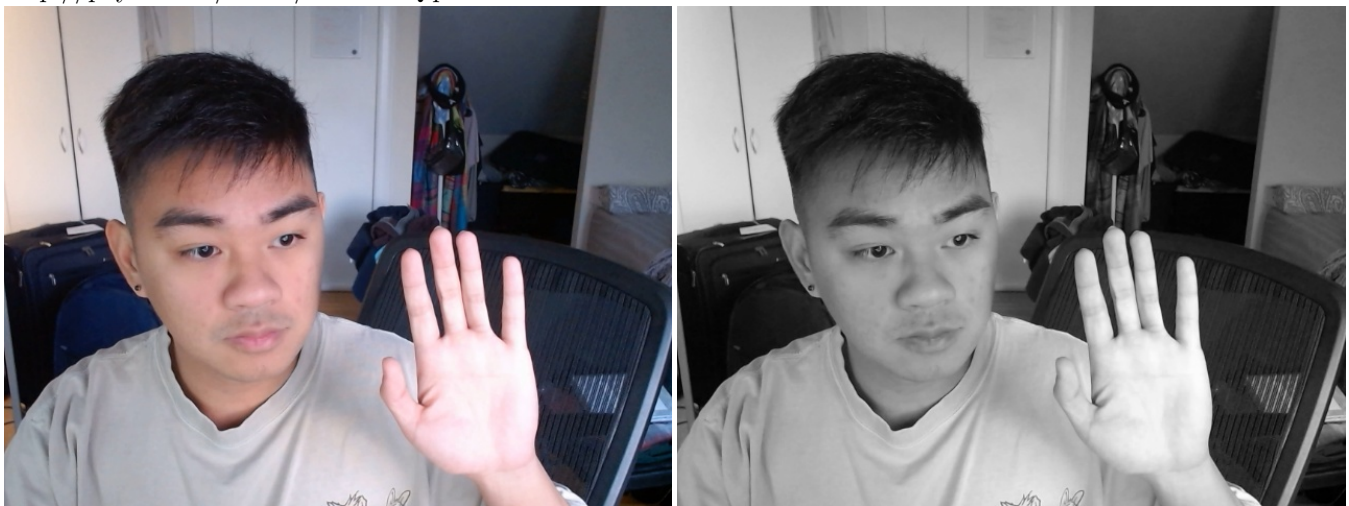


Christian Madali

Project 1 involves real-time filtering of video frames taken from my webcam. Initially, I read in an image that is displayed to the user. I build upon this by displaying a live video feed to the user. Finally, various filters are placed on the video feed. Some of these filters include but are not limited greyscale, blurring, and cartoonization.



Above shows the original (left) and cvtColor (right) version of a frame taken from my webcam. The calculation used to create this greyscale image is $Y = 0.299 * R + 0.587 * G + 0.114 * B$ where R corresponds to red, G corresponds to green, and B corresponds to blue. These weights come from the BT.601 standard for use in colour video encoding. The weights align well with the sensitivity of our eye cones to the wavelength of primary colors. For example, since green has the highest sensitivity, it has the highest weight. More information can be found here: <http://poynton.ca/PDFs/ColorFAQ.pdf>.



Above shows an alternative method of generating a greyscale video. This was done by taking the green color channel and applying it to the other two. The logic behind this is similar to above. Since our eye cones have the highest sensitivity to green, I wanted to use the green color channel rather than blue or red.

Further implementation of the project was attempted but ran into issues with normalization of matrices (could not insert new values into the image upon applying filter and could not be solved within the given time frame).

Overall, I learned basic usage of OpenCV and application filters. Additionally, I am aware of how much optimization and effort was put into the creation of OpenCV function. Upon reflection, I realize I need to focus a little bit more on data types and how to properly format variables so that they are compatible and do not run into seg fault errors.