Vogel_Max_ImageProcessing

This is my livescript for the image processing assignment!

Author: Max Vogel

```
x=5 % prints output
```

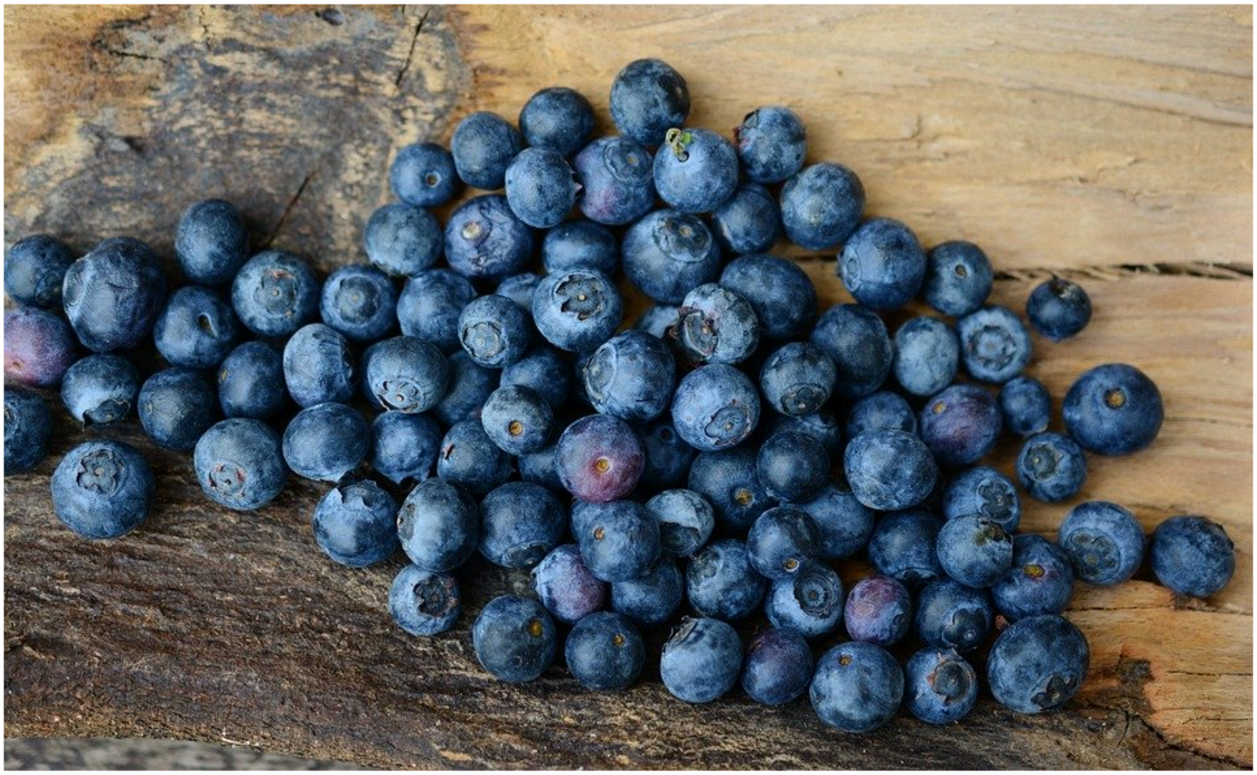
```
x = 5
```

```
y=3; % semicolon doesn't print output
x*y % if not assigned to variable, saves to ans
```

```
ans = 15
```

Importing the Image

```
url = "https://cdn.pixabay.com/photo/2017/04/29/09/51/blueberries-2270379_960_720.jpg";
img = imread(url); % pulls image from URL
imshow(img) % displays the image
```



Color space transformations

We'll convert the image to HSV format.

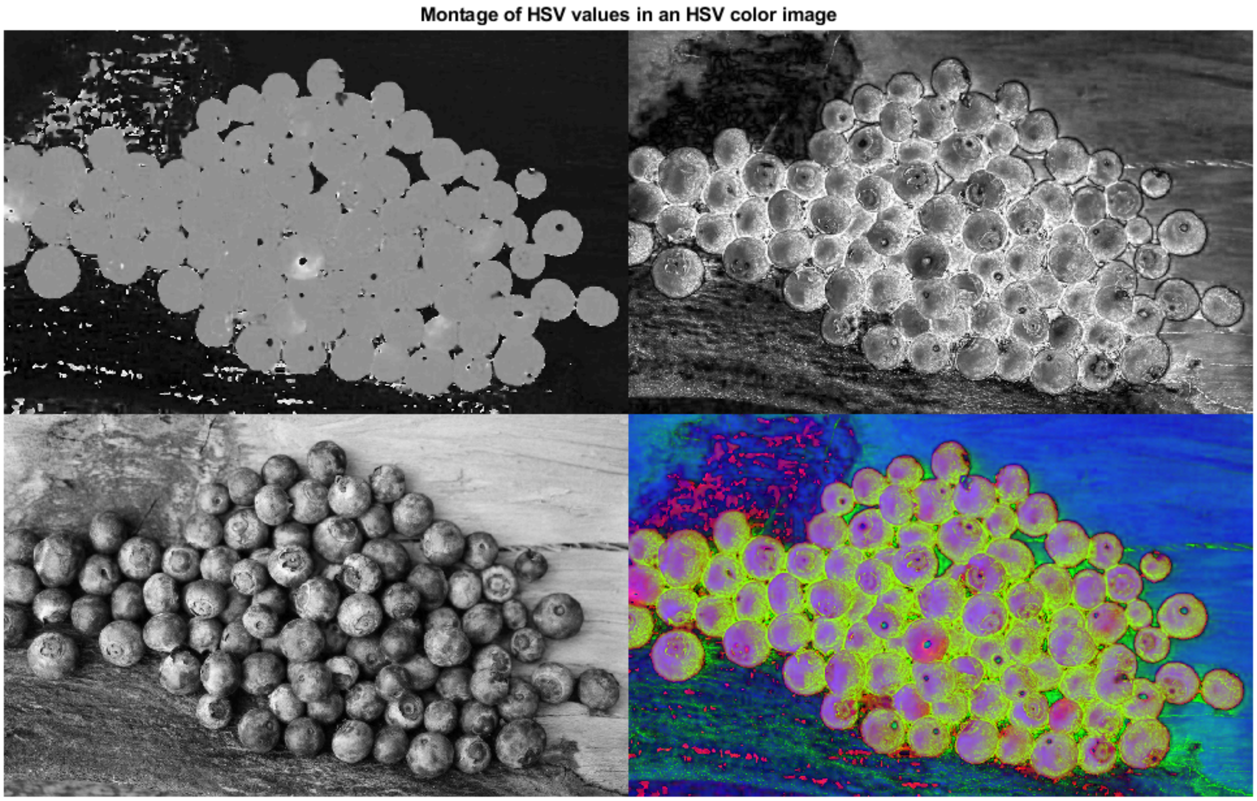
```
hsvImage = rgb2hsv(img);
```

Then we'll split our HSB image into the hue, saturation, and value (brightness) channels with imsplit.

```
[h,s,v] = imsplit(hsvImage);
```

We can then compare each different channel via a montage.

```
montage({h,s,v,hsvImage})
title('Montage of HSV values in an HSV color image')
```



Using `findcircles` to find blueberries

We'll start by selecting the s (saturation) channel. This channel has the most defined edges, and through (my) experimentation has worked the best.

```
X=s;
```

We'll then us imfindcircles to find circles within our image. We'll pass `X` as the image to look at, `[22 39]` as the radius range, and various parameters to refine our scope. We'll then define the output of this function to `[centers,radii,metric]`. `centers` is a P-by-2 matrix that contains the coordinates of the circle centers. `radii` is a column vector with the estimated radii of each circle. `metric` is a column vector with the confidence for each circle.

```
[centers,radii,metric] = imfindcircles(X,[22 39], ...
    'ObjectPolarity','bright', ...
    'Sensitivity',.98, ...
    'EdgeThreshold',0.26, ...
    'Method','TwoStage');
```

We'll then filter so only the circles with a confidence greater than `0.035` are selected. Once we have this range, we'll use it to filter the corrisponding `center`, `radii`, and `metric` values.

```
range=1:length(radii(metric>0.035));
centersStrong = centers(range,:);
radiiStrong = radii(range);
metricStrong = metric(range);
```

We can then show our resulting circles ontop of our original image!

```
imshow(img);
viscircles(centersStrong, radiiStrong,'EdgeColor','r');
title('Highest metric circles')
```