

ECE F344
Information Theory and Coding
Assignment 2

Sem-2

2024-25

Date- 20 March 2025

Submitted by

Lakshya Jain

2022AAPS0247P

Code

```
img = imread('space_picture.jpeg');  
figure(1); imshow(img); title('Input Image');  
imgYCbCr = rgb2ycbcr(img);  
figure(2); imshow(imgYCbCr); title('YCbCr Format');
```

```
blk = 8;  
[rows, cols, ~] = size(imgYCbCr);  
rows = floor(rows/blk)*blk;  
cols = floor(cols/blk)*blk;  
imgYCbCr = imgYCbCr(1:rows, 1:cols, :);
```

```
imgDbl = double(imgYCbCr) - 128;  
Y = imgDbl(:, :, 1);  
Cb = imgDbl(:, :, 2);  
Cr = imgDbl(:, :, 3);
```

%DCT

```
dct2d = @(block) dct2(block.data);  
Y_DCT = blockproc(Y, [blk blk], dct2d);  
Cb_DCT = blockproc(Cb, [blk blk], dct2d);  
Cr_DCT = blockproc(Cr, [blk blk], dct2d);
```

% Quantization matrices

```
QY = [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];
QC = [17 18 24 47 99 99 99 99;
18 21 26 66 99 99 99 99;
24 26 56 99 99 99 99 99;
47 66 99 99 99 99 99 99;
99 99 99 99 99 99 99 99;
99 99 99 99 99 99 99 99;
99 99 99 99 99 99 99 99;
99 99 99 99 99 99 99 99];

```

% Quantizing

```

qY = blockproc(Y_DCT, [blk blk], @(b) round(b.data./QY));
qCb = blockproc(Cb_DCT, [blk blk], @(b) round(b.data./QC));
qCr = blockproc(Cr_DCT, [blk blk], @(b) round(b.data./QC));

```

%Y channel DCT coefficients

```

figure(3);
logY = log(abs(qY)+1);
imshow(logY, []); title('DCT Coefficients (Y)');
colormap(jet); colorbar;

```

% Inverse quantization

```

invQuantY = @(b) b.data .* QY;
invQuantC = @(b) b.data .* QC;
Y_invQuant = blockproc(qY, [blk blk], invQuantY);
Cb_invQuant = blockproc(qCb, [blk blk], invQuantC);
Cr_invQuant = blockproc(qCr, [blk blk], invQuantC);

```

%Inverse DCT

```
idct2d = @(b) idct2(b.data);  
Y_rec = blockproc(Y_invQuant, [blk blk], idct2d);  
Cb_rec = blockproc(Cb_invQuant, [blk blk], idct2d);  
Cr_rec = blockproc(Cr_invQuant, [blk blk], idct2d);
```

% Merge channels and convert back to uint8 format after adding offset

```
recYCbCr = zeros(size(imgYCbCr));  
recYCbCr(:, :, 1) = Y_rec + 128;  
recYCbCr(:, :, 2) = Cb_rec + 128;  
recYCbCr(:, :, 3) = Cr_rec + 128;  
recYCbCr = uint8(recYCbCr);  
recRGB = ycbcr2rgb(recYCbCr);
```

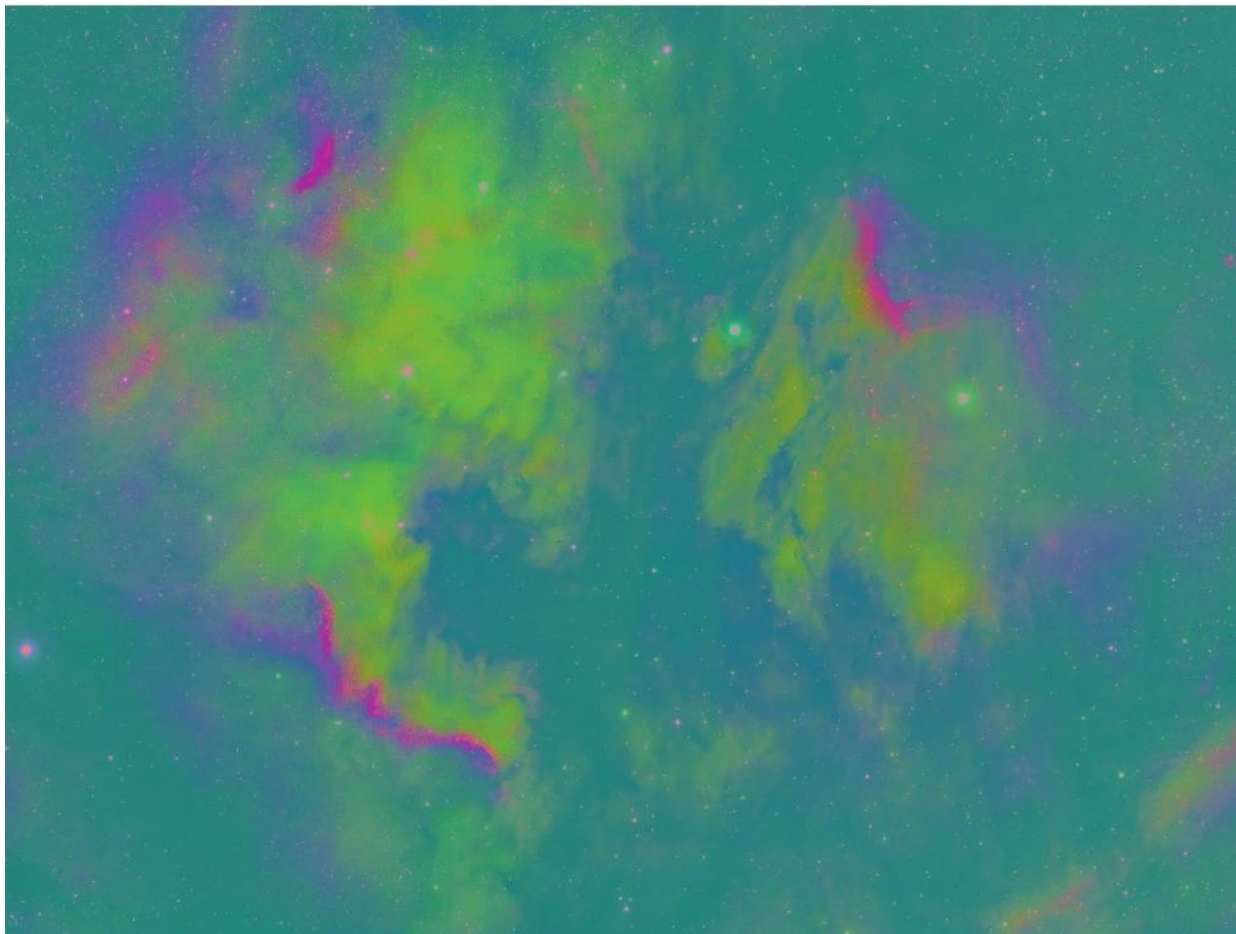
```
figure(4); imshow(recRGB); title('Reconstructed Image');
```

```
origBits = numel(img) * 8;  
nzCoeffs = numel(find(qY)) + numel(find(qCb)) + numel(find(qCr));  
approxBits = nzCoeffs * 8;  
compRatio = origBits / approxBits;  
disp(['Compression Ratio: ', num2str(compRatio)]);  
figure(5);  
subplot(1,3,1); imshow(img); title('Original');  
subplot(1,3,2); imshow(logY, []); title('DCT View'); colormap(jet);  
subplot(1,3,3); imshow(recRGB); title('Reconstructed');
```

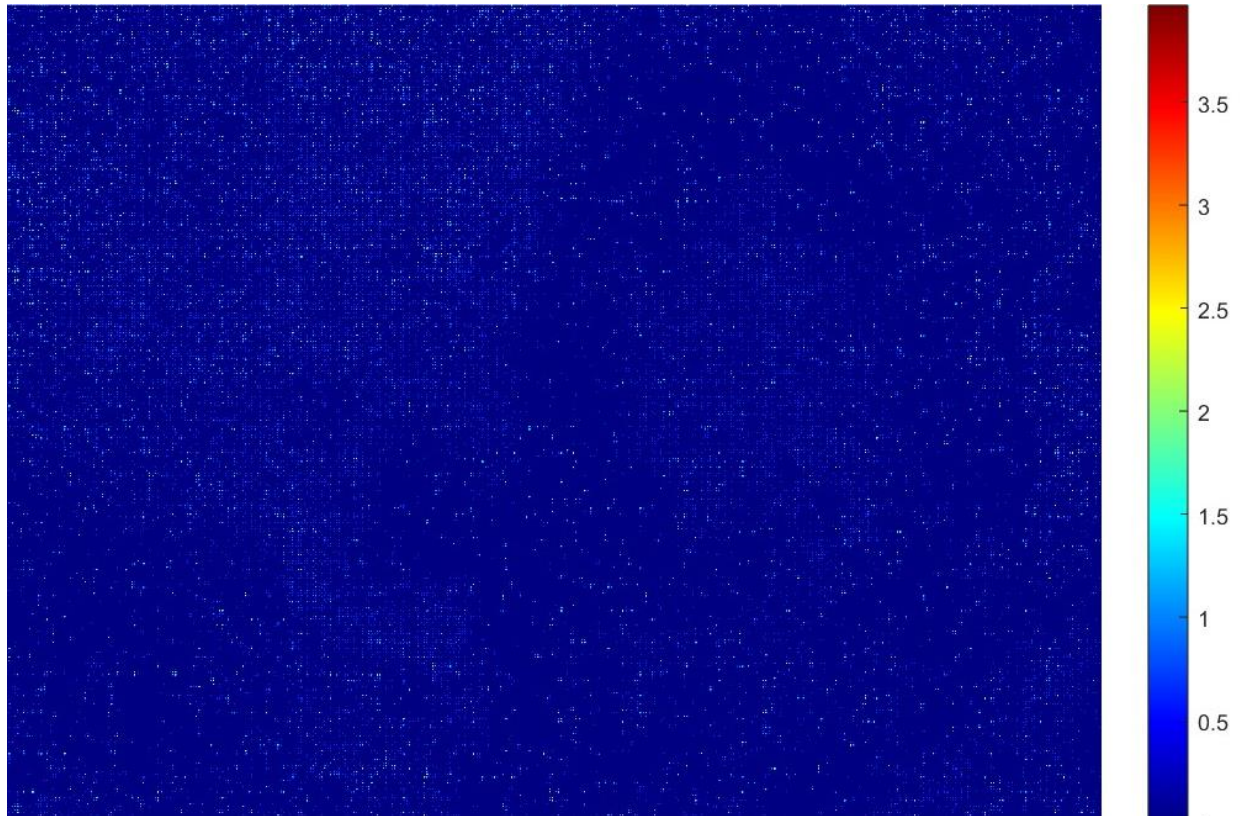
Input image



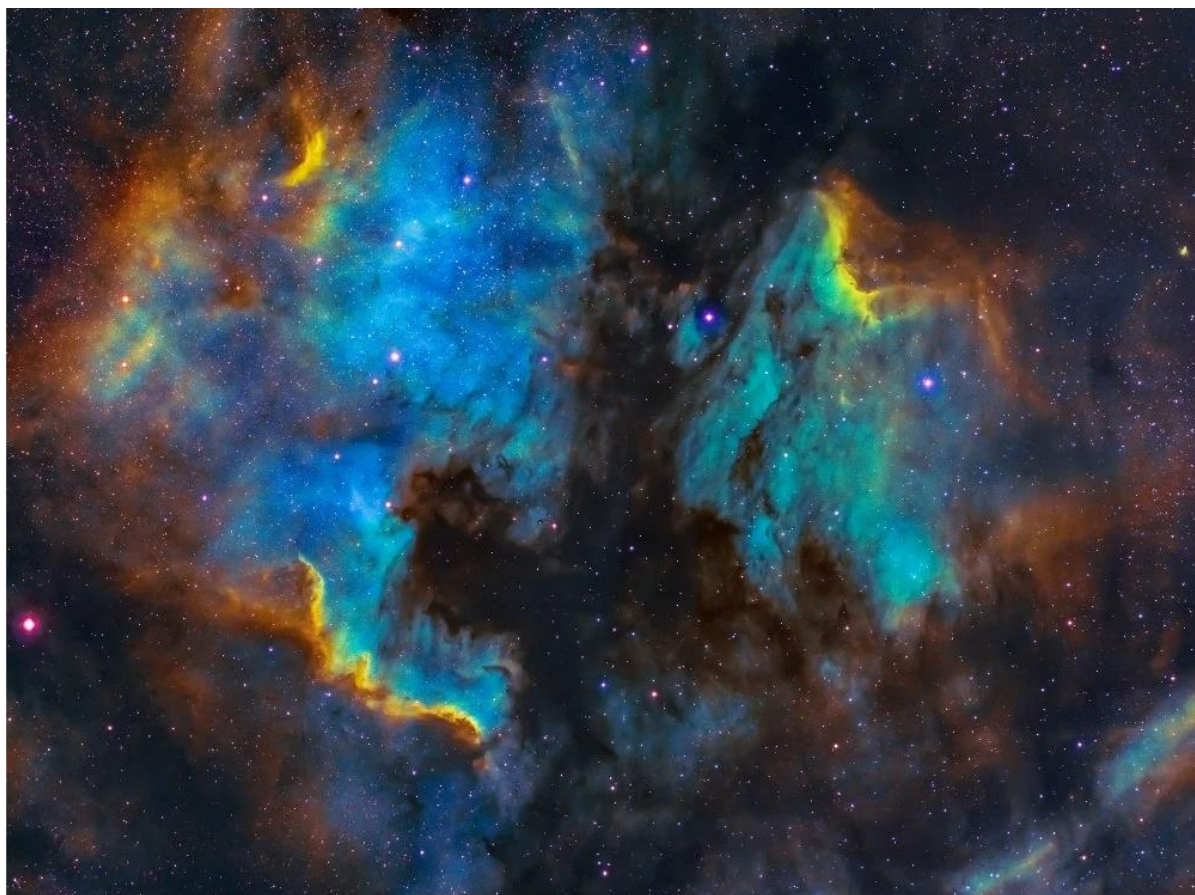
YCbCr Format



DCT Coefficients (Y)



Reconstructed Image



Result

Compression ratio – 24.4768