The program size will be at least full screen or almost the entire screen size because there are a lot of features that need to be deployed.
The original values at the beginning is the following

- **Brightness**: 50
- **Rotation**: 0
- **Viewpoint**: None
- **Select a box in the image, then write the numbers for the color below**: The values for RBG will be blank
- **Computer Vision Algorithm**: The range will be 200 for lower bound and 255 for upper bound. When the user clicks the checker, it should get a new pop up as saying pass.
- **Color Checker**: Red: 207, Green: 20, and Blue: 43. Note that blue and green can't change as the user modify the options.

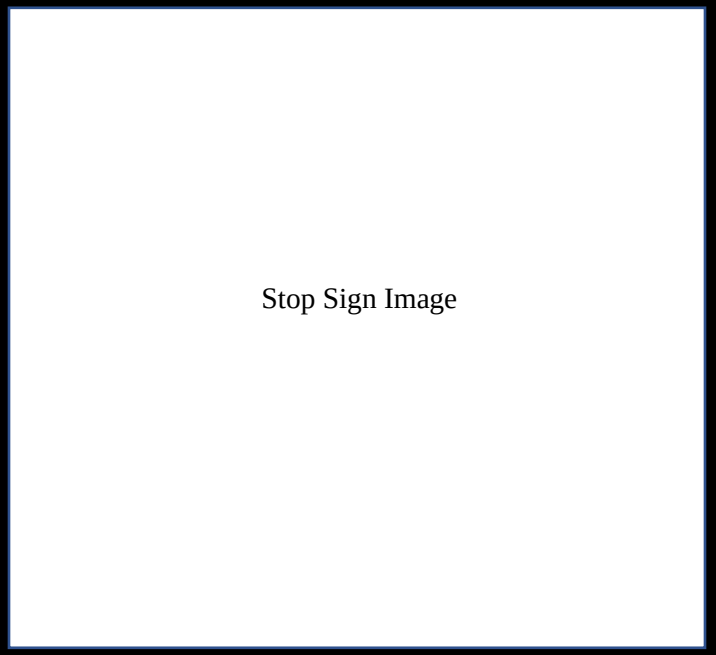
Features in the modification

- **Brightness**: it will change either entering the value of the number or using the slider. Once the user selects the button right next to the word brightness (on the left of the word), then it will modify the image. (If the user input invalid value, then will return an error says invalid range) (range: 0 ~ 100: 0 as dark while 100 is brightest)
- **Rotation**: same idea with brightness. (range: -180° ~ 180°)
- **Viewpoint**: It will have an option to select, then clicking the button will change the image that is located next to the word of Viewpoint. The options are up, down, back, left, right, front, and none.
- **Select a box in the image, then write the numbers for the color below**: When the user selects a pixel (or box), then entering the numbers of RBG and clicking the button next to Preview Color will only change the Preview Color Box that is located below the word. To modify the pixel in the image, then the user will need to select the button next to Red.
- **Computer Vision Algorithm**: User can change the range of red that the computer will recognize as red. As the user press confirm, it will update the text message below as well as the algorithm of identification. As the user uses the checker, it will show up and pop up whether the checker pass or fail. When we say the color red, we are talking about the lower bound number of red.
- **Reset Everything**: It will go back to the original values as the image goes back its original.
- **Survey Link for this Study**: The button selection will be linked to a hyperlink which will go to the survey

As the user changes the values of the image, the image and the words will change; for example, if the value change in the color of RBG, then it will update the image with that color, and might change the computer thoughts of the image(from stop sign to a dog)

You find this program in this GitHub Page: https://github.com/MrKangs/URSA_2020

Title of the Program. Created by Kenneth Kang for URSA 2020



Brightness #

Rotation #

Viewpoint

Select a box in the image, then write the numbers for the color below.

If the red value is in

Red: # Blue: # Green: #

Preview Color

Computer think this thing as:_____

Reset Everything

Survey Link for this Study

Computer Vision Algorithm

Lower Bound for Red #

0 255

Upper Bound for Red #

0 255

Confirm

Please click the checker if it passes those three conditions. If not, the program will fail.

Does…

- Red > Green
- Red > Blue
- Blue ≈ Green

Checker

If the red value is in between [lower bound] and [upper bound], then the computer will understand that it is red.