

Color detection

To do the color detection in an image we first had to read the image as so:

```
int rgb = bufferedImage.getRGB(x,y);
```

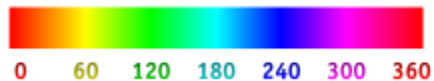
We obtained the data buffer from the image and directly obtained the pixels as an int[] array.

After that we did a for loop to go over every pixel testing what color it is but we encountered a problem because every pixel isn't perfectly red blue or green in most of the cases so we couldn't test the pixels as so: ~~rgb == Color.BLUE.getRGB()~~

Instead, we analyzed the hue of the RGB value, by converting it into HSB color space.

```
int rgb= bimg.getRGB(x,y);  
  
float hsb[] = new float[3];  
  
int r = (rgb >> 16) & 0xFF;  
int g = (rgb >> 8) & 0xFF;  
int b = (rgb << 8) & 0xFF;  
  
Color.RGBtoHSB(r, g, b, hsb);
```

Then the hsb array will contain the hue, saturation and brightness of the color. These are values between 0 and 1. The hue multiplied by 360 will give the tone of the color:



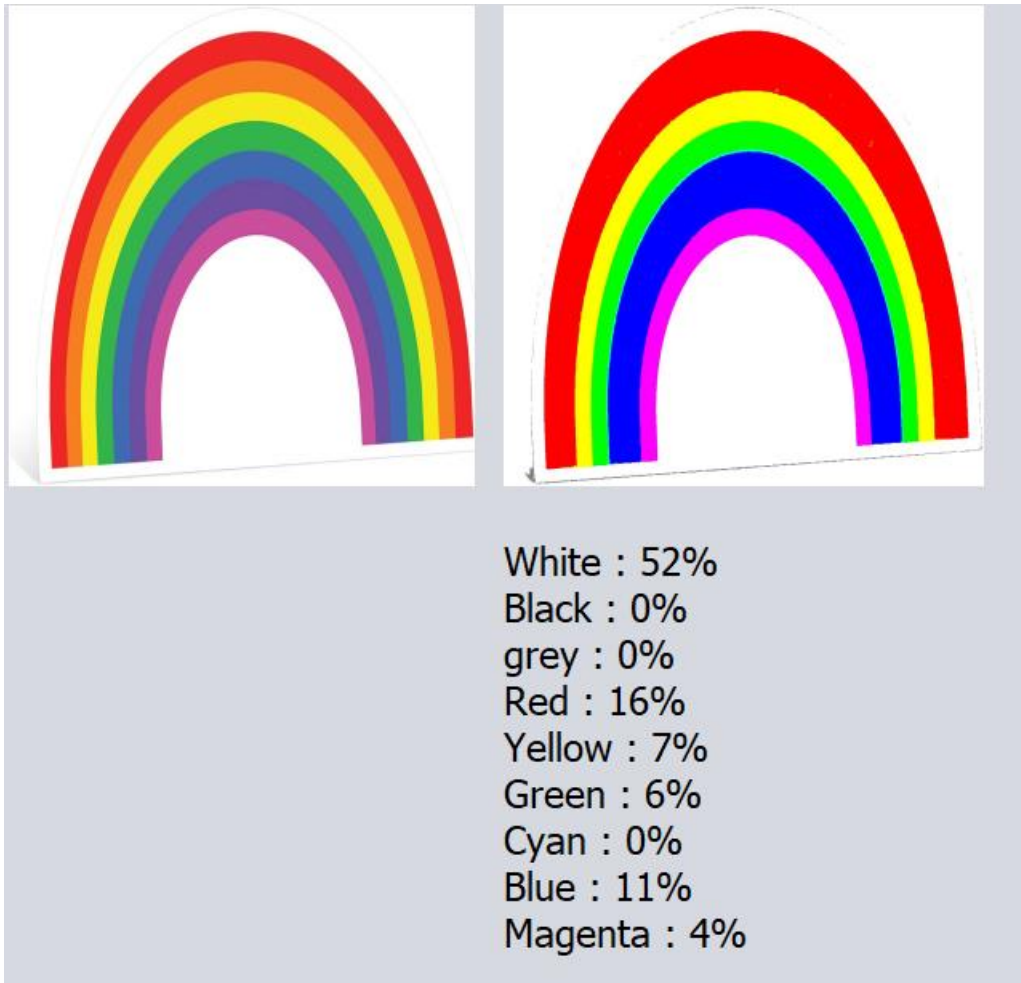
The saturation and brightness can be used to detect pixels that are nearly black or nearly white.

And we classified every pixel with r=g=b as grey.

We initialized variables for each color equal to zero: white, black, grey, yellow, green, cyan, blue, magenta and red; and a variable called "whole" as the number of pixels in the image.

On each pixel we tested which color it is and incremented the corresponding variable value by 1.

At the end as simple as math we multiplied each color variable by 100 and divided it by the variable whole to get the percentage of that color in all the picture and displayed the result on the screen.



For the screen shot I replaced each pixel by the color that was detected to test if the result was accurate and as we can see the dark purple was classified as blue and the orange as red.