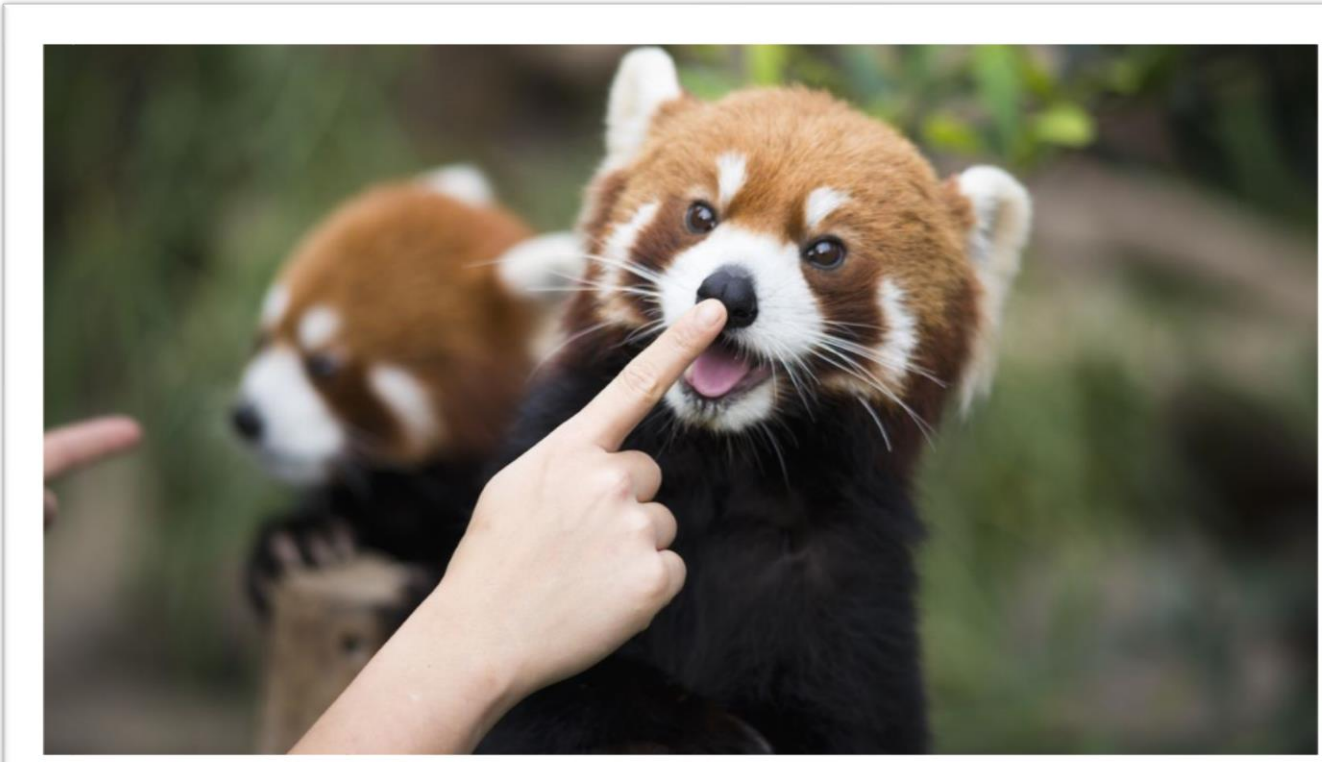


Red Panda Image Recognition

(PULLING AN OBJECT OUT FROM A PICTURE)

JACOB NADING

Process (Thought & Code)

I have created a basic MATLAB script to have a general format and structure for analyzing images the way I did. *FinalProject.m* is that script. It is included in the folder among all 4 image analysis iterations.

The first step was to load in the image. Once the image has been given to MATLAB, I separated the image into its three color spectrums: red, green and blue. After looking at all of them, this is the step in which After looking at all of them, this is the step in which you must decide which of the three RBG spectrums is most differentiated between the background and the intended object to be isolated.

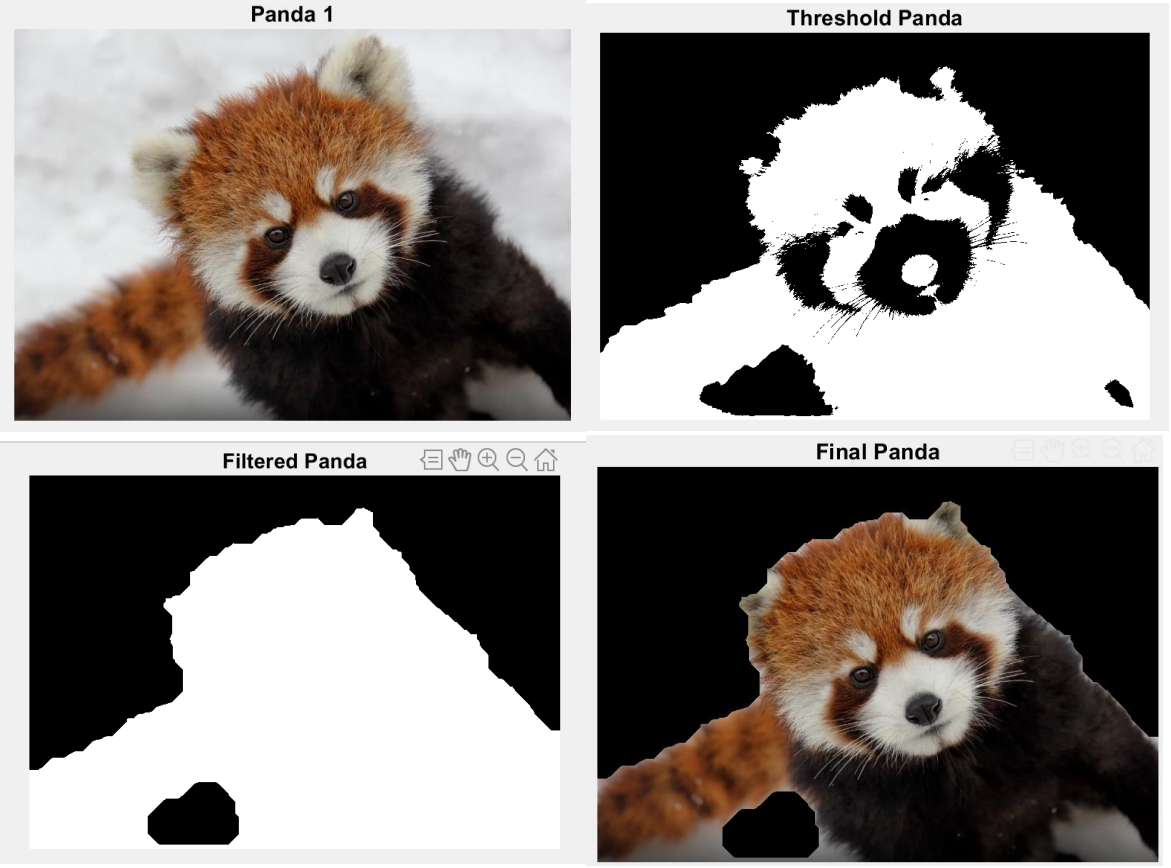
Next, after the one spectrum is selected, the thresholding part of the program begins. This differed with all four images, but you want to choose a threshold value anywhere between 150-225 based on the image and output of the RGB spectrum that you have selected. This will display a binary image of the single color spectrum image originally chosen. This new image has the purpose of trying to get a good frame of the red panda while also *trying* to eliminate the background.

To further eliminate background figures and objects, I've set up the morphological functions. First you select a desired kernel size based on the binary image that was created in the previous step. The goal is to choose a large enough kernel to delete some of the random background marks. This kernel is used in the eroding function for the binary image. After, that same kernel is used for dilating the image, but at the square of the kernel. So for example, if you choose a kernel size of 9, the image is eroded at that 9 kernel size, but the dilation occurs with a kernel of 9^2 or 81. The idea here is to eliminate small background marks while also trying to keep the panda's frame intact and fill it in with the huge squared kernel size.

Lastly, with all these steps, there should be an image outline of a general place on where the panda is thought to be. That swept binary image is then complemented to make the background 1 and make the panda region 0. The original panda image is then subtracted from this newly created background image to yield a region where the panda is, and hopefully, the entire panda itself with the background left out.

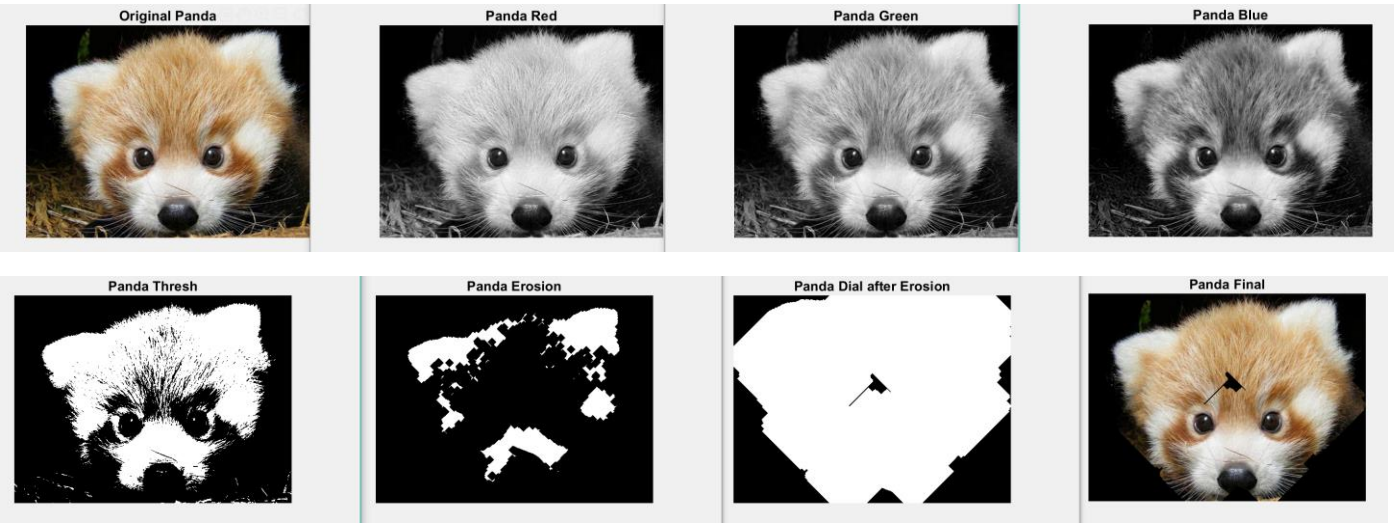
Panda Picture 1 (snow)

A very simple example to get started on. Because snow is such a prominent white, it reflects all colors in the RBG spectrum. I decided to go with the blue spectrum for this panda, and took the complement of the blue spectrum to get the most apparent color inverse. After, the image is eroded and dilated to encompass the whole panda's body and figure. After, the image is again complemented. The complement then selects all the background for the panda, and it can be subtracted from the original to yield just the panda in the picture.



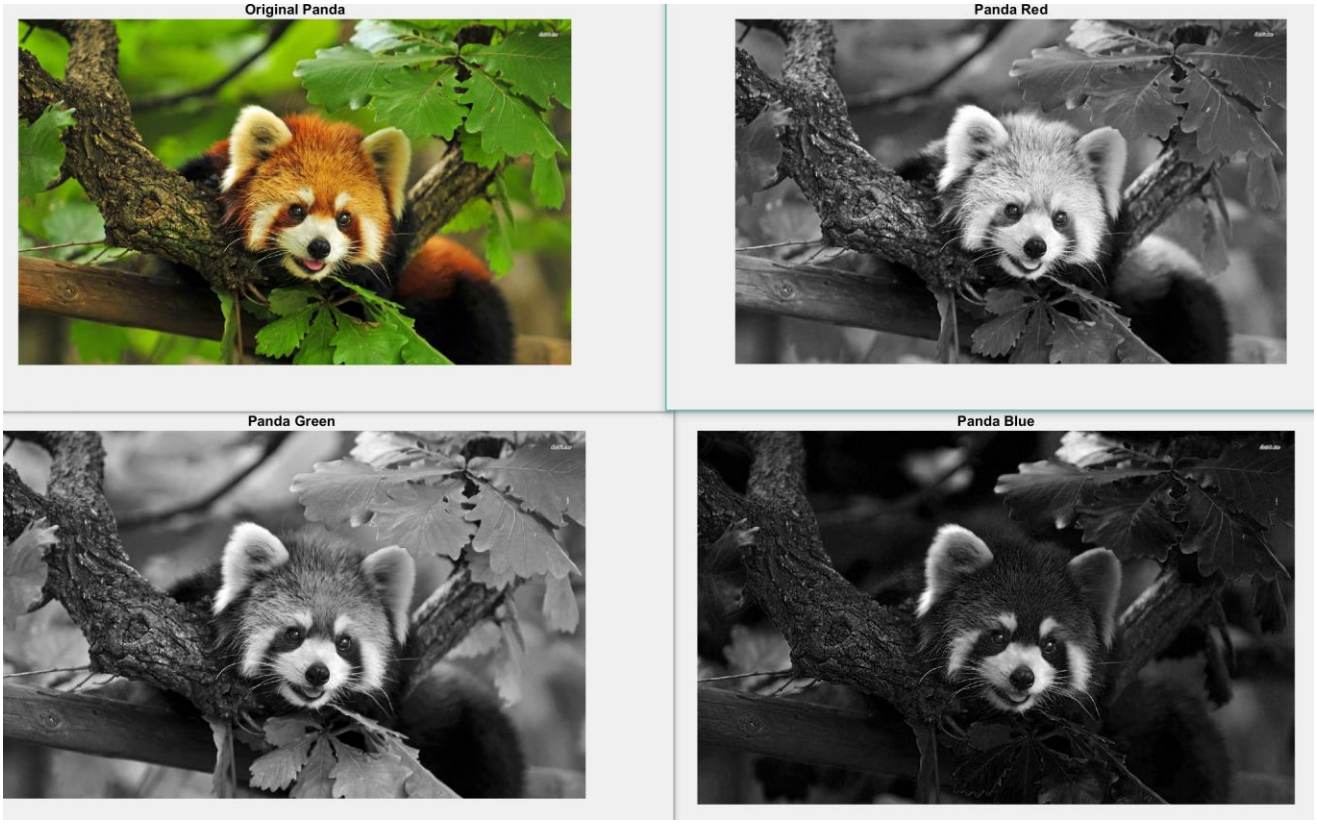
Panda Picture 2 (Baby)

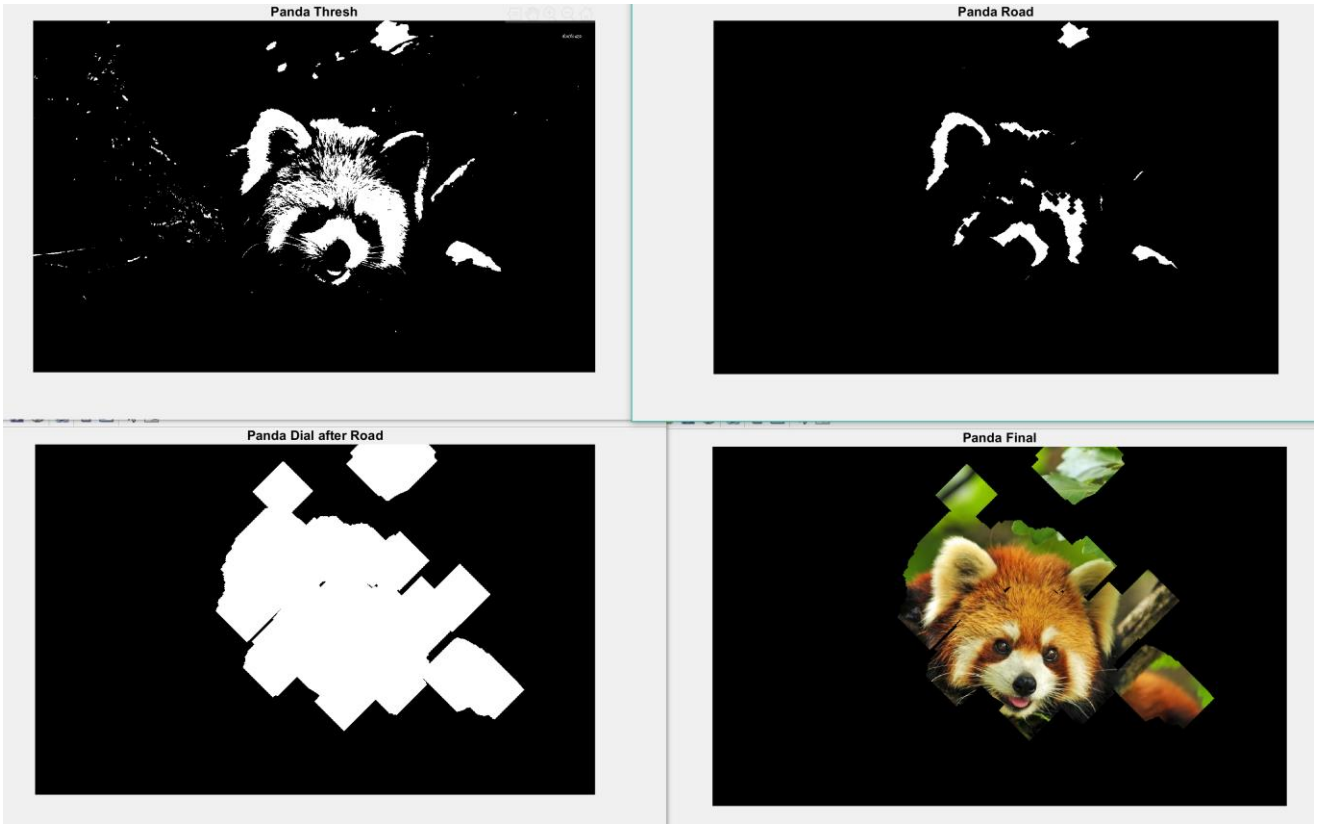
Next, this is another pretty basic picture, as the background is mostly already black and eliminated. For a black background situation, it's easiest to take the most prominent RGB spectrum for the panda (the red spectrum). With that, I take a spectrum of something around 180 to binarize the image. After the image is either a 1 or a 0 to signify the actual panda, I erode the image for any extra background obstructions leaving some of the red panda frame intact. The eroded image is then dilated to expand the frame of the red panda and fill in the face. After the erosion and dilation, the image is complemented to obtain the background, and the background is subtracted from the original to obtain the final image.



Panda Picture 4 (Happy)

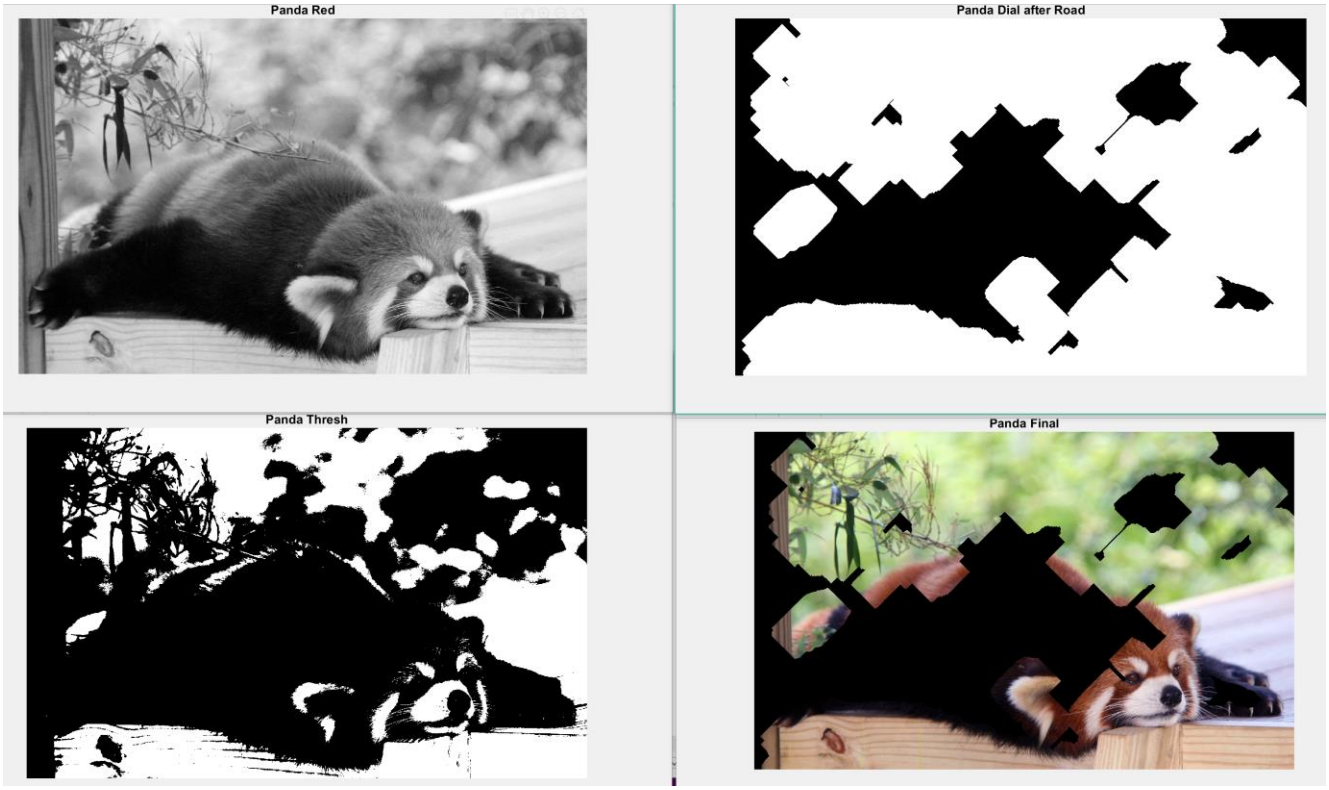
This is the part where the panda image selection starts to get obstructed by the color pallet of the actual environment. The multiple colors and different lighting angles combined with the way some parts of the leaves and trees reflect more or less light really interferes. This process is mostly identical to the previous one (Panda Picture 2) but with a bigger kernel structure.



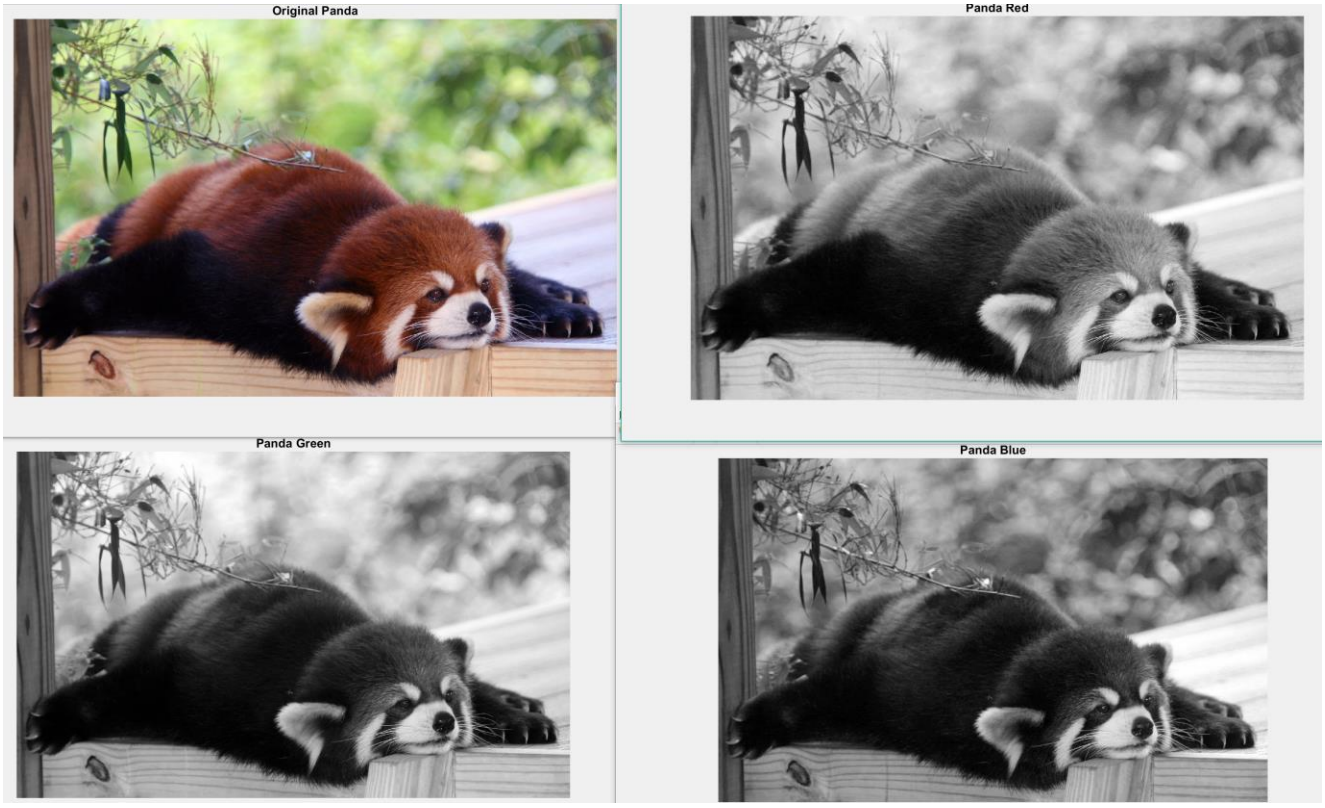

Panda Thresh
Panda Road
Panda Dial after Road
Panda Final

Panda Picture 3 (Tired)

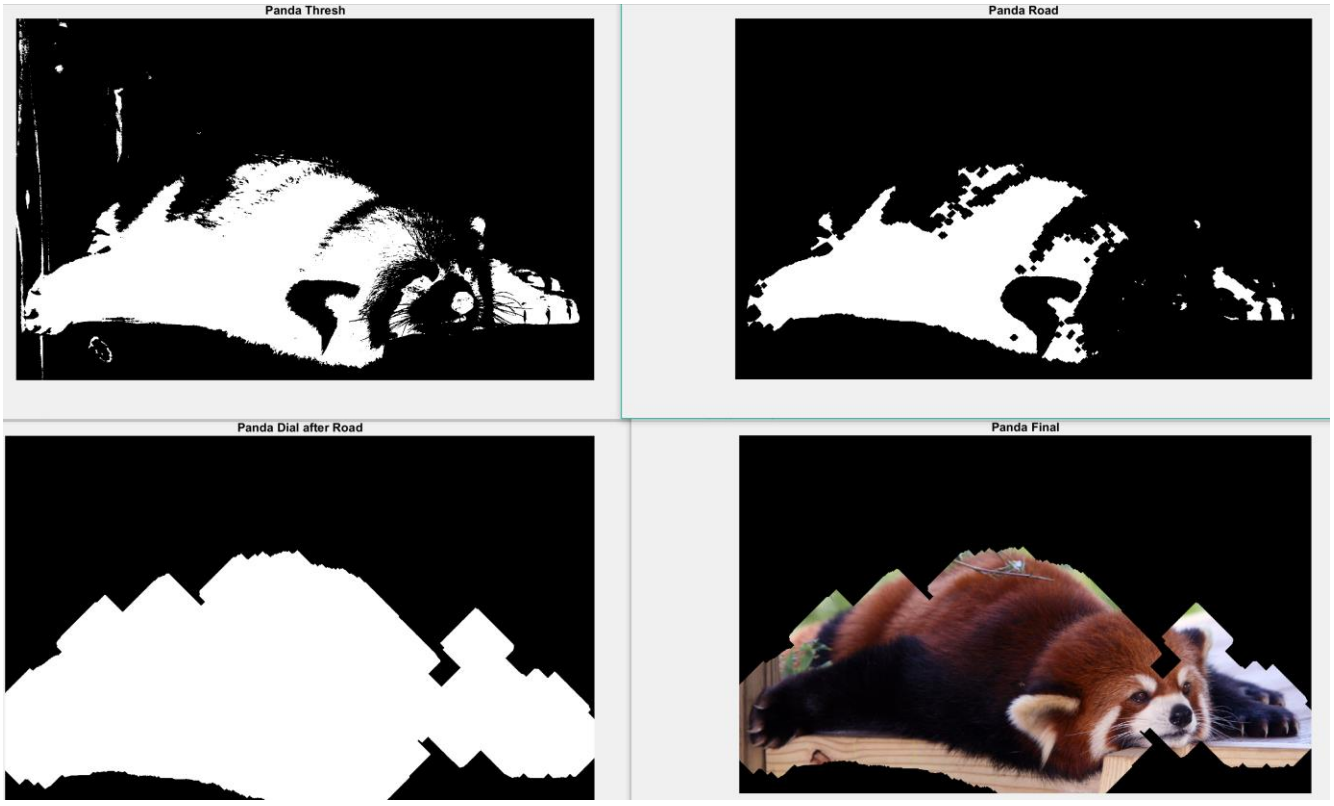
Where the algorithm breaks:



Instead of taking the red partition of the RGB spectrum and utilizing the red intensity of the red panda's natural fur color, I can take the lack of green in the fur and pair it against the green background. This is more similar to the first image set, where the panda is the darker image against the white background of snow. I took the compliment of the panda to select it instead of the background, and again, used erosion and dilation.



Thresholding off of the complement of the green spectrum of the RBG band:



Conclusion

Ultimately, because of the red panda's fur color differentiation between white and black, it makes it difficult to take the complete panda off from the background. It also requires a bit of manual work instead of working entirely autonomously. Instead of using the RGB spectrum to analyze the pictures, I thought to take a blurred version of the image and subtract it from the original image. In theory, because of the fine level of detail in the red panda's fur in contrast with some of the background, it would create an outline of the red panda, then the rest could be filled in. However, this approach often left leaves and other natural environmental structures in the images occasionally. So I went with the color approach because I thought this could have the greatest, most straightforward path to success. Even though the process is somewhat manual and requires certain values to be adjusted for each image, the basis for the code is there and can be adjusted to fit the image's need to pull out a particular object within an image with a slightly different color contrast from that of its background.