

**1) Test the MATLAB image functions to read, display, and write images. Use buckeyes\_gray.bmp and buckeyes\_rgb.bmp from the class materials webpage. [2 pts]**

This problem mainly uses `imwrite()` function to write images to graphics files, with `jpg` extension.

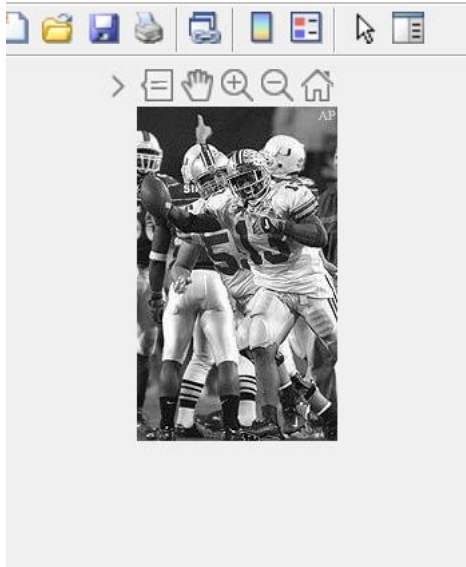
The first group uses `buckeyes_gray.bmp` to generate `bucyeyes_gray.jpg`, which is a grayscale image. The second group uses `buckeyes_rgb.bmp` to generate `buckeyes_rgb.jpg`, which is a full color image.





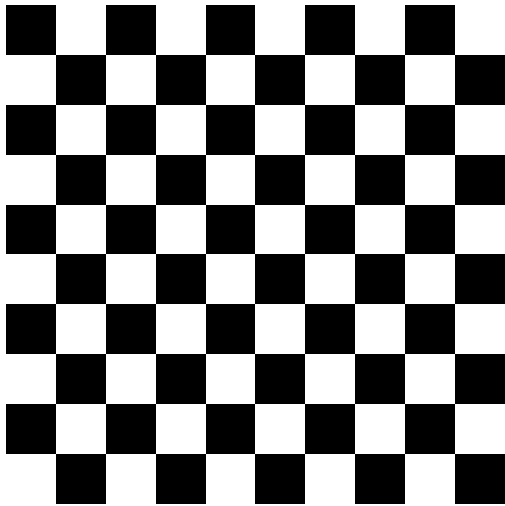
- 2) Read and convert the rgb image to grayscale using the NTSC conversion formula via the MATLAB function `rgb2gray`. Display your image to verify the result. [1 pt]

In this question, I use `rgb2gray(RGB)`, which can Read the RGB image, and then convert it to grayscale. The result is shown as a file named `HW1 Q2.fig` in the folder, which is the same as the first question's outcome.



**3) Test more fully by creating, writing, and reading a checker-board image. [2 pts]**

With the `repmat()` and `imwrite()` function, we generated a 10-by-10 checker board image, then use `colormap('gray')` to generate a grayscale checker board image, that's what we saw below:



## Code:

```
% Yukun Duan
% CSE5524 - HW1
% 8/25/2022

%% Problem 1

grayIm = imread('buckeyes_gray.bmp'); %read the image from the file
imagesc(grayIm); %display the data in bmp
axis('image');
colormap('gray');
imwrite(grayIm, 'buckeyes_gray.jpg'); %write gray image to jpg file

%%

rgbIm = imread('buckeyes_rgb.bmp'); %read the image from the file
imagesc(rgbIm);
axis('image');
imwrite(rgbIm, 'buckeyes_rgb.jpg'); %write rgb image to jpg file

%% Problem 2

grayIm = rgb2gray(rgbIm); %convert RGB images to grayscale
imshow(grayIm); %display image

%% Problem 3

zBlock = zeros(10,10);
oBlock = ones(10,10)*255;
pattern = [zBlock oBlock; oBlock zBlock];
checkerIm = repmat(pattern, 5, 5); %generate a 25 set 10-by-10 array
imwrite(uint8(checkerIm), 'checkerIm.bmp'); %generate bmp image file
Im = imread('checkerIm.bmp');
imagesc(Im)
colormap('gray') %convert image to grayscale
axis('image');
```