**Exercise 1.**

**Explanation:**

Read the given image,decrease the resolution as mentioned and observe the results

**Procedure:**

1. Read the image using fopen() and fread() with the given length and width.
2. As the image is 90 degree left rotated,so we rotated it 270 degree.
3. Now for 2 pixels, we used imfilter().
4. For 4 , 8 , 16 pixels,continued step 3.
5. Observe the result

**Code:**

fileId = fopen('D:\Documents\DIP Lab\crossroad.dat','r');

a = fread(fileId,[580 435],'\*uchar');

b = imrotate(a, 270);

figure, subplot(2,4,1), imshow(b), title('Original image');

i1 = imfilter(b, ones(2,2)/4);

i2 = imfilter(b, ones(4,4)/16);

i3 = imfilter(b, ones(8,8)/64);

i4 = imfilter(b, ones(16,16)/256);

subplot(2,4,5), imshow(i1), title('2x2');

subplot(2,4,6), imshow(i2), title('4x4');

subplot(2,4,7), imshow(i3), title('8x8');

subplot(2,4,8), imshow(i4), title('16x16');

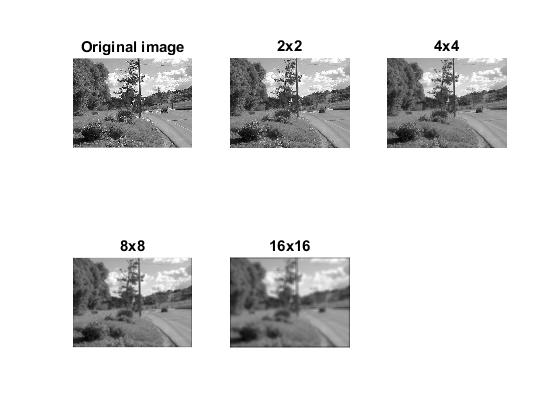
figure, subplot(2,4,1), plot(b), title('Original image');

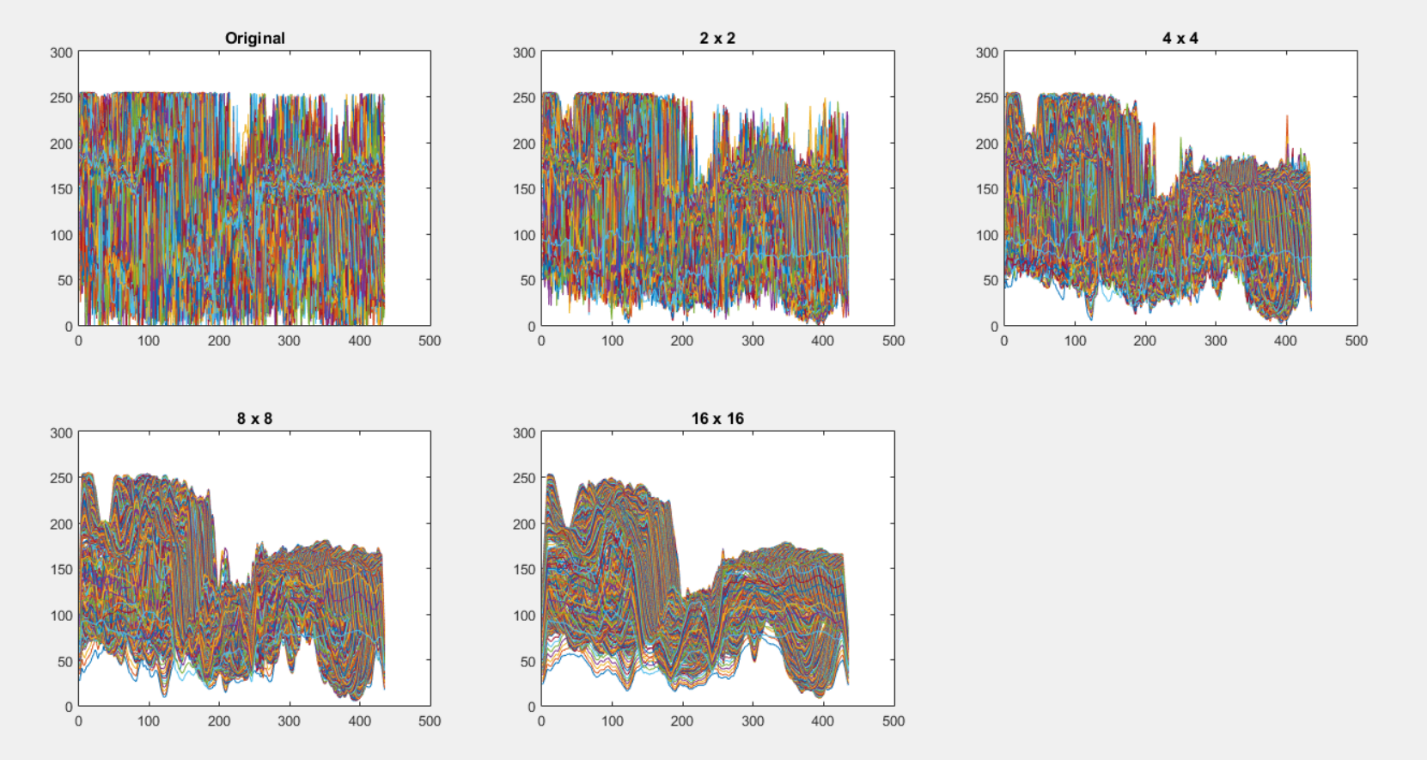
subplot(2,4,5), plot(i1), title('2x2');

subplot(2,4,6), plot(i2), title('4x4');

subplot(2,4,7), plot(i3), title('8x8');

subplot(2,4,8), plot(i4), title('16x16');

**Output:**

****

A screenshot of a cell phone

Description generated with very high confidence

**Observation:**

1. As image quality highly depends on resolution, it is not same for all images.
2. It differs in different portion and in different object in same image.
3. In the given image, the tall tree and the car fade faster than other objects.

**Exercise 2.**

**Problem Statement:**

**Explanation:**

The problem is to read the given image and gradually reduce the number of bits to represent each pixel and observe the result.

**Procedure:**

1. Read the image using fopen() and fread() with given line length.
2. As the given is rotated 90 degree left, we rotate it 270 degree.
3. Make a new image with same length and width of the original image.
4. Assign each pixel with original image by diving it by 8 for 5 bits pixel.
5. Follow step 4 by diving the original image by 16, 32, 64 and 128.
6. Observe the output.

**Code:**

fileId = fopen('D:\Documents\DIP Lab\crossroad.dat','r');

a = fread(fileId,[580 435], '\*uint8');fclose all;

b = imrotate(a, 270);

b = uint8(b);

i1 = b./8;

i2 = b./16;

i3 = b./32;

i4 = b./64;

i5 = b./128;

figure,

subplot(2,3,1), imshow(b), title('Original');

subplot(2,3,2), imshow(mat2gray(i1)), title('5 bit');

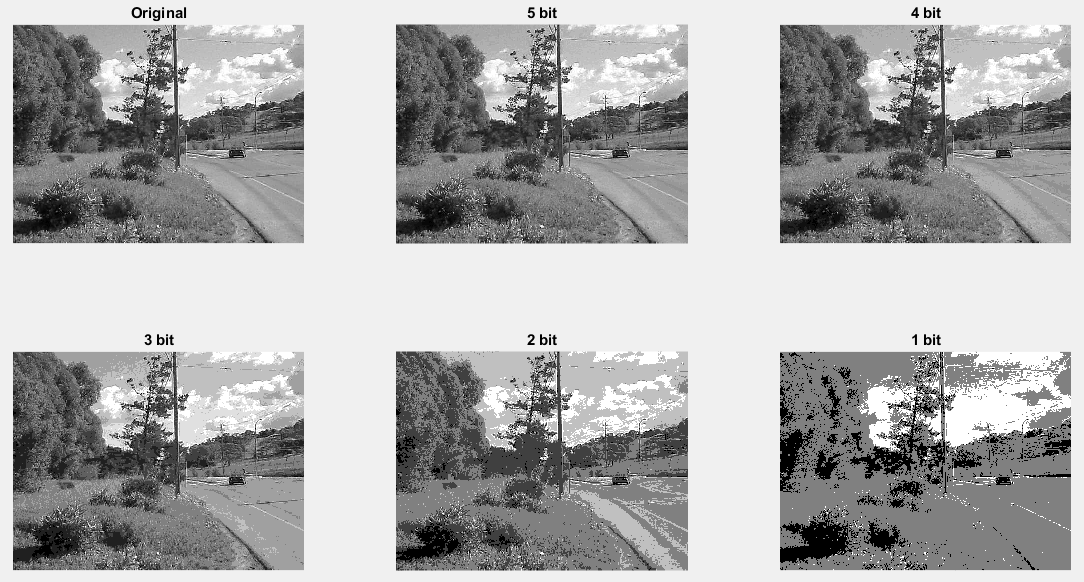
subplot(2,3,3), imshow(mat2gray(i2)), title('4 bit');

subplot(2,3,4), imshow(mat2gray(i3)), title('3 bit');

subplot(2,3,5), imshow(mat2gray(i4)), title('2 bit');

subplot(2,3,6), imshow(mat2gray(i5)), title('1 bit');

**Output:**



**Observation:**

The coloring details decays as the number of bits reduces and finally it becomes a fully binary image.

**Exercise 3.**

**Explanation:**

The problem is to read the given image. From grayscale image, we have to change it to a blue image. Then we have to change the color gradually from black to green to yellow to white for given range of pixel value.

**Procedure:**

1. Read the image using fopen() and fread() with given line length.
2. Make a new image with same height and width and depth of 3.
3. Assign the given image to third depth of new image which is the blue layer.
4. Make another new image with same height and width and depth of 3.
5. For pixel value 0 to 100, assign the pixel value of the image to the green layer.
6. For pixel value 101 to 200, assign the pixel value of the image to green layer and red layer to make the image yellowish.
7. For pixel value 201 to 255, assign the pixel value of the image to green , red and blue layer to whiten the image.
8. Observe the output.

**Code:**

fileId = fopen('D:\Documents\DIP Lab\standford.dat', 'r');

a = fread(fileId, [580 435], '\*uint8');

b = imrotate(a, 270);

imgblue = zeros(435, 580, 3);

imgblue(:, :, 3) = mat2gray(b);

imgnew = uint8(zeros(435, 580, 3));

for i = 1: 435

for j = 1: 580

if b(i, j) <= 100

imgnew(i, j, 1) = 0;

imgnew(i, j, 2) = 255;

imgnew(i, j, 3) = 0;

elseif b(i, j) <= 200

imgnew(i, j, 1) = 255;

imgnew(i, j, 2) = 255;

imgnew(i, j, 3) = 0;

else

imgnew(i, j, 1) = 255;

imgnew(i, j, 2) = 255;

imgnew(i, j, 3) = 255;

end

end

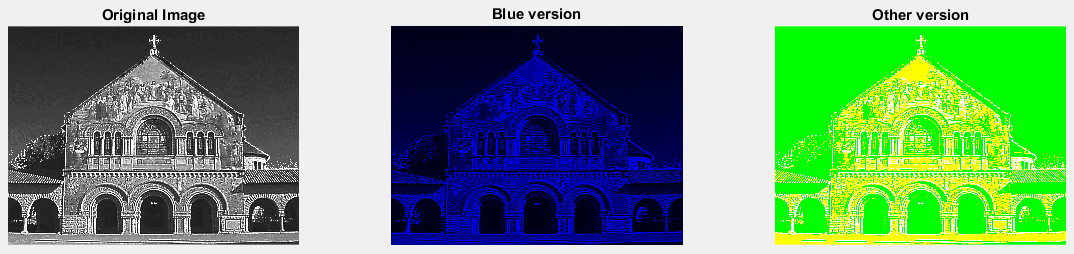
end

figure, subplot(1,3,1), imshow(b), title('Original Image');

subplot(1,3,2), imshow(imgblue), title('Blue version');

subplot(1,3,3), imshow(imgnew), title('Other version');

**Output:**



**Exercise 4.**

**Explanation:**

The problem is to read an RGB image. Then change it to HSV image. After that, rotate the hue 120 degree and -120 degree. And get the same result processing an RGB image.

**Procedure:**

1. Read the image using fopen and fread() with given line length.
2. Change it to HSV image using rgb2hsv() function.
3. Take the hue layer of HSV image and rotate the value 120 degree in both directions.
4. Revert it to RGB image using hsv2rgb() function and display.
5. Rotating 120 degree means replacing red by blue, green by red and blue by green in RGB image. This also can be achieved by swapping color planes.
6. Rotating -120 degree means replacing red by green, green by blue and blue by red in RGB image. We can achieve it by swapping the color planes.

**Code:**

i = imread('D:\Documents\DIP Lab\flip.jpg');

hsvImg = rgb2hsv(i);

hue = hsvImg(:, :, 1) \* 360;

hsvImg(:,:,1) = (mod(hue + 120, 360)) / 360;

img120 = hsv2rgb(hsvImg);

hsvImg(:,:,1) = (mod(hue - 120, 360)) / 360;

imgneg120 = hsv2rgb(hsvImg);

[h, w, d] = size(i);

rgb120 = uint8(zeros(h, w, d));

rgb120(:, :, 1) = i(:, :, 3);

rgb120(:, :, 2) = i(:, :, 1);

rgb120(:, :, 3) = i(:, :, 2);

rgbneg120 = uint8(zeros(h, w, d));

rgbneg120(:, :, 1) = i(:, :, 2);

rgbneg120(:, :, 2) = i(:, :, 3);

rgbneg120(:, :, 3) = i(:, :, 1);

subplot 231; imshow(i);title('Original');

subplot 232;imshow(img120);title('120 Degree Rotated HSV');

subplot 233; imshow(imgneg120), title('-120 Degree Rotated HSV');

subplot 235;imshow(rgb120), title('120 Degree Rotated RGB');

subplot 236;imshow(rgbneg120), title('-120 Degree Rotated RGB')

**Output:**

