

```

% Name : Nenavath Dhanush Rathod
% BT23ECE120

% JPEG Compression
clear all;
close all;
clc;

% Read Image
img = imread('c.png');
img = double(img);

% Convert RGB to YCbCr

Y = 0.299*img(:,:,1) + 0.587*img(:,:,2) + 0.114*img(:,:,3);
Cb = -0.168736*img(:,:,1) - 0.331264*img(:,:,2) + 0.5*img(:,:,3) + 128;
Cr = 0.5*img(:,:,1) - 0.418688*img(:,:,2) - 0.081312*img(:,:,3) + 128;

channels = {Y, Cb, Cr};
reconstructed_channels = cell(1,3);

% JPEG Parameters

blockSize = 8;

% Luminance and chrominance quantization matrices
Q_Y = [16 11 10 16 24 40 51 61;
        12 12 14 19 26 58 60 55;
        14 13 16 24 40 57 69 56;
        14 17 22 29 51 87 80 62;
        18 22 37 56 68 109 103 77;
        24 35 55 64 81 104 113 92;
        49 64 78 87 103 121 120 101;
        72 92 95 98 112 100 103 99];
Q_C = Q_Y * 2; % coarser for Cb, Cr
Q_matrices = {Q_Y, Q_C, Q_C};

% Process Each Channel

for c = 1:3
    channel = channels{c};
    [h, w] = size(channel);
    h2 = ceil(h/blockSize)*blockSize;
    w2 = ceil(w/blockSize)*blockSize;

    padded = zeros(h2, w2);
    padded(1:h, 1:w) = channel - 128; % level shift

    Q = Q_matrices{c};
    recon = zeros(h2, w2);

    for i = 1:blockSize:h2
        for j = 1:blockSize:w2
            block = padded(i:i+7, j:j+7);

```

```

        % DCT
        dctBlock = dct2(block);

        % Quantization
        quantBlock = round(dctBlock ./ Q);

        % Dequantization
        dequantBlock = quantBlock .* Q;

        % Inverse DCT
        idctBlock = idct2(dequantBlock);

        recon(i:i+7, j:j+7) = idctBlock;
    end
end

recon = recon + 128; % reverse level shift
reconstructed_channels{c} = uint8(recon(1:h, 1:w));
end

% Convert YCbCr back to RGB

Y = double(reconstructed_channels{1});
Cb = double(reconstructed_channels{2}) - 128;
Cr = double(reconstructed_channels{3}) - 128;

R = Y + 1.402 * Cr;
G = Y - 0.344136 * Cb - 0.714136 * Cr;
B = Y + 1.772 * Cb;

reconstructed_RGB = uint8(cat(3, R, G, B));

% Display Results

figure;
subplot(1,2,1);
imshow(uint8(img));
title('Original RGB Image');

subplot(1,2,2);
imshow(reconstructed_RGB);
title('JPEG Compressed Image');

```

Original RGB Image



JPEG Compressed Image

