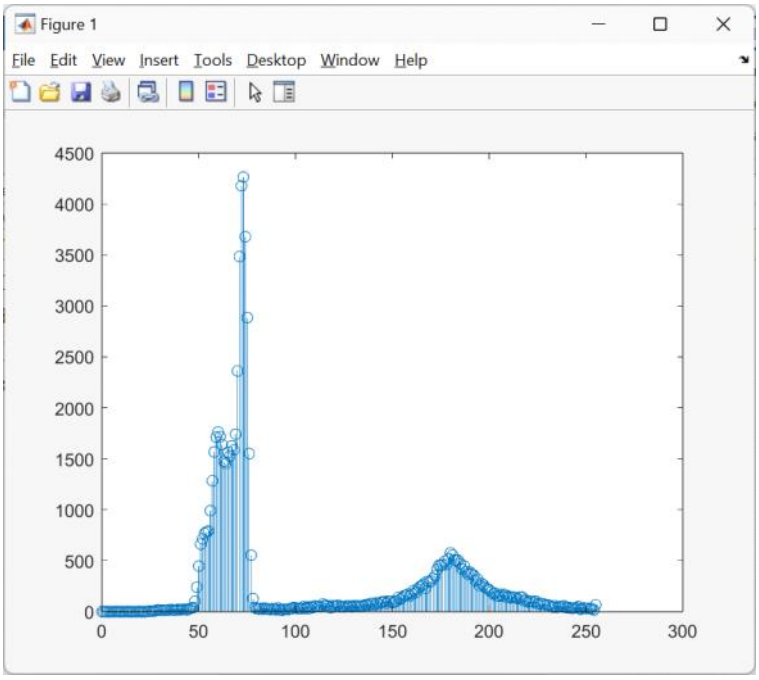
Lecture 2

Imhist function (cont'd)

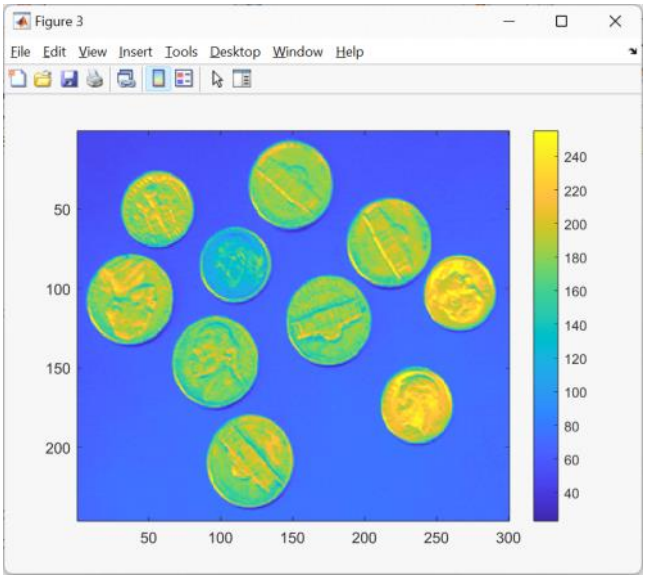
[counts,binLocations] = imhist(I) calculates the histogram for the grayscale image I. The imhist function returns the histogram counts in counts and the bin locations in binLocations. The number of bins in the histogram is determined by the image type.

```
>> [c, b] = imhist(I);
>> stem(b, c)

>> c(1:10)'

ans =

   0   0   0   0   0   0   0   0   0   0

>> b(1:10)'

ans =

   0   1   2   3   4   5   6   7   8   9
```

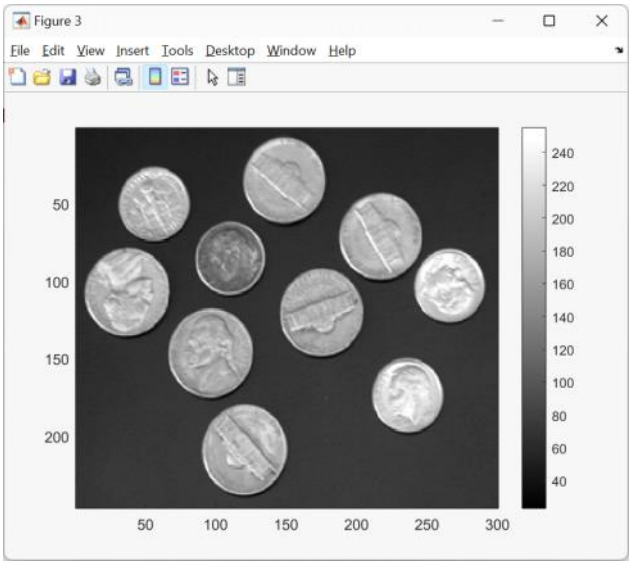


- imagesc function

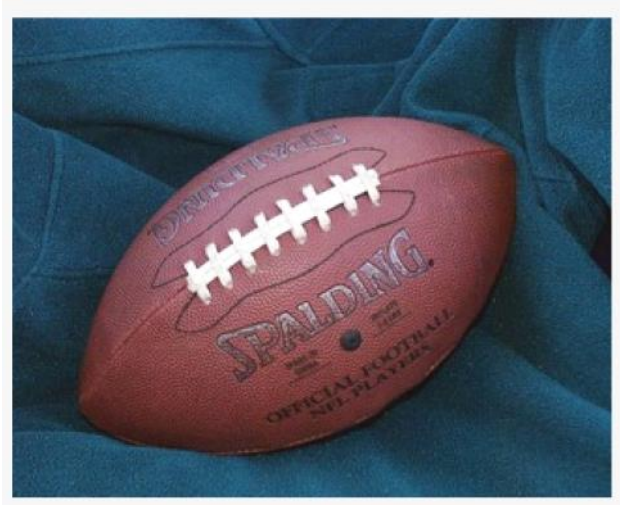
```
>> figure; imagesc(I)
>> colorbar
```



```
>> colormap('gray')
```



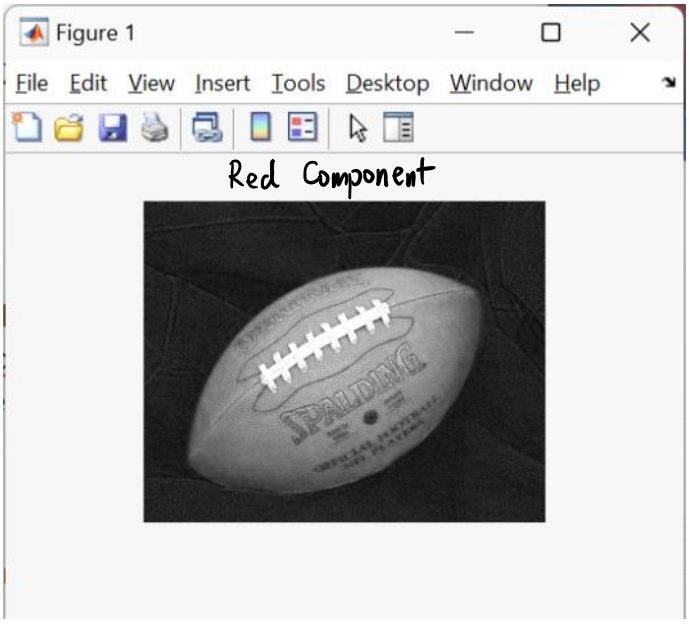
- Color Images



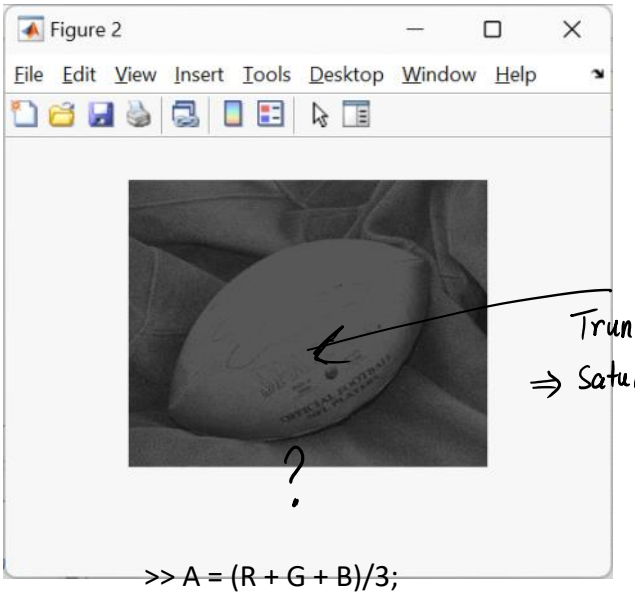

```
>> I = imread('football.jpg');
>> imtool(I)

>> whos I
  Name        Size                Bytes  Class
Attributes

  I         256x320x3            245760  uint8
```

R, G, B Components

```
>> R = I(:,:,1);
>> whos R
  Name        Size                Bytes  Class    Attributes

  R         256x320               81920  uint8

>> figure; imshow(R)
```



```
>> G = I(:,:,2);
>> B = I(:,:,3);
>> whos G
  Name        Size              Bytes  Class    Attributes

  G         256x320             81920  uint8

>> whos B
  Name        Size              Bytes  Class    Attributes

  B         256x320             81920  uint8
```




```
>> R(146,146) + G(146,146) + B(146,146)

ans =

  uint8

   255
```

```
>> R(146,146)

ans =

  uint8

   185
```

```
>> G(146,146)
ans =
  uint8
   119
```

```
>> B(146,146)
ans =
  uint8
   129
```

```
>> (185 + 119 + 129)

ans =

   433
```

⇒ Truncated to 255 (uint8 type)

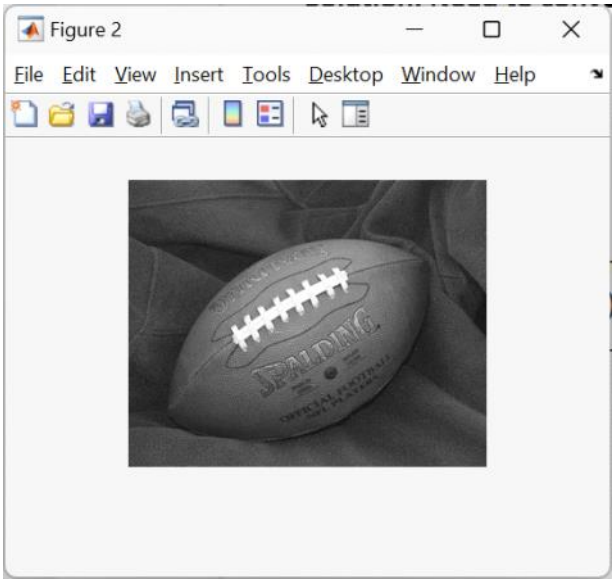
```
>> A = (R + G + B)/3;

>> A(146,146)
ans =
  uint8
   85
```

Solution: Need to convert uint8 type to the double type for arithmetic operations.

```
>> R = ID(:,:,1); G = ID(:,:,2); B = ID(:,:,3);
>> AD = (R + G + B)/3;
>> AD (146, 146)
ans =
  144.3333
```

Convert the result back to uint8 type

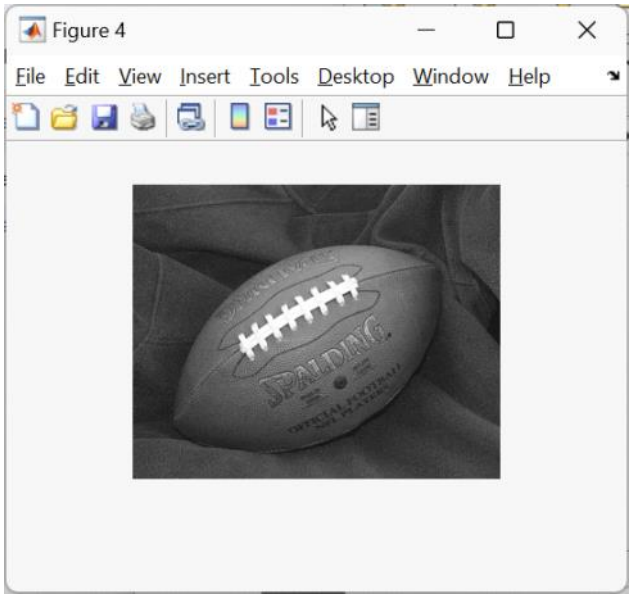
```
>> AD_uint8 = uint8(AD);
>> figure; imshow (AD_uint8)
```



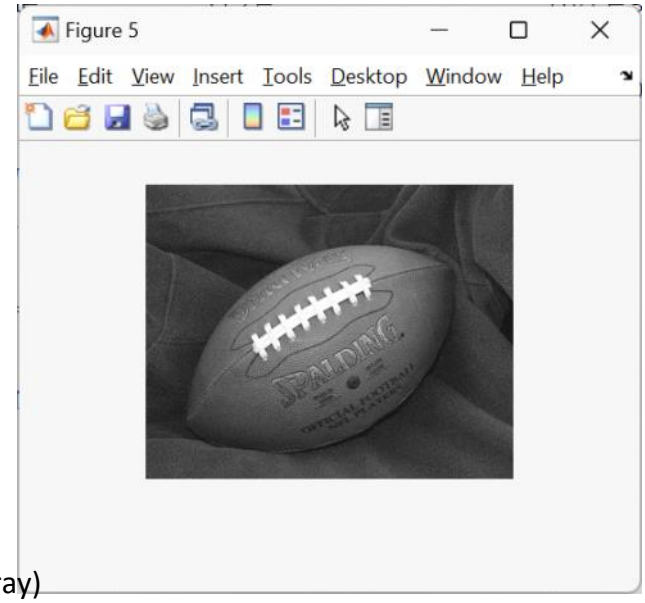
rgb2gray converts RGB values to grayscale values by forming a weighted sum of the *R*, *G*, and *B* components:

0.2989 * R + 0.5870 * G + 0.1140 * B

```
>> I_rgb2gray = rgb2gray(I);
>> figure; imshow(I_rgb2gray)
```




```
>> AW = 0.2989 * R + 0.5870 * G + 0.1140 * B;
>> whos AW
  Name      Size            Bytes  Class    Attributes
  AW        256x320         655360  double

>> AW_uint8 = uint8(AW);
```



identical

```
>> isequal(AW_uint8, I_rgb2gray)

ans =
  logical
   1
```

- Size of the JPG image file

  On the hard drive: 27,130 football.jpg

```
>> whos I
  Name      Size            Bytes  Class    Attributes

  I         256x320x3       245760  uint8
```

Raw Image Size

Raw image was compressed from 245760 bytes to 27130 bytes with a compression ratio about 9:1

```
>> 245760/27130

ans =
```

9.0586

https://hexed.it/