Homework Assignment 1
Computer Vision for HCI
Prof. Jim Davis
TA: Sayan Mandal

Anirudh Ganesh
CSE5524 (Au '18)
Score: ___/7
Due Date: 08/28/18

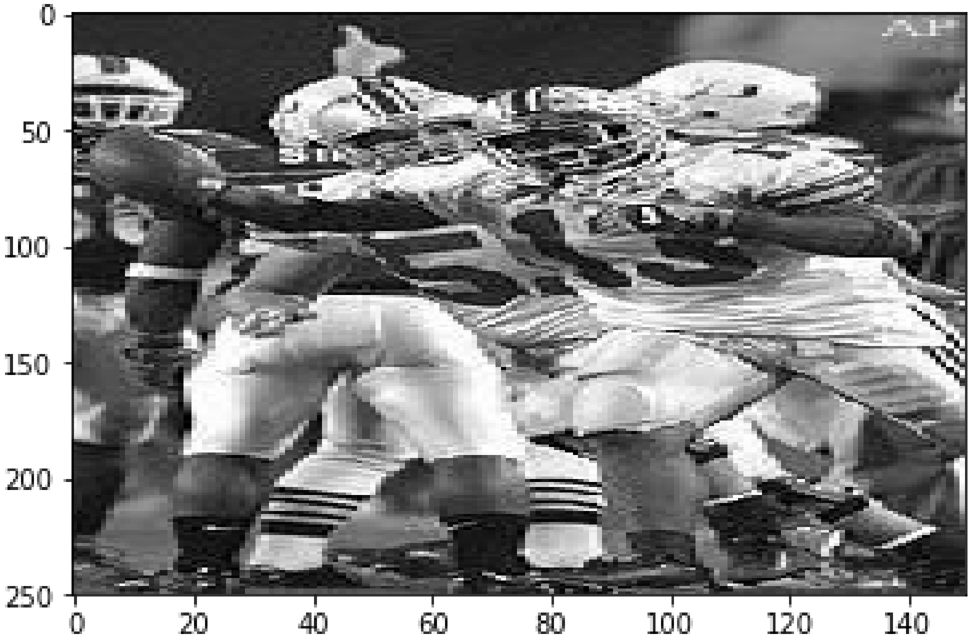
Importing libraries

```
In [1]: from skimage.io import imread
        from skimage.color import rgb2gray
        import numpy as np
        from matplotlib import pyplot as plt
```

1. Test the MATLAB image functions to read, display, and write images. Use buckeyes_gray.bmp and buckeyes_rgb.bmp from the class webpage

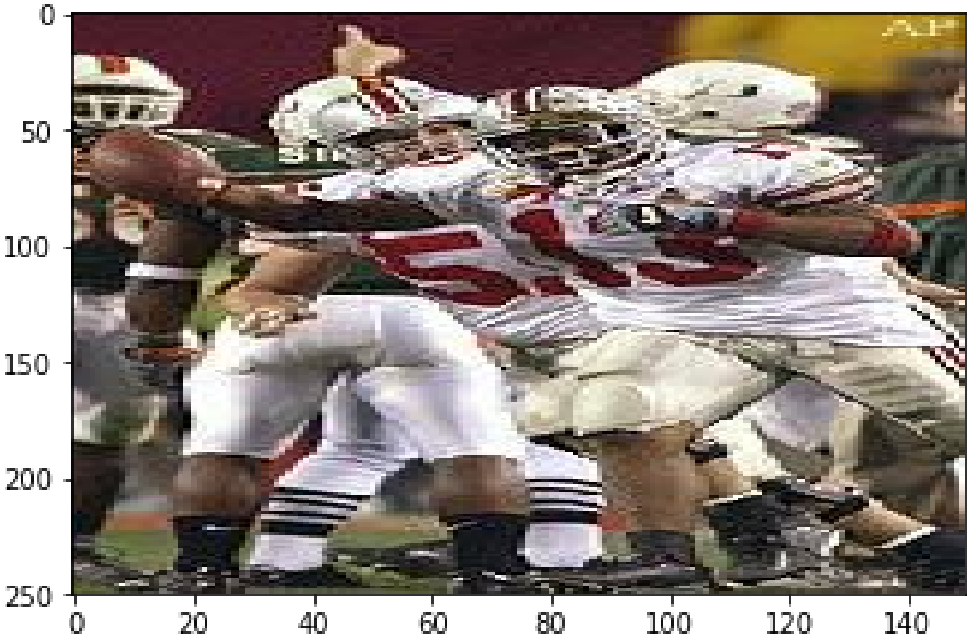
```
In [2]: grayIm = imread('buckeyes_gray.bmp')
        plt.imsave('output/buckeyes_gray.jpg', grayIm, cmap= plt.get_cmap('gray'))
        plt.imshow(grayIm, aspect='auto', cmap = plt.get_cmap('gray'))
```

```
Out[2]: <matplotlib.image.AxesImage at 0x1f8d37172b0>
```



```
In [3]: rgbIm = imread('buckeyes_rgb.bmp')
        plt.imsave('output/buckeyes_rgb.jpg', rgbIm)
        plt.imshow(rgbIm, aspect='auto')
```

```
Out[3]: <matplotlib.image.AxesImage at 0x1f8d36ac908>
```



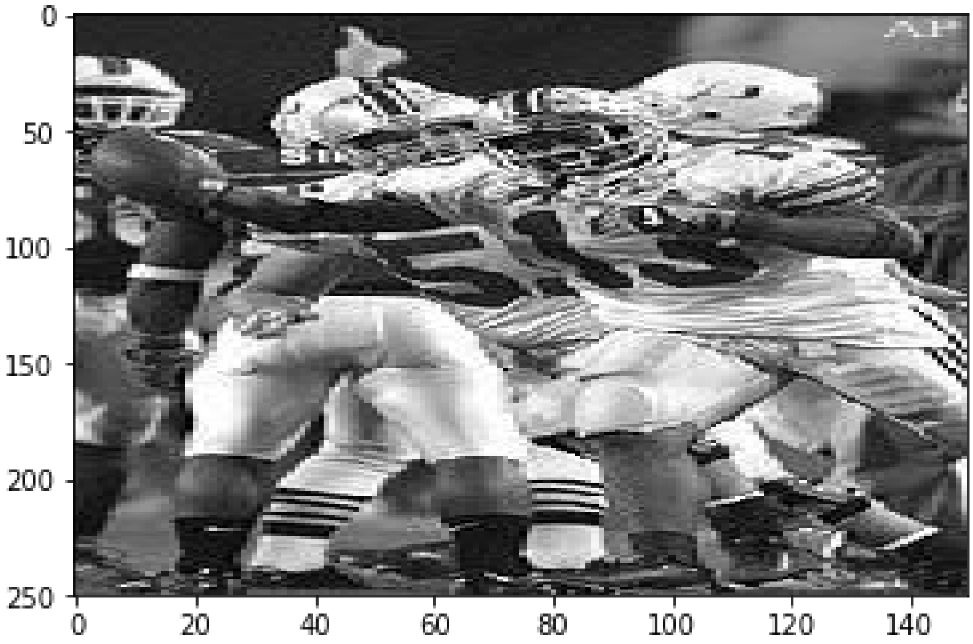
Q. Read and convert the rgb image to grayscale using the NTSC conversion formula via the MATLAB function rgb2gray. Display your image to verify the result The NTSC Conversion formula is given by

$$intensity = 0.2989 * red + 0.5870 * green + 0.1140 * blue$$

These values have be derived experimentally to match the human cognitive biases regarding colours.

```
In [4]: grayIm_converted = rgb2gray(rgbIm)
          plt.imsave('output/buckeyes_gray_converted.bmp', grayIm_converted, cmap = plt.get_cmap(
          plt.imshow(grayIm_converted, aspect='auto', cmap = plt.get_cmap('gray'))
```

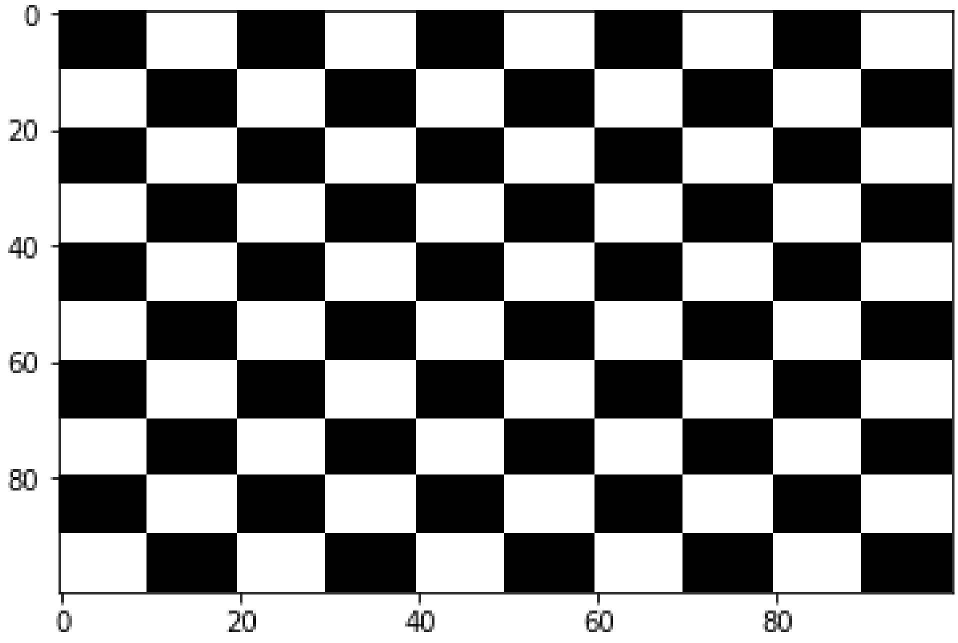
```
Out[4]: <matplotlib.image.AxesImage at 0x1f8d4e9ce48>
```



Q. Test more fully by creating, writing, and reading a checkerboard image

```
In [5]: zBlock = np.zeros((10,10))
         oBlock = np.ones((10,10))*255

         pattern = np.block([[zBlock,oBlock], [oBlock,zBlock]])

         checkerIm = np.tile(pattern, (5,5))

         plt.imsave('output/checker.bmp', checkerIm, cmap = plt.get_cmap('gray'))

         plt.imshow(checkerIm, aspect='auto', cmap = plt.get_cmap('gray'))
```

```
Out[5]: <matplotlib.image.AxesImage at 0x1f8d4ccf2b0>
```


0
20
40
60
80
0
20
40
60
80