Task 7: RGB Thresholding

When I did thresholding on the RGB channels separately, each channel showed different results because they react differently to colors. The red channel highlighted bright, warm areas like sunlight or light bulbs. The green channel gave balanced and clear details because it's closer to how human eyes see things. The blue channel had more noise and was good at showing darker or cooler areas. Using sliders made it easier to clearly see these differences.

Colour Space Conversions (Tasks 10 & 11)

Converting the image to HSV and YCbCr colour spaces made thresholding clearer compared to RGB:

HSV Thresholding: Thresholding the Saturation (S) channel clearly separates bright colors from dull ones. Thresholding the Value (V) channel helped identify bright areas from darker shadows. Overall, HSV thresholding gave cleaner and less noisy results compared to RGB because it separates brightness and color better.

YCbCr Thresholding: Thresholding the Chroma-red (Cr) channel clearly identified skin tones and red areas, which is helpful for face detection. The luminance (Y) channel effectively showed the difference between bright and dark areas. YCbCr was also clearer and less noisy compared to RGB.

Face Detection and Modification

Using ml5's Face API, faces were detected well, creating boxes around them. The privacy filters I applied were:

- Greyscale: Made faces less clear, but you could still recognize some details.
- Blur: Made faces hard to recognize by hiding details well.
- HSV Colour Conversion: Changed colours in a way that also helped hide identities.
- Pixelation: Provided the best privacy by making the face very unclear. These filters showed how image processing can protect privacy effectively.

Unique Extension:

Hand Gesture Control (Task 14 & Extension) I added real-time hand gesture detection using the HandPose model from ml5.js. This feature allowed users to pick filters using hand gestures, making the app more interactive:

- Gesture Control: Recognized finger positions and connected certain gestures to specific filters.
- Technical Challenges: Needed careful adjustments for accurate detection in different lighting.
- User Experience: Made the app easier and more fun to use by allowing hands-free control.

Problems and Solutions

I faced issues with face detection inaccuracies when lighting changed. I addressed this by fine-tuning the detection settings to improve accuracy. Hand gesture detection was inconsistent, either due to thresholding issues or gestures not being recognized properly. I improved this by carefully adjusting detection parameters and refining finger recognition through extensive testing.

Reflections on Development

Looking back, if I had tested my code more regularly and systematically from the start, it would have reduced problems later on. Adding logging earlier would have helped me quickly spot small errors, making the whole development process smoother and more efficient.