

Basics Lab - 01

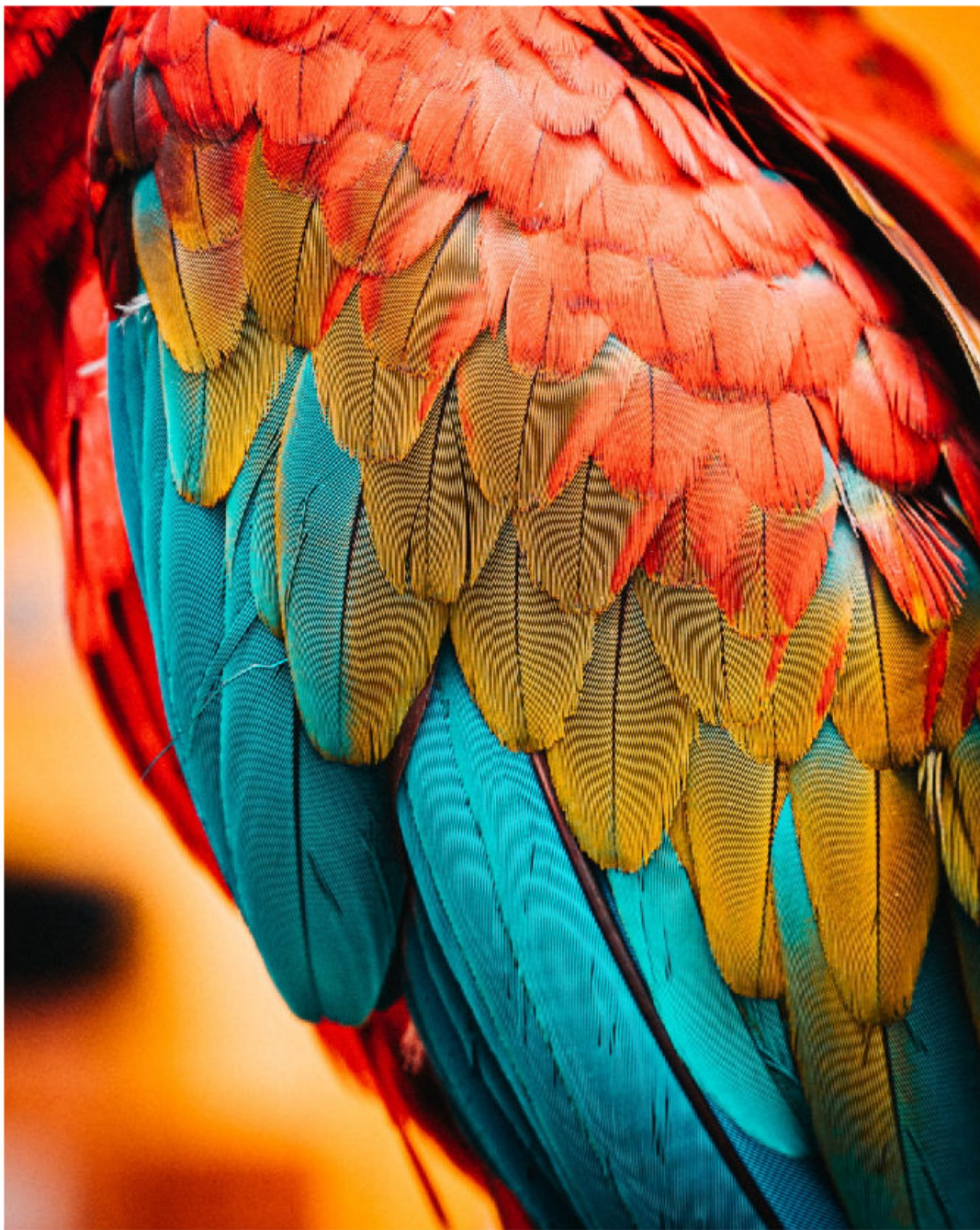
Importing Images

```
clc
close all
clear all

% import image into workspace
I = imread("parrot.jpg");
size(I)
```

```
ans = 1×3
      1250      1000         3
```

```
% image visualization
figure
imshow(I)
```



```
% saving image to file  
imwrite(I, "my_image.png")
```

```
% checking info  
imfinfo("my_image.png")
```

ans = struct with fields:

```
Filename: 'C:\Users\nilad\Videos\Lectures\Video Signals\Laboratory-20211122\LAB-01\my_image.png'
FileModDate: '28-Nov-2021 18:49:58'
FileSize: 2470993
Format: 'png'
FormatVersion: []
Width: 1000
Height: 1250
BitDepth: 24
ColorType: 'truecolor'
FormatSignature: [137 80 78 71 13 10 26 10]
Colormap: []
Histogram: []
InterlaceType: 'none'
Transparency: 'none'
SimpleTransparencyData: []
BackgroundColor: []
RenderingIntent: []
Chromaticities: []
Gamma: []
XResolution: []
YResolution: []
ResolutionUnit: []
XOffset: []
YOffset: []
OffsetUnit: []
SignificantBits: []
ImageModTime: '28 Nov 2021 17:49:58 +0000'
Title: []
Author: []
Description: []
Copyright: []
CreationTime: []
Software: []
Disclaimer: []
Warning: []
Source: []
Comment: []
OtherText: []
```

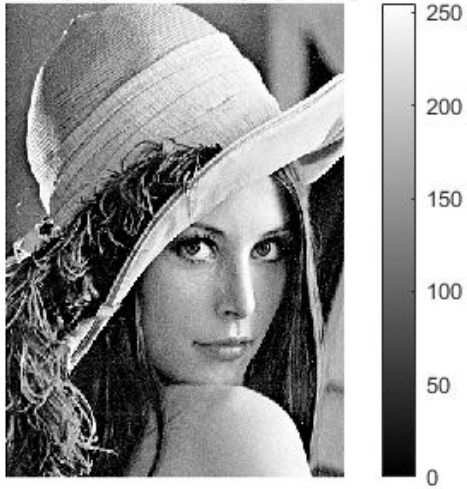
Working with Grayscale Image

```
clc
close all
clear all

img_gray = imread("Lena_grayscale.bmp");

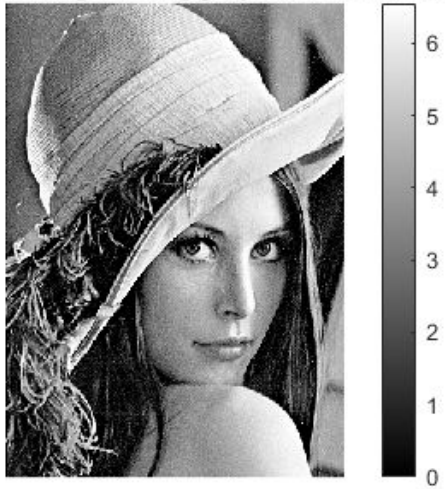
figure
imshow(img_gray)
colorbar
title("uint8 grayscale image [0,255]")
```

uint8 grayscale image [0,255]



```
% grayscale uint8 to grayscale uint16 conversion
img_gray_uint16 = im2uint16(img_gray);
figure
imshow(img_gray_uint16)
colorbar
title("uint8 --> uint16 [0,65535] grayscale conversion")
```

uint8 --> uint16 [0,65535] grayscale conversion



```
% grayscale uint8 to grayscale double conversion
img_gray_double = im2double(img_gray);
figure
imshow(img_gray_double)
colorbar
title("uint8 --> double [0,1] grayscale conversion")
```

uint8 --> double [0,1] grayscale conversion



TRUE COLOR IMAGES - RGB

```
clc
close all
clear all

img_rgb = imread("parrot.jpg");

img_rgb_red_plane = img_rgb(:, :, 1); % red plane image
img_rgb_green_plane = img_rgb(:, :, 2); % green plane image
img_rgb_blue_plane = img_rgb(:, :, 3); % blue plane image

figure
subplot(2,2,1)
imshow(img_rgb)
title("Truecolor image")

subplot(2,2,2)
imshow(img_rgb_red_plane)
title("Truecolor image - Red channel");

subplot(2,2,3)
imshow(img_rgb_green_plane)
title("Truecolor image - Green channel");

subplot(2,2,4)
imshow(img_rgb_blue_plane)
```



```
title("Truecolor image - Blue channel");
```

Truecolor image



Red plane image



Green plane image



Blue plane image



```
% for more clear visualization
null_image_3 = uint8(zeros(size(img_rgb)))
```

```
null_image_3 = 1250x1000x3 uint8 array
null_image_3(:,:,1) =
```

[illegible]

```
img_rgb_red = null_image_3;  
img_rgb_red(:,:,1) = img_rgb(:,:,1);  
  
img_rgb_green = null_image_3;  
img_rgb_green(:,:,2) = img_rgb(:,:,2);  
  
img_rgb_blue = null_image_3;  
img_rgb_blue(:,:,3) = img_rgb(:,:,3);
```

```
figure
subplot(2,2,1)
imshow(img_rgb)
title("Truecolor image")
```

```
subplot(2,2,2)
imshow(img_rgb_red)
```

```

title("Red channel image");

subplot(2,2,3)
imshow(img_rgb_green);
title("Green channel image");

subplot(2,2,4)
imshow(img_rgb_blue);
title("Blue channel image");

```

Truecolor image



Red channel image



Green channel image



Blue channel image

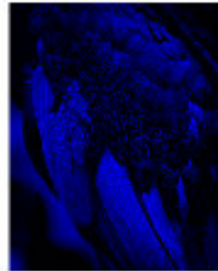


Image Conversion and Representation

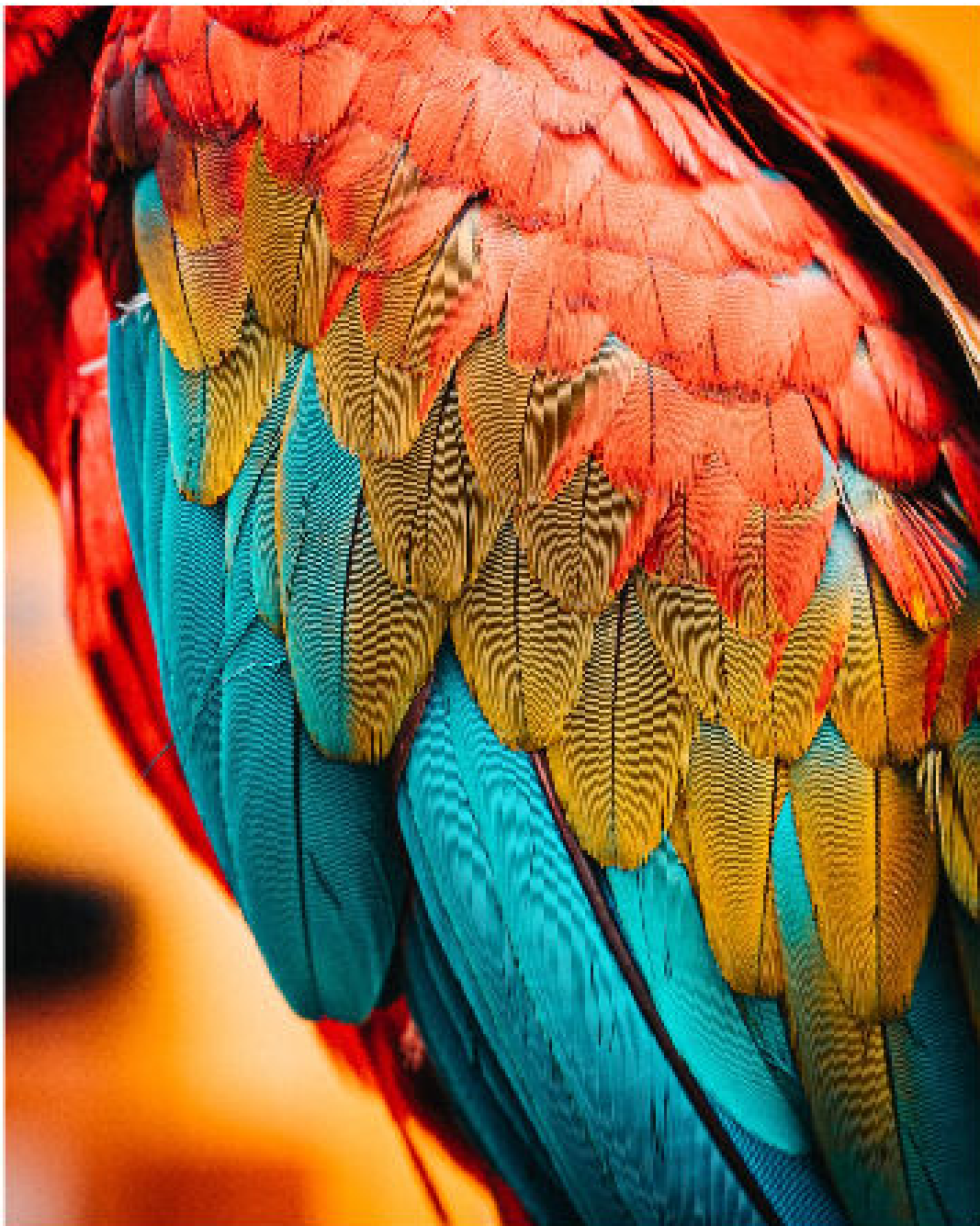
```

clc
close all
clear all

img_rgb = imread("parrot.jpg");
figure
imshow(img_rgb)
title("Truecolor image")

```


Truecolor image



```
% truecolor image to grayscale image conversion  
img_gray = rgb2gray(img_rgb);  
figure  
imshow(img_gray)  
title("Grayscale image")
```

Grayscale image



```
% grayscale image to binary or black and white image
% setting threshold to 0.5 (pixel intensity set to 1(white) if above threshold, or else 0(black))
img_bw_05 = im2bw(img_gray, 0.5);
figure
imshow(img_bw_05)
title("Black and White image")
```

Black and white image



```
figure
subplot(1, 4, 1)
imshow(img_gray)
title("Grayscale image")

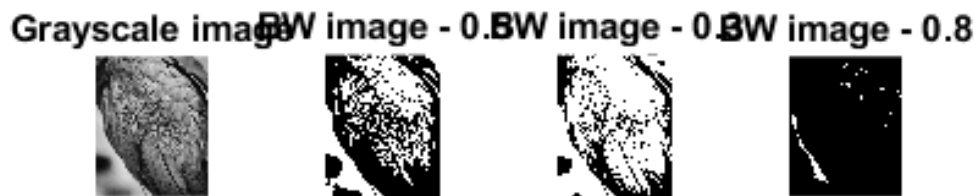
subplot(1, 4, 2)
imshow(img_bw_05)
title("BW image - 0.5") % plotting black and white image with 0.5 threshold
```

```

subplot(1, 4, 3)
img_bw_03 = im2bw(img_gray, 0.3);
imshow(img_bw_03)
title("BW image - 0.3")           % plotting black and white image with 0.3 threshold

subplot(1, 4, 4)
img_bw_08 = im2bw(img_gray, 0.8);
imshow(img_bw_08)
title("BW image - 0.8")           % plotting black and white image with 0.8 threshold

```



```

% grayscale to binary or black and white image using imbinarize function
figure
subplot(1, 4, 1)
imshow(img_gray)
title("Grayscale image")

subplot(1, 4, 2)
img_bw_auto_global = imbinarize(img_gray);
imshow(img_bw_auto_global)
title("Black and white - global method")           % plotting black and white auto global method

subplot(1, 4, 3)
img_bw_auto_local = imbinarize(img_gray, "adaptive");
imshow(img_bw_auto_local)
title("Black and white - local method")           % plotting black and white auto local method

subplot(1, 4, 4)

```

```
img_bw_detherring = dither(img_gray);
imshow(img_bw_detherring)
title("Black and white - detherring")
```

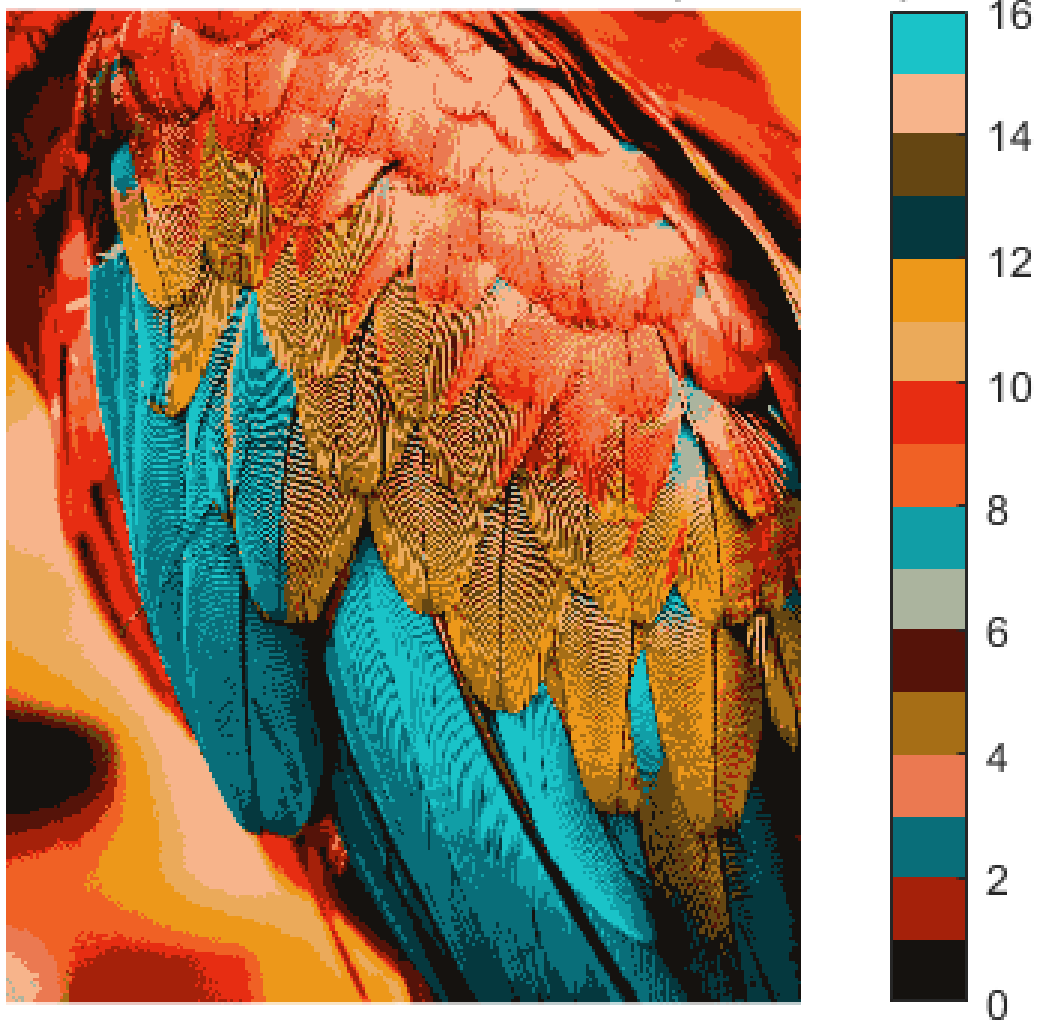
```
% plotting black and white detherring
```

Grayscale image Black and white image Black and white image with dithering Black and white image - detherring



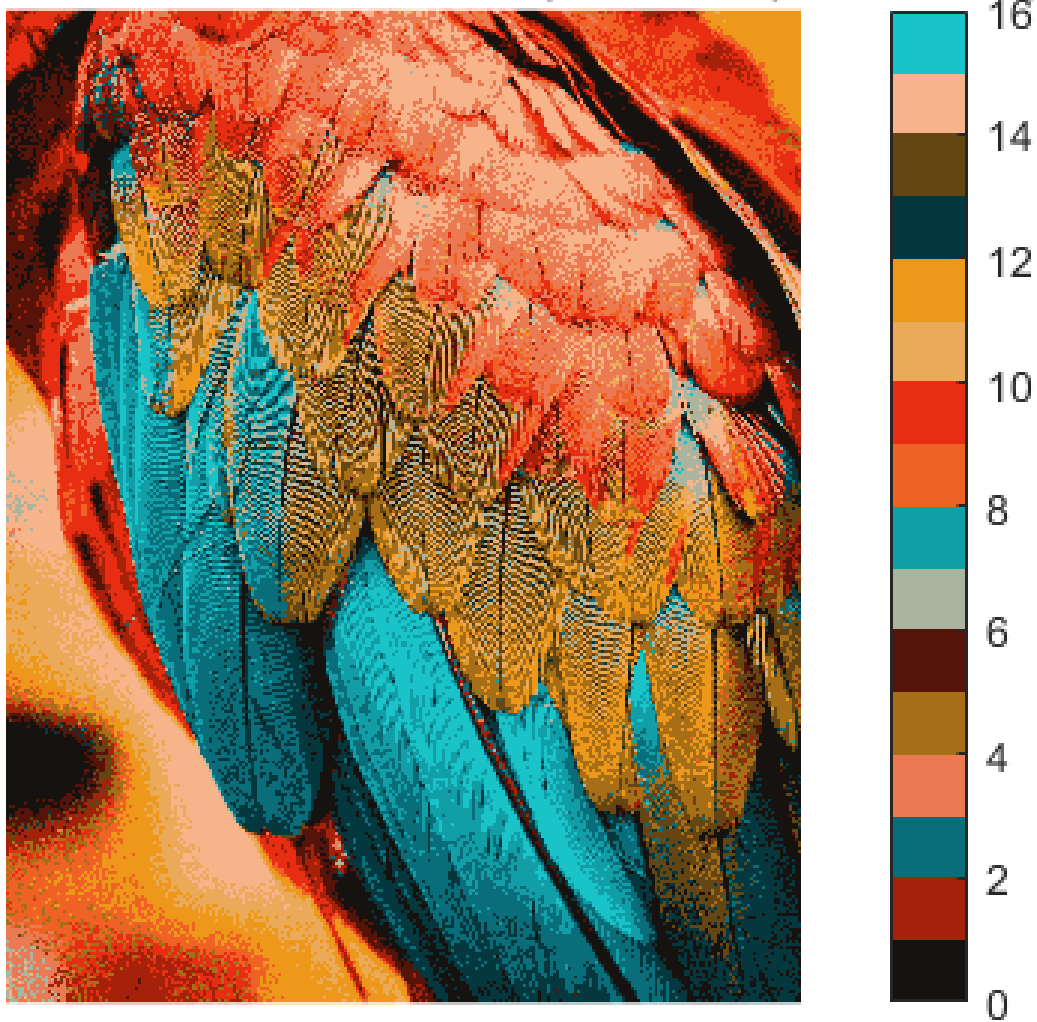
```
% indexed representation and conversion
[img_ind, map_ind] = rgb2ind(img_rgb, 16, "nodither");
figure
imshow(img_ind, map_ind)
title("Truecolor to Indexed conversion (16 colors)")
colorbar
```

Truecolor to Indexed conversion (16 colors)



```
[img_ind, map_ind] = rgb2ind(img_rgb, 16);  
figure  
imshow(img_ind, map_ind)  
title("Truecolor to Indexed conversion (16 colors) - dithering")  
colorbar
```


Truecolor to Indexed conversion (16 colors) - dithering



```
% indexed to truecolor conversion
[img_ind_rgb] = ind2rgb(img_ind, map_ind);
figure
imshow(img_ind_rgb)
title("Indexed (16 colors) --> Truecolor conversion")
```

indexed (16 colors) --> truecolor conversion



Image Resizing

```
clc
close all
clear all

img_gray = imread("Lena_grayscale.bmp");
figure
```

```
imshow(img_gray)
title("Grayscale image")
```

Grayscale image



```
img_gray_reduced = imresize(img_gray, 1/3);
figure
imshow(img_gray_reduced)
title("Reduced by 1/3rd ")
```

Reduced by 1/3rd



```
img_gray_enlarged = imresize(img_gray_reduced, 3, "nearest");
figure
imshow(img_gray_enlarged)
title("Reduced (1/3rd) and enlarged (x3) using Nearest-Neighbor interpolation")
```

rd) and enlarged (x3) using Nearest-Neighbor



```
img_gray_enlarged = imresize(img_gray_reduced, 3, "bilinear");  
figure  
imshow(img_gray_enlarged)  
title("Reduced (1/3rd) and enlarged (x3) using Bilinear interpolation")
```

d (1/3rd) and enlarged (x3) using Bilinear inter



```
img_gray_enlarged = imresize(img_gray_reduced, 3, "bicubic");  
figure  
imshow(img_gray_enlarged)  
title("Reduced (1/3rd) and enlarged (x3) using Bicubic interpolation")
```

d (1/3rd) and enlarged (x3) using Bicubic interpolation



```
img_gray_resized = imresize(img_gray, [200 500], "nearest");  
figure  
imshow(img_gray_resized)  
title("Resized to specific size using Nearest-Neighbor")
```

Resized to specific size using Nearest-Neighbor



Arithmetic Operations on Images

```
clc  
clear all
```



```
close all

img_gray = imread("Lena_grayscale.bmp");
figure
imshow(img_gray)
title("Lena Grayscale - [0, 255]")
```



```
lena_size = size(img_gray)
```

```
lena_size = 1x2
          420   300
```

```
img_rgb = imread("parrot.jpg");
img_gray_parrot = imresize(rgb2gray(img_rgb), lena_size(1:2));
figure
imshow(img_gray_parrot)
title("Parrot Grayscale - [0,255], resized as the previous image")
```

t Grayscale - [0,255], resized as the previous i



```
% addition operation on images
img_gray_lena_add_parrot = img_gray + img_gray_parrot;
figure
imshow(img_gray_lena_add_parrot)
title("Lena + Parrot")
```

Lena + Parrot



```
% subtraction operation on images  
img_gray_lena_sub_parrot = img_gray - img_gray_parrot;  
figure  
imshow(img_gray_lena_sub_parrot)  
title("Lena - Parrot")
```

Lena - Parrot



```
% multiplication operation on images  
img_gray_lena_mult_parrot = img_gray.*img_gray_parrot;  
figure  
imshow(img_gray_lena_mult_parrot)  
title("Lena x Parrot")
```

Lena x Parrot

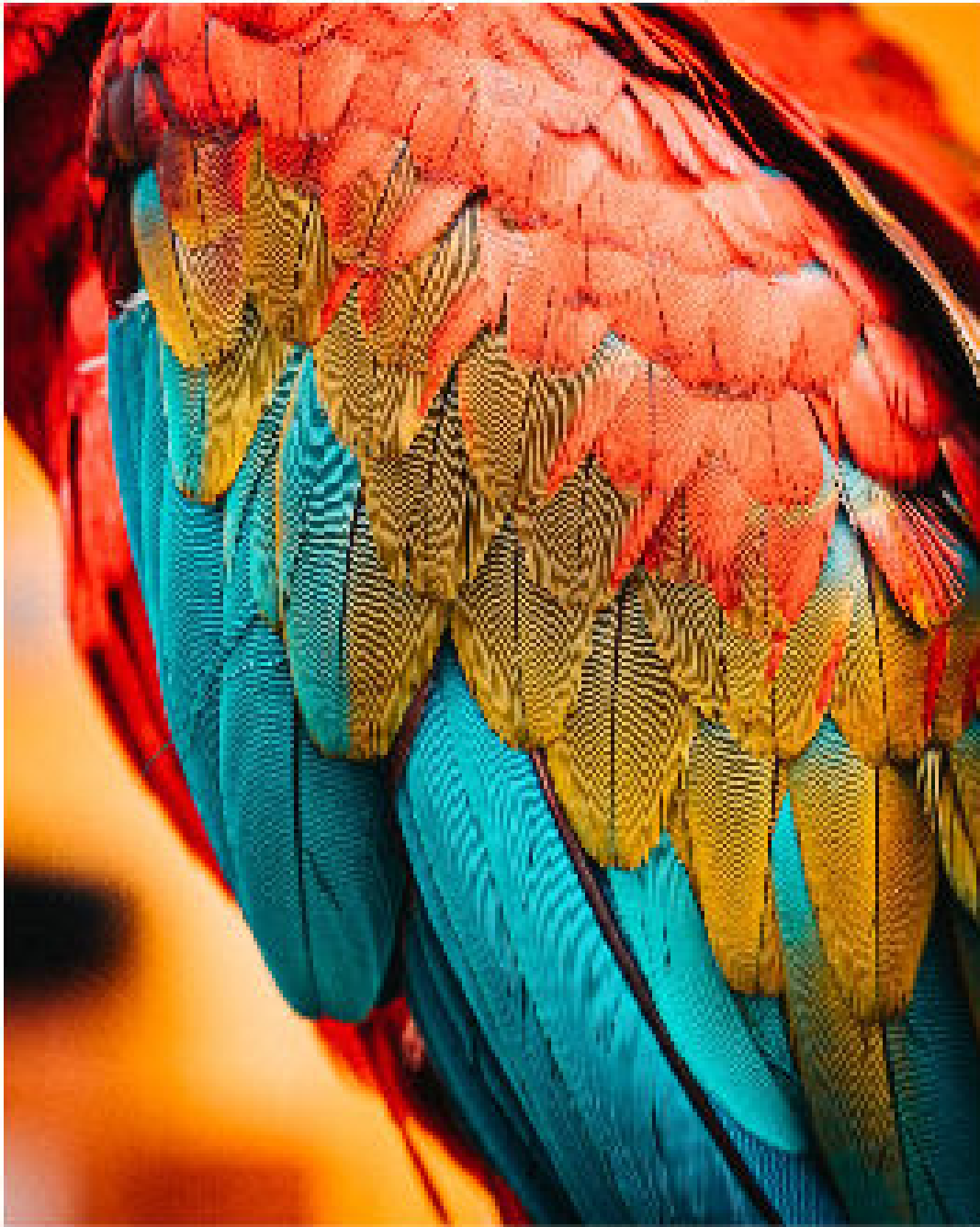


Cropping an Image

```
clc
clear all
close all

img_rgb = imread("parrot.jpg");
figure
imshow(img_rgb)
title("Original RGB image")
```

Original RGB Image



```
% Select the area by dragging the mouse, press the dx button and select  
% 'Crop Image'  
img_rgb_cropped = imcrop(img_rgb);  
figure  
imshow(img_rgb_cropped)  
title("Cropped image")
```


Cropped image

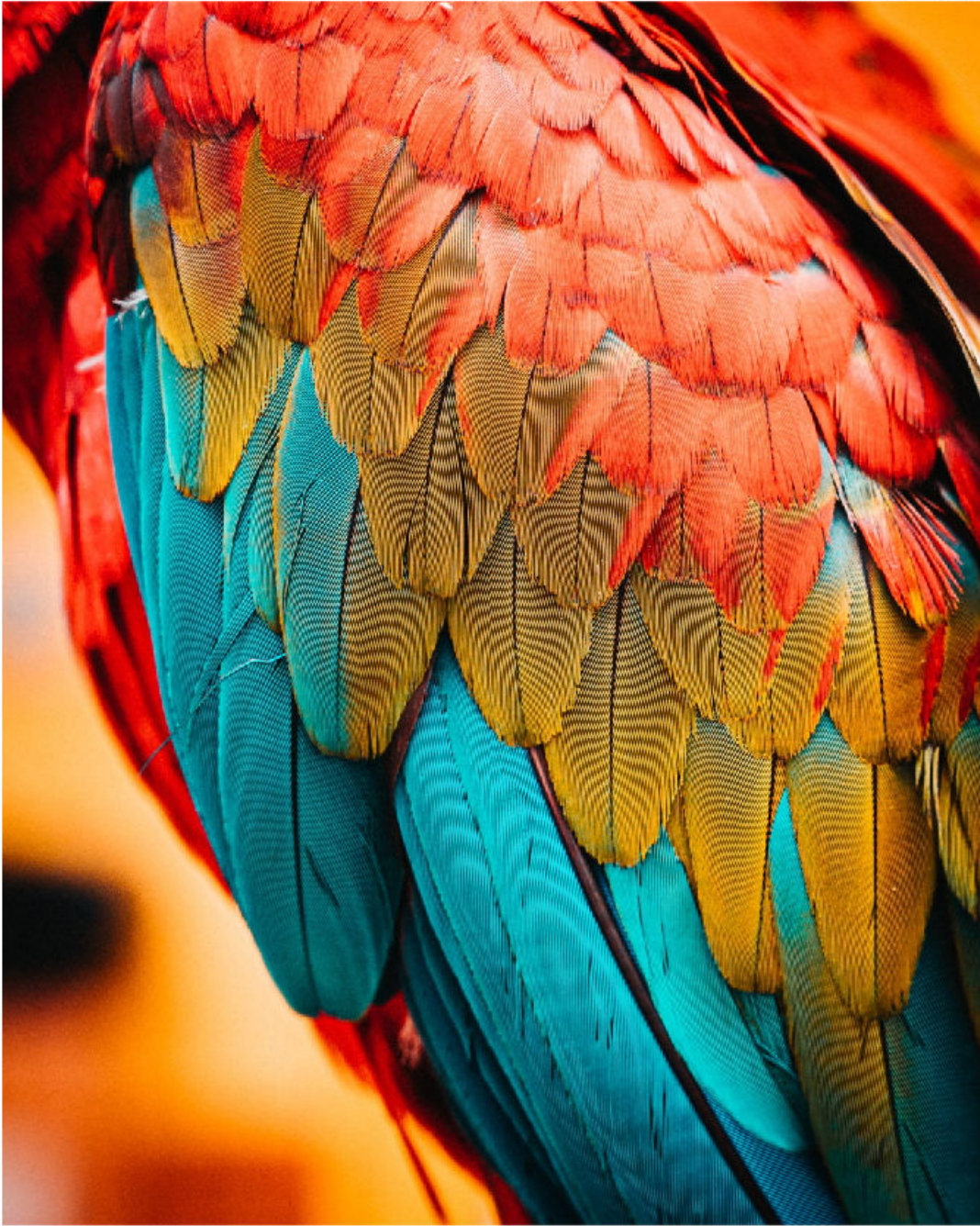


```
img_rgb_cropped_spec = imcrop(img_rgb, [1 100 200 50]);  
figure  
imshow(img_rgb_cropped_spec)  
title("Cropped image with region [1 100 200 10]")
```

Cropped image with region [1 100 200 10]



```
% alternative method  
img_rgb_cropped_spec_alt = img_rgb(100:150,1:201,:);  
figure  
imshow(img_rgb_cropped_spec_alt)
```



```
title("Cropped image with region [1 100 200 10]")
```

Cropped image with region [1 100 200 10]



Image Rotation

```
clc
clear all
close all

img_gray = imread("Lena_grayscale.bmp");
figure
imshow(img_gray)
title("Lena Grayscale Image")
```

Lena Grayscale Image



```
img_gray_rotated = imrotate(img_gray, 35, "nearest");
figure
imshow(img_gray_rotated)
title("35° rotation using Nearest-Neighbor interpolation")
```

rotation using Nearest-Neighbor interpolat



```
img_gray_rotated = imrotate(img_gray, 35, "nearest", "crop");  
figure  
imshow(img_gray_rotated)  
title("35° rotation using Nearest-Neighbor interpolation (cropped)")
```

ation using Nearest-Neighbor interpolation (cr



```
img_gray_rotated = imrotate(img_gray, 35, "bilinear");  
figure  
imshow(img_gray_rotated)  
title("35° rotation using Bilinear interpolation")
```

35° rotation using Bilinear interpolation



```
img_gray_rotated = imrotate(img_gray, 35, "bicubic");  
figure  
imshow(img_gray_rotated)  
title("35° rotation using Bicubic interpolation")
```


35° rotation using Bicubic interpolation



```
img_gray_transformed = imrotate(imresize(img_gray, 0.5), -180, "bicubic", "crop");  
figure  
imshow(img_gray_transformed)  
title("General transformation")
```

General transformation



Circular Transformation of an Image

```
clc
close all
clear all

% 150x150 round image

I = imread("Lena_grayscale.bmp");
figure
imshow(I)
```



```
t1 = [75 120];
br = t1 + 220;
I_crop = I(t1(2):br(2),t1(1):br(1));

target_size = 150;
I_crop = imresize(I_crop,[target_size,target_size]);

figure
imshow(I_crop)
```



```
for i=1:target_size
    for j=1:target_size
        if(((i-target_size/2)^2+(j-target_size/2)^2)>(target_size/2)^2)
            I_crop(i,j) = 254;
        end
    end
end
```

```
figure
imshow(I_crop)
```

