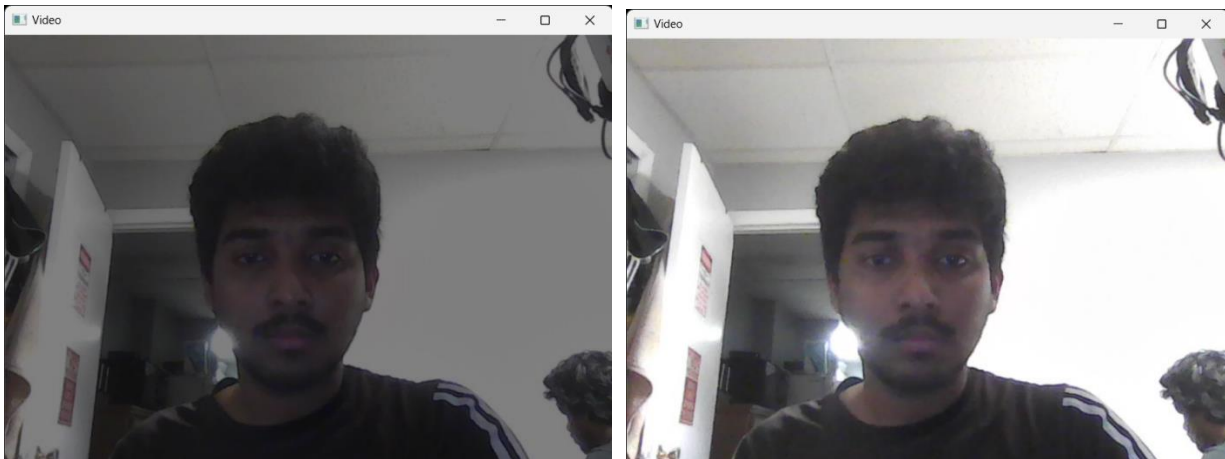
b) Increase Contrast:

Keypress 'o' will increase the contrast. The screenshot of normal and increased contrast is given below:



c) Decrease Contrast:

Keypress 'p' will decrease the contrast. The screenshot of normal and decreased contrast is given below:



d) GUI:

We built a GUI using python and C++. The code for the same is uploaded in canvas.

