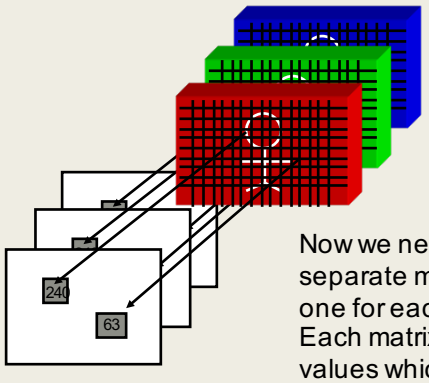
Coloured Images

Modifications to our structure for colour



Our matrix is now
Of size
640 x 480 x 3

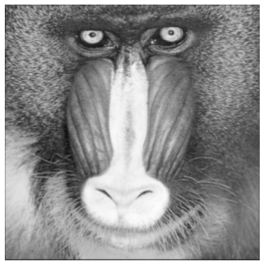
Now we need three separate matrices, one for each colour. Each matrix holds values which represent the corresponding colour in the image

Baboon in RGB

- Three components
  - M = {R, G, B}

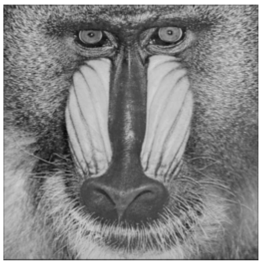
$$R = \begin{bmatrix} 73 & \cdots & 87 \\ \vdots & \ddots & \vdots \\ 27 & \cdots & 17 \end{bmatrix}, G = \begin{bmatrix} 66 & \cdots & 98 \\ \vdots & \ddots & \vdots \\ 36 & \cdots & 13 \end{bmatrix}, B = \begin{bmatrix} 31 & \cdots & 61 \\ \vdots & \ddots & \vdots \\ 36 & \cdots & 14 \end{bmatrix}$$

R

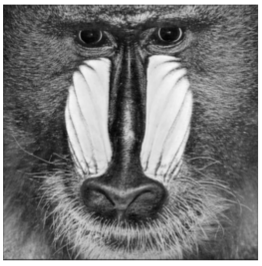


Red nose is brightest!

G



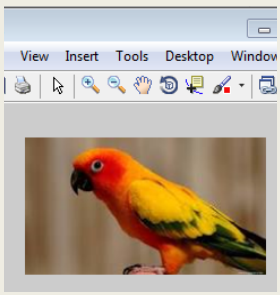
B

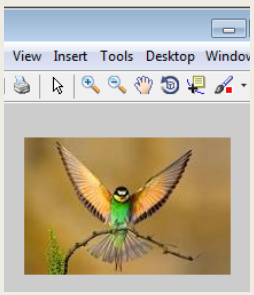


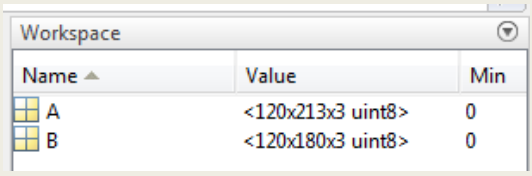
Blue Cheek is brightest

Loading, saving and showing bitmaps in Matlab

```
clc
clear all

A=imread('download1.bmp');
figure(1), imshow(A);

B=imread('download2.bmp');
figure(2), imshow(B);

imwrite(B, 'Save.bmp');
```

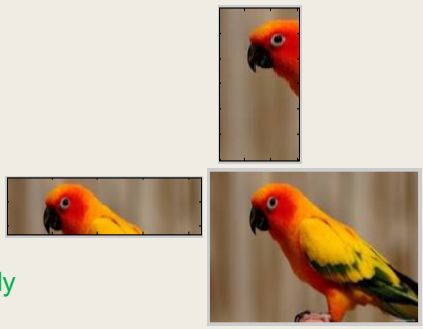






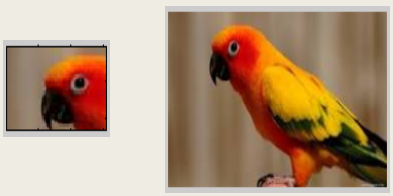
Manipulating the image

```
clc
clear all
A=imread('download1.bmp');
figure(1), imshow(A);

% operator to select certain rows only
newim1= A( 5:50, : , :);
figure(2), imshow(newim1);

% operator to select certain columns only
newim2= A( :, 20:80 , :);
figure(3), imshow(newim2);
```



Manipulating the image

```
clc
clear all
A=imread('download1.bmp');
figure(1), imshow(A);

% cropping part of the image
newim1= A( 5:50, 20:80 , :);
figure(2), imshow(newim1);
```



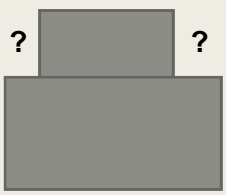
Piecing two images together.

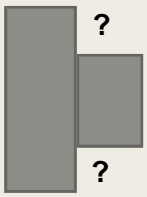
- Vertical concentration
  - *newim= [A(: , start:end , :) ; B(:,start:end,:)]*

Images 1 and 2 must have same number of columns

- Horizontal concentration
  - *newim= [A(start:end, :, :) B(start:end , :, :)]*

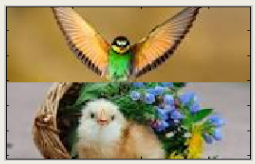
Images 1 and 2 must have same number of lines.





Piecing two images together (example)

```
clc
clear all
A=imread('download2.bmp');
figure(1), imshow(A);

B=imread('download3.bmp');
figure(2), imshow(B);

% Piecing two images together row wise
newim1 = [A( 20:80, : , :) ; B(20:80,:,:)];
figure(3), imshow(newim1);

% Piecing two images together column wise
newim2 = [A( :, 60:110, : ) B( :, 50:100,:)];
figure(4), imshow(newim2);
```



Altering the overall brightness and contrast of an image

- If we want a neutral change we must make the same change to all three primaries.
- Brightness
  - *An offset operation*
  - *Use + or –*
- Contrast
  - *A scaling operation*
  - *Use * or /*
- Exceeding 255 will cause ‘clipping’.
- Numbers less than zero will cause ‘crushing’

Changing Brightness

- newim=A-127;

imshow(newim);

- We can increase it again

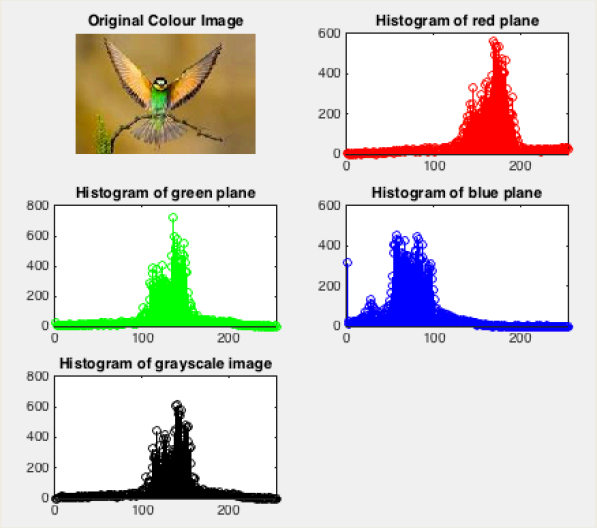
newim= A+127;

imshow(newim);

- If we increase/decrease values below 0 or above 255 we lose information in the image, try it.

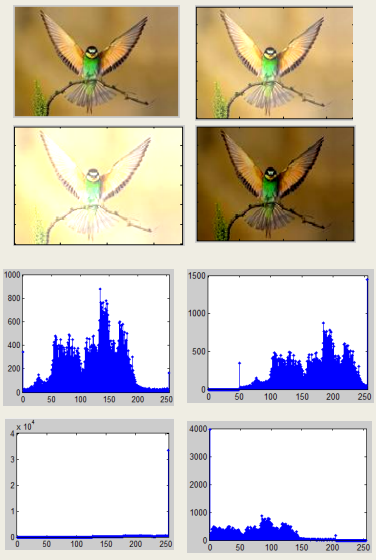
Histogram (example)

```
[rows, columns, numberOfColorBands] = size(A);
subplot(3, 2, 1);
imshow(A, []);
title('Original Colour Image');
%set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

redPlane = A(:, :, 1);
greenPlane = A(:, :, 2);
bluePlane = A(:, :, 3);
grayscale=rgb2gray(A);

% Let's get its histograms.
[pixelCountR, grayLevelsR] = imhist(redPlane);
subplot(3, 2, 2);
stem(pixelCountR, 'r');
title('Histogram of red plane');
xlim([0 grayLevelsR(end)]);
% Scale x axis manually. end indicates last array index

[pixelCountG, grayLevelsG] = imhist(greenPlane);
subplot(3, 2, 3);
stem(pixelCountG, 'g');
title('Histogram of green plane');
xlim([0 grayLevelsG(end)]); % Scale x axis manually.

[pixelCountB, grayLevelsB] = imhist(bluePlane);
subplot(3, 2, 4);
stem(pixelCountB, 'b');
title('Histogram of blue plane');
xlim([0 grayLevelsB(end)]); % Scale x axis manually.

[pixelCountgray, grayLevelsgray] = imhist(grayscale);
subplot(3, 2, 5);
stem(pixelCountgray, 'black');
title('Histogram of grayscale image');
xlim([0 grayLevelsgray(end)]); % Scale x axis manually.
```



Changing Brightness (Example)

```
clc
clear all
A=imread('download2.bmp');

figure(1), imshow(A);
newim1 = A + 50 ; figure(2),
imshow(newim1);
newim2 = A + 127 ; figure(3),
imshow(newim2);
newim3 = A - 50 ; figure(4),
imshow(newim3);
```



Changing Contrast

- Now use multiplication to increase the contrast.

*newim=A*2*
imshow(newim)

Try it and see the effect in image and the histogram

Changing the colour of an image.

- Operate (+ - * /) on different colours
- Example
  - *Give picture a red lift*

newim=A;
newim(:, :, 1)=newim(:, :, 1)+120;
imshow(newim);

Try it and see the effect in image and the histogram

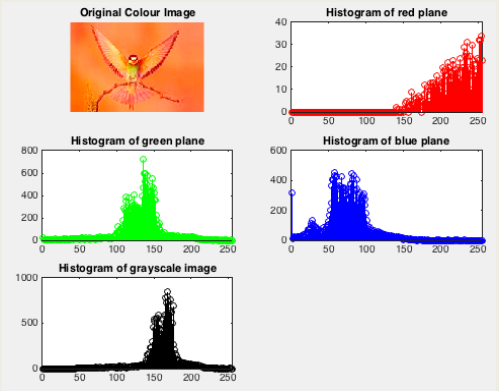


Image quantisation

```
A=imread('download1.bmp');
imshow(A);
threshRGB = multithresh(A,4); %Multilevel image thresholds
quantRGB = imquantize(A, threshRGB, [threshRGB 255]); %Quantize
image using specified quantization levels and output values
imshow(quantRGB);
```

120x213x3 uint8

186	189	184	194	201	200	201	208	206	193	187	194	169	143	167
163	172	169	180	189	189	189	191	185	171	173	163	138	141	172
136	148	149	152	160	171	179	176	169	157	162	152	127	145	178
129	136	133	133	140	150	159	161	160	155	152	147	127	143	174
108	105	106	107	112	120	130	138	144	144	137	135	117	125	155
81	68	77	80	84	92	103	113	123	127	131	128	109	106	131
60	47	53	55	61	68	79	91	103	112	117	117	98	85	95
48	39	34	34	39	45	56	70	87	101	106	104	91	76	69
39	27	17	13	17	22	33	49	71	88	99	95	89	77	64
37	25	7	2	3	7	16	29	48	63	75	79	83	79	71

threshRGB [43,93,129,183]

120x213x3 uint8

255	255	255	255	255	255	255	255	255	255	255	255	183	183	183
183	183	183	183	255	255	255	255	255	183	183	183	183	183	183
183	183	183	183	183	183	183	183	183	183	183	183	129	183	183
129	183	183	183	183	183	183	183	183	183	183	183	129	183	183
129	129	129	129	129	129	183	183	183	183	183	183	129	129	183
93	93	93	93	93	93	129	129	129	183	129	129	129	129	183
93	93	93	93	93	93	93	93	129	129	129	129	129	93	129
93	43	43	43	43	93	93	93	93	129	129	129	93	93	93
43	43	43	43	43	43	43	93	93	93	129	129	93	93	93
43	43	43	43	43	43	43	43	93	93	93	93	93	93	93

8-bit video games