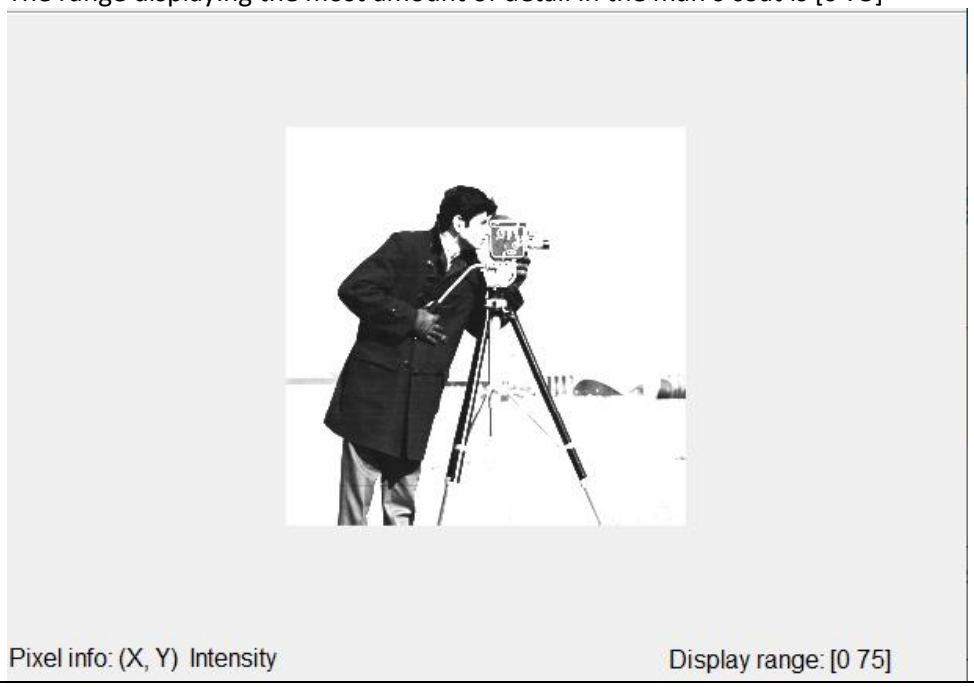
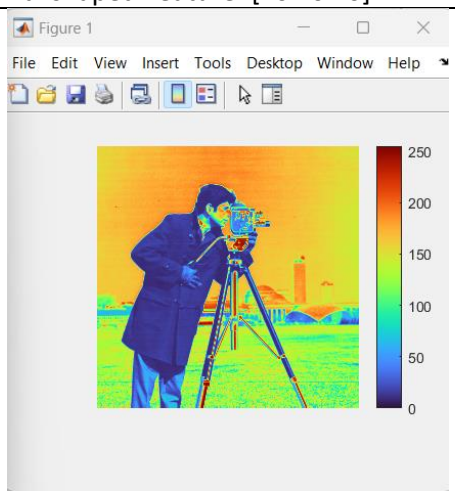
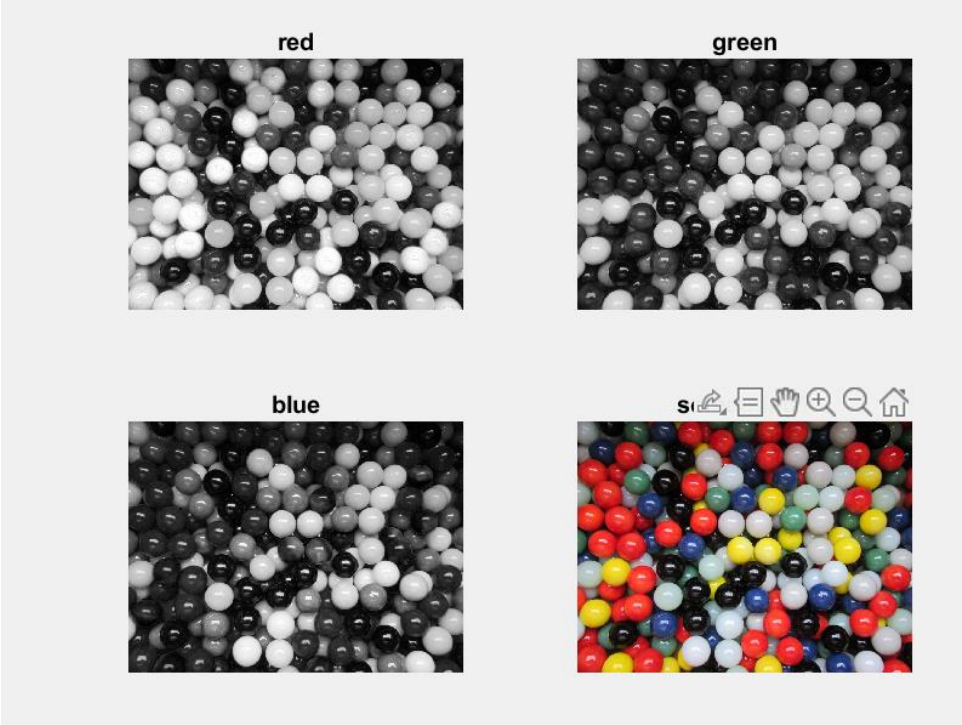
Name: Luis Hernandez Aguirre CM
Name: Fox Warner CM

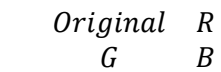
Lab 4: Color Image Analysis – Worksheet

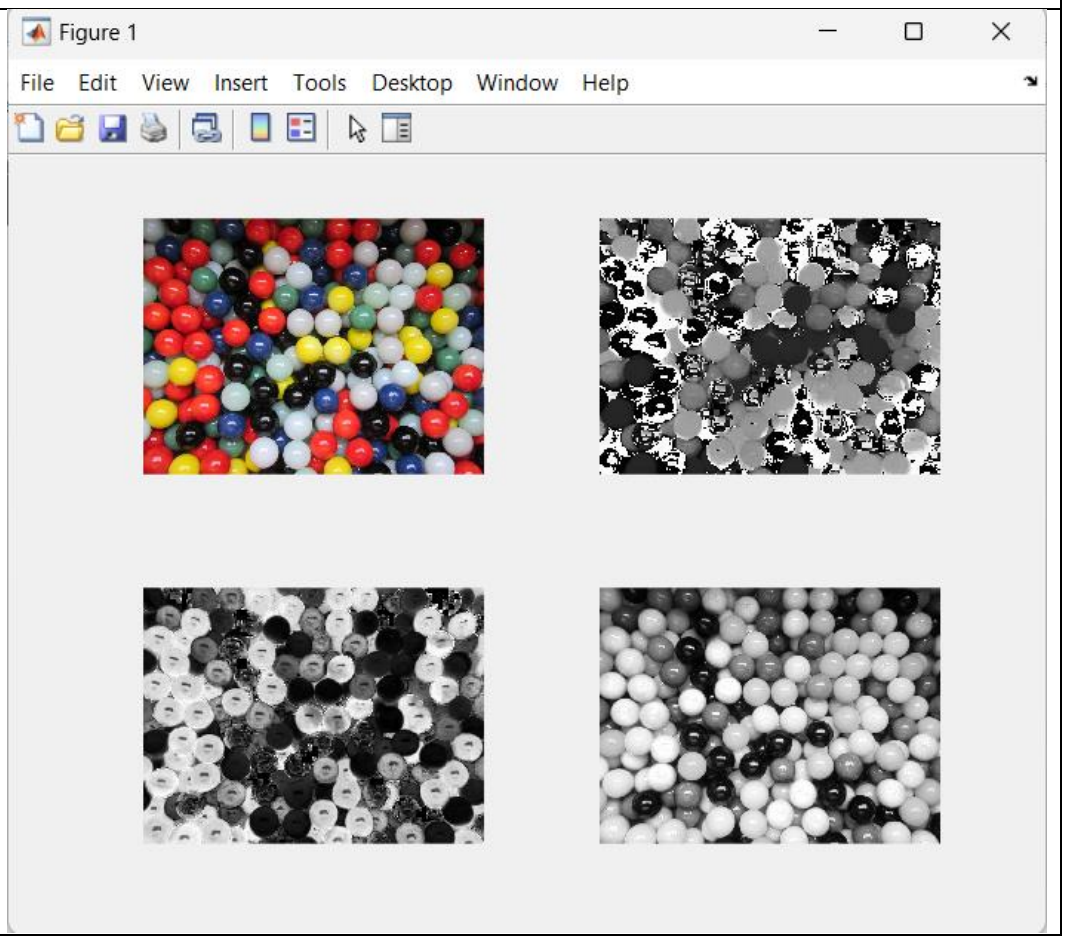
ECE180: Introduction to Signal Processing

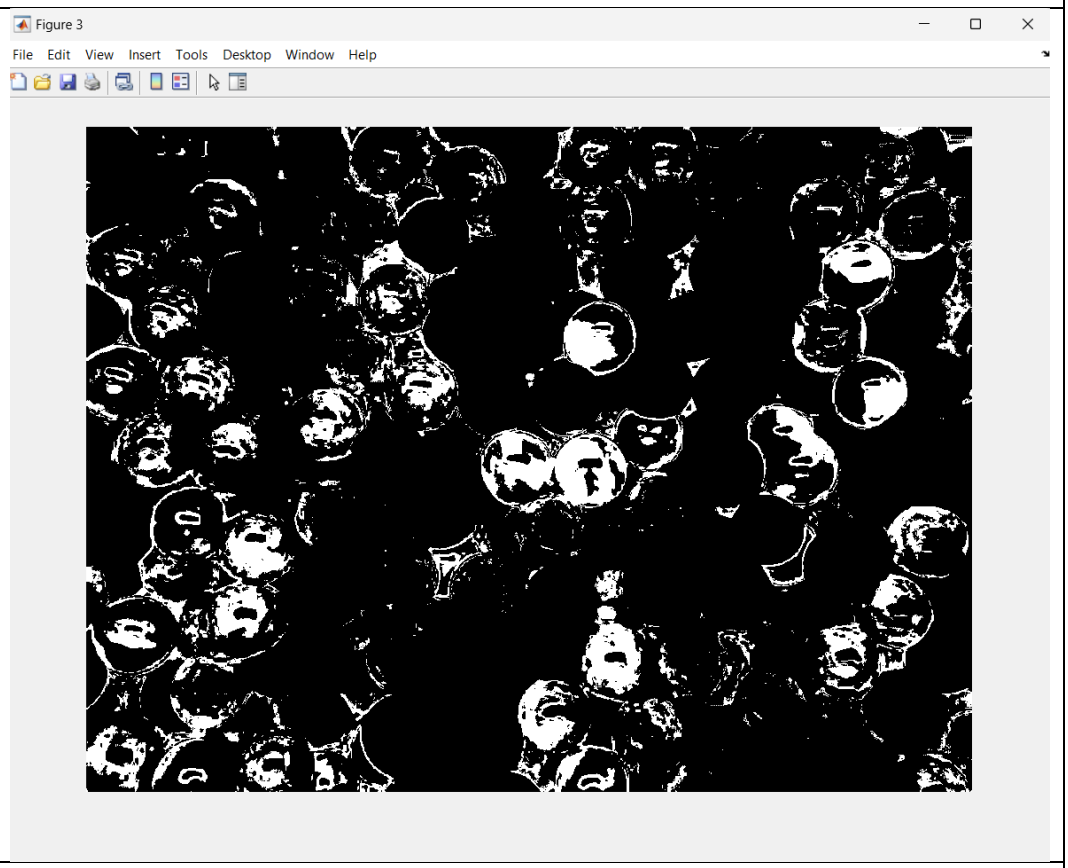
1.1	Width: 256 Height: 256 BitDepth: 8 ColorType: ‘grayscale’
1.2	Width: 640 Height: 464 BitDepth: 8 ColorType: 'indexed'
1.3	Width: 768 Height: 512 BitDepth: 24 ColorType: 'truecolor'
2.4(a)	The first two values are the x,y coordinates of the pixel being hovered over. The third is the intensity of the pixel.
2.4(b)	The top left corner is the origin
2.4(c)	The increasing y is down
2.4(d)	the minimum was 8. It was on the darkest spots on the screen, the man’s coat and hair
2.4(e)	the max was 251, it was located on the sun glare off of the tripod’s leg
2.5	They both allow you to zoom in further in the image. The inpsect pixel value tool allows you to zoom in further, but also divides the image into a grid of # height pixels, x # of width pixels. And within each square lies in the intensity.
2.6(a)	The data range is [7, 253].
2.6(b)	We were close being 1 off the minimum and 2 off the maximum
2.7(a)	The most common are intensity values 9 and 12
2.7(b)	The man’s coat and hair as well as the black parts of the tripod legs
2.8(a)	The pixels with intensities over the specified upper limit become black. As a result, as the upper limit decreases and less pixels are within that range, the image becomes more white

2.8(b)	The pixels with intensities under the specified lower limit become black. As a result as the lower limit increases and less pixels are within that range, the image becomes more black
2.8(c)	The range displaying the most amount of detail in the man's coat is [0 75] 
3.3	Sky: [100 130 150] Roof of connecting Building: [130 60 50] Side of light Tower: [170 165 140] Ballshaped Feature: [40 40 40]
4.4	 Color map : Turbo
5.2	It is rotated 90 degrees clockwise and flipped on the y axis
6.5(a)	`function [imageData] = showplanes(imageData)` `% Plots the data of an image, and the r plane, g plane, and b plane`

	`` for k = 1:4 subplot(2,2,k); if k == 1 % plots entire image imshow(imageData); else % plots each plane data = imageData(:,:,k-1); imshow(data); end end ```
6.5(b)	

	`` function showplan(img) subplot(2,2,1); imshow(img(:,:,1)); title('red') subplot(2,2,2); imshow(img(:,:,2)); title('green') subplot(2,2,3); imshow(img(:,:,3)); title('blue') subplot(2,2,4); imshow(img) title('something?') end ``` They are plotted as such;  $Original \quad R$ $G \qquad B$
6.6	In the r plane, all of the red marbles are white. The same goes for green marbles in the g plane and blue marbles in the b plane however the red plane shows its color the most. Green also shows its color as a whiter color, however, there are other light colors that make it hard to distinguish. Blue still shows its color as kind of white however it looks very similar to the green color and is very hard to distinguish.
6.7	The yellow marbles are white within the r and g plane. They are somewhat visible in the b plane. The threshold would be high red, high green, mid blue [200, 200, 125]
6.9(a)	The original image is uint8 The individual planes are doubles.
6.9(b)	2250000 bytes or 2.25 * 10^6 bytes for the image
6.9(c)	6000000 bytes or 6 * 10^6 bytes per plane Storing the entire image vs the three panes saves 8x as much space.

6.10(a)	No, each of the planes become pure black
6.10(d)	`function [h,s,v] = showHSVplanes(imageData)` `%UNTITLED2 Summary of this function goes here` `new = uint8(255 * rgb2hsv(imageData));` `for k = 1:4` `  subplot(2,2,k);` `  if k == 1 % plots entire image` `  imshow(imageData);` `  else  % plots each plane` `  data = new(:,:,k-1);` `  imshow(data);` `  end` `end` `h = new(:,:,1);` `s = new(:,:,2);` `v = new(:,:,3);`
6.10(e)	

6.11	Yes, within a certain value range in each plane the yellow marbles will remain unchanged
6.12	[1 50]
6.13	[125 210]
6.16(a)	`featH = h>= 0 & h<=50;` `featS = s>= 125 & s<=210;`
6.16(b)	
6.16(c)	Fairly well, we focused on the yellow marbles in the center of the images. They pop out vividly, but we could've gone narrower with the saturation. range
6.17(a)	Logical
6.17(b)	It determines if the value is within the specified range and show it if it is.
7.6	`% Luis Antonio Hernandez Aguirre & Fox Warner` `% ECE180 Lab 4` `% 26 September 2023` `a = imread('marbles.png');` `p = 600;`

	[h, s, v] = showHSVplanes(a); featH = h>= 0 & h<=50; featS = s>= 140 & s<=200; feat = featS & featH; feat = bwareaopen(feat, p); feat = bwconvhull(feat, 'objects'); % figure; % imshow(feat); cc = bwconncomp(feat); rp = regionprops(cc); for k=1:length(rp) a=insertShape(a,'Rectangle',rp(k).BoundingBox); a=insertText(a,rp(k).Centroid,num2str(k),'AnchorPoint','Center'); end figure imshow(a)
7.7