

Exp2: Study of different transformation techniques.

- a. Perform various Image type conversions.
- b. Image resize
- c. Image rotation
- d. Image negative

[NOTE: Please don't copy paste the sample codes given in the sections. Read an RGB image once. Then maintain consistent variables with meaningful names for each subsequent processing.]

a. Perform various Image type conversions:

1. Color → Individual Channels

Extracting R, G, B used frequently in pre-processing.

```
rgb = imread('peppers.png');  
R = rgb(:,:,1);  
G = rgb(:,:,2);  
B = rgb(:,:,3);
```

2. RGB → Grayscale

Converting a 3-channel color image to a single-channel intensity image.

```
rgb = imread('peppers.png');  
  
gray = rgb2gray(rgb);  
  
imshow(gray);
```

3. Grayscale → Binary (Thresholding)

Converts the grayscale image to a logical (0/1) image using a threshold.

```
gray = rgb2gray(imread('peppers.png'));  
  
bw = imbinarize(gray); % Otsu by default  
  
imshow(bw);
```

4. RGB → HSV

HSV separates color (hue) from intensity, useful for segmentation.

```
rgb = imread('peppers.png');  
  
hsvImg = rgb2hsv(rgb);  
  
imshow(hsvImg(:,:,1)); % Hue channel
```

5. HSV → RGB

Reverse of the above.

```
rgb2 = hsv2rgb(hsvImg);  
  
imshow(rgb2);
```

6. RGB → YCbCr

Used in compression, skin segmentation, etc.

```
rgb = imread('peppers.png');  
  
ycbcr = rgb2ycbcr(rgb);  
  
imshow(ycbcr(:,:,1)); % Y channel
```

7. RGB → Indexed Image

Assign colors to a colormap.

```
rgb = imread('peppers.png');  
  
[IND, map] = rgb2ind(rgb, 256);  
  
imshow(IND, map);
```

Indexed to RGB.

```
rgb2 = ind2rgb(IND, map);
```

```
imshow(rgb2);
```

8. Intensity Scaling (uint8 ↔ double ↔ uint16)

Conversion between numeric classes.

```
img = imread('cameraman.tif');
```

```
imgD = im2double(img);
```

double (0–1) → uint8

```
imgU8 = im2uint8(imgD);
```

uint8 → uint16

```
imgU16 = im2uint16(img);
```

b. Image resize:

Basic Image Resizing:

Use `imresize()` to change image dimensions.

Example 1: Resize to 50% of original size

```
img = imread('peppers.png');
```

```
resizedImg = imresize(img, 0.5);
```

```
imshow(resizedImg);
```

Resize to Specific Dimensions

Specify rows and columns directly.

Example 2: Resize to 256 × 256

```
img = imread('peppers.png');  
resizedImg = imresize(img, [256 256]);  
imshow(resizedImg);
```

c. Image rotation:

Basic Image Rotation

Use the MATLAB function `imrotate()`.

Example: Rotate by 45 degrees (counterclockwise).

```
img = imread('peppers.png');  
rot45 = imrotate(img, 45);  
imshow(rot45);
```

Rotate Clockwise:

MATLAB rotates counterclockwise for positive angles.
To rotate clockwise, give a negative angle.

Example: Rotate by 30 degrees clockwise

```
img = imread('peppers.png');  
rot_cw = imrotate(img, -30);  
imshow(rot_cw);
```

c. Image negative:

Concept of Image Negative:

The negative of an image reverses intensities:

For grayscale or each channel of RGB (0–255 range):

Negative pixel = $255 - \text{Original pixel}$

This converts light areas to dark and vice versa.

```
img = imread('cameraman.tif');           % or any RGB image
neg = imcomplement(img);
imshow(neg);
title('Negative Image');
```

Manual Method (Formula-Based)

Grayscale or RGB (uint8 type):

```
img = imread('cameraman.tif');
neg = 255 - img;
imshow(neg);
```

Image Negative in Double Format (0–1 range)

If image is converted to double:

```
img = im2double(imread('cameraman.tif'));
neg = 1 - img;    % because range is [0,1]
imshow(neg);
```