**JPEG IMAGE COMPRESSION USING MATLAB**

*By*

**AMIRTHA PRASAD (22BEC1002)**

**DEEPIKA B (22BEC1367)**

**SHERINA ESTHER D (22BEC1198)**

A project report submitted to

**Dr. ASHOK MONDAL**

**SCHOOL OF ELECTRONICS ENGINEERING**

In partial fulfilment of the requirements for the course

**BECE301L DIGITAL SIGNAL PROCESSING**

in

**B.TECH ELECTRONICS AND COMMUNICATION ENGINEERING**



**Vandalur – Kelambakkam Road**

**Chennai – 600127**

**February 2024**

**PROBLEM STATEMENT:**

Designing an effective jpeg image compression remains a challenge. This study aims to develop a robust algorithm incorporating block-based sampling, quantization, and zigzag encoding techniques using the concept of Discrete Cosine Transform .The system will utilise DSP algorithms to implement the JPEG compression standard, enabling the reduction of image file sizes while maintaining acceptable image quality. The objective is to strike a balance between achieving high compression ratios while minimising compression artefacts for practical application across diverse domains

**CODE:**

clc;

clear all;

close all;

% Reading and displaying of original image

name = input('Enter the image name (with extension): ', 's');

A = imread(name);

subplot(3, 2, 1);

imshow(A);

title('Original image');

% Convert RGBA to RGB if image has alpha channel

if size(A, 3) == 4

A = A(:, :, 1:3);

end

% Converting original image into grayscale and plotting it

B = rgb2gray(A);

subplot(3, 2, 2);

imshow(B);

title('Original to grayscale image');

% n1, n2 shows the resolution of image

b = double(B);

[n1, n2] = size(b);

% plotting grayscale image

subplot(3, 2, 3);

imshow(B);

title('Grayscale image');

% Converting Grayscale image to RGB image and plotting it

C = gray2ind(B);

subplot(3, 2, 4);

imshow(C);

title('Grayscale to RGB image');

% plotting histogram of grayscale image

subplot(3, 2, 5);

imhist(B);

title('Histogram of grayscale image');

% Resize the image based on user input

new\_height = input('Enter the new height of the image: ');

new\_width = input('Enter the new width of the image: ');

resized\_A = imresize(A, [new\_height, new\_width]);

% Compression of resized image

rate = input('Enter compression quality % (1 < x < 100): ');

original = double(resized\_A) / 255;

rate = 1000 \* rate;

RGB = zeros(size(resized\_A));

for i = 1:3

im = original(:,:,i);

img\_dct = dct2(im);

img\_pow = (img\_dct).^2;

img\_pow = img\_pow(:);

[B, index] = sort(img\_pow);

B = flipud(B);

index = flipud(index);

compressed\_dct = zeros(size(im));

for k = 1:min(rate, numel(index)) % Ensure not to exceed the number of DCT coefficients

compressed\_dct(index(k)) = img\_dct(index(k));

end

img\_dct = idct2(compressed\_dct);

RGB(:,:,i) = img\_dct;

end

% plotting compressed image

subplot(3, 2, 6);

imshow(RGB);

title('DCT compressed image');

% Writing compressed image to file

[~, name, ext] = fileparts(name);

imwrite(RGB, ['compressed3\_' name '.jpg']);

figure

subplot(2,2,1)

imshow(A);

title('Original image');

subplot(2,2,4)

imshow(RGB)

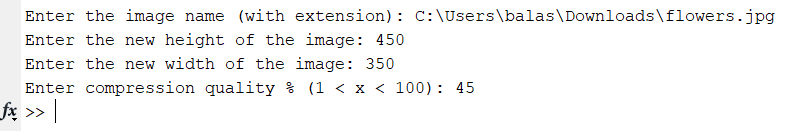
title('DCT compressed image');

**EXPLANATION OF THE CODE:**

This MATLAB code performs several image processing tasks:

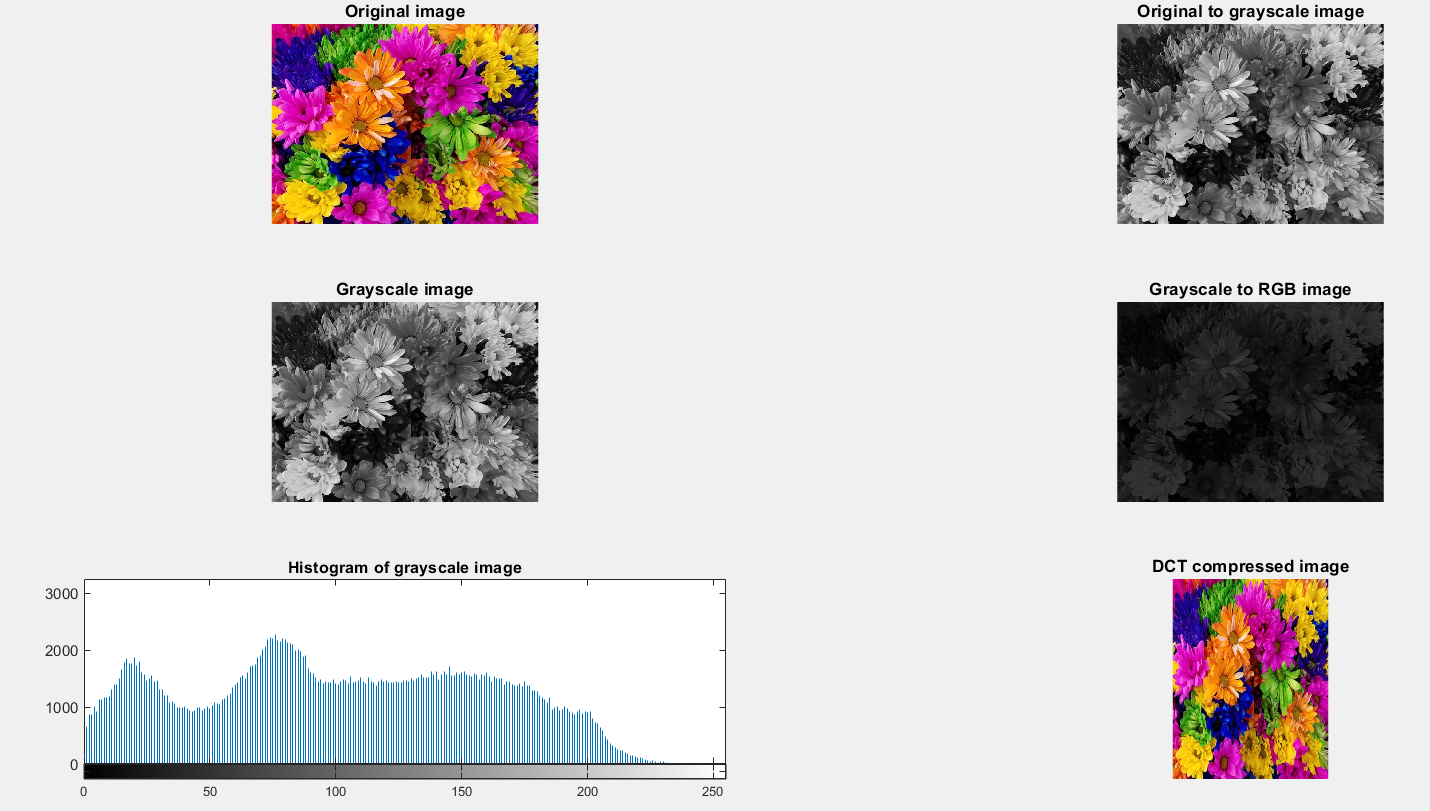
1. Reads an image specified by the user, displays it, and converts it to grayscale.
2. Converts the grayscale image back to RGB and displays it.
3. Plots the histogram of the grayscale image.
4. Resizes the image based on user input.
5. Implements a DCT-based image compression algorithm on the resized image.
6. Displays the compressed image and saves it as a JPEG file with a filename indicating the compression factor.

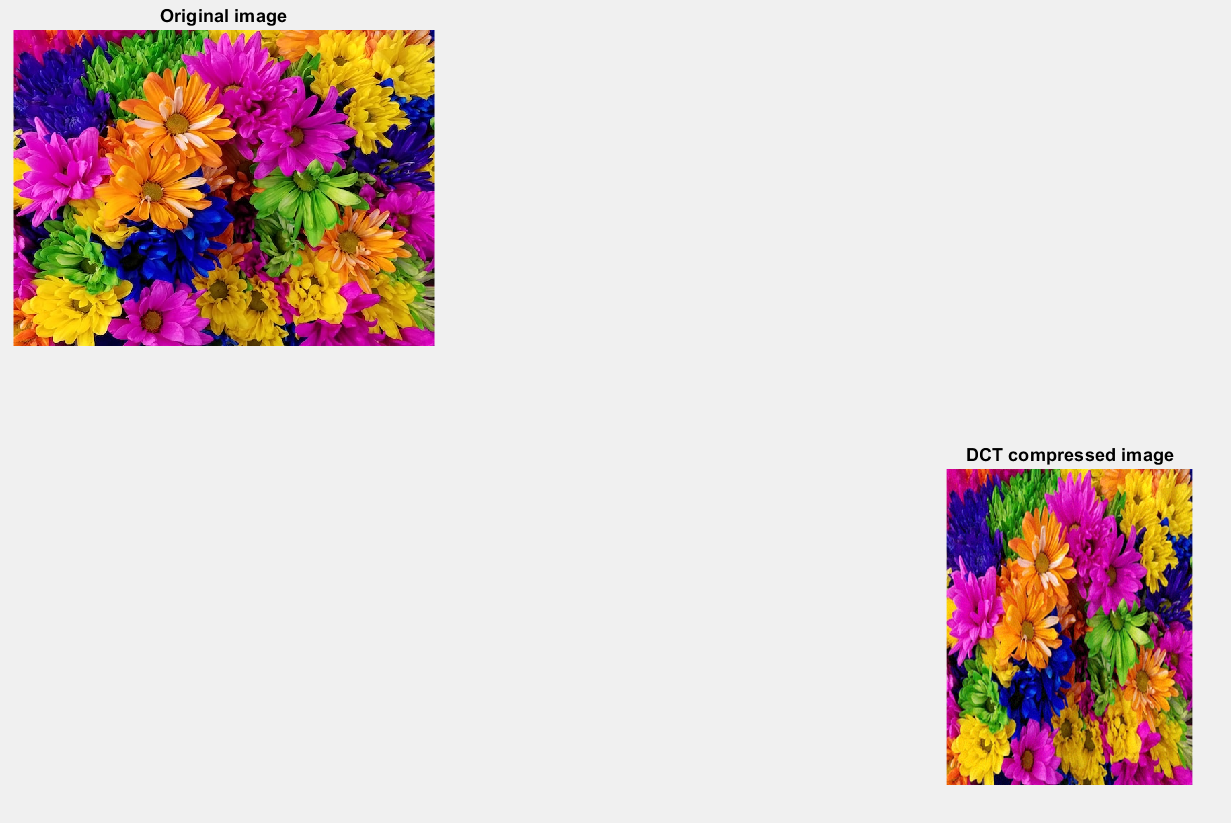
**RESULT-1:**

****

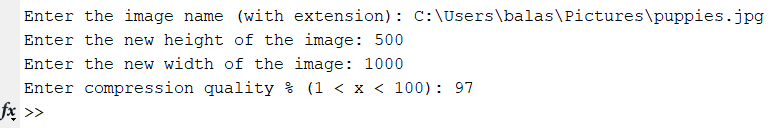
**Original size: 132.2KB**

**Compressed size: 42.7KB**

****

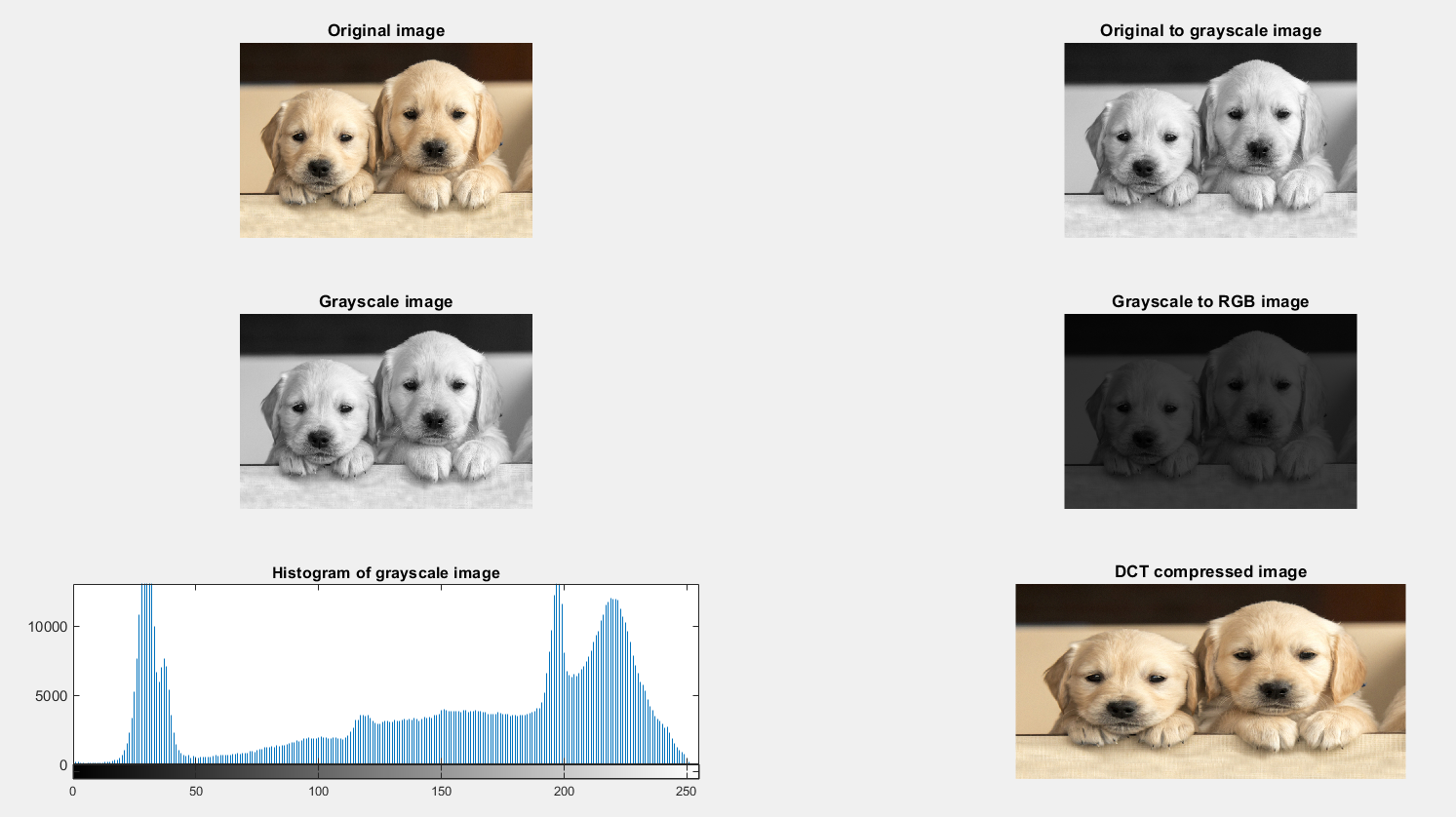
****

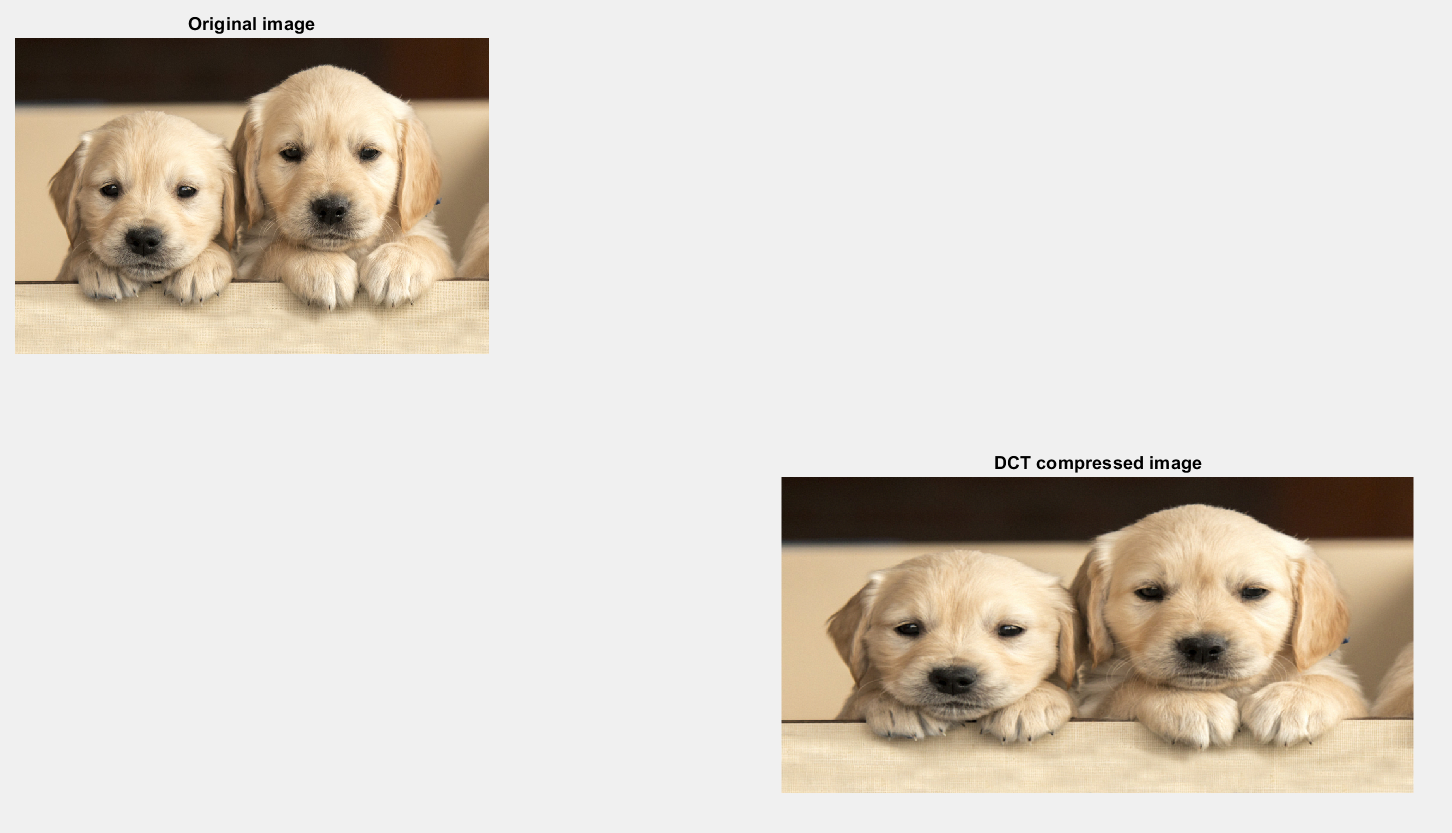
**RESULT-2**

****

**Original size: 205.4KB**

**Compressed size: 50.1KB**

****

****

Thus, JPEG image compression using DCT algorithm is achieved.