% verifying if the algorithm is working properly by taking 8x8 matrix

%Reading the image

I\_RGB = imread('ankit.jpg');

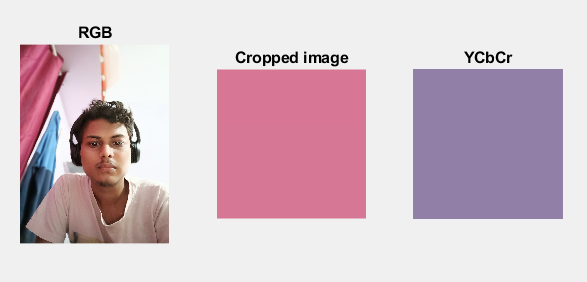
figure();

subplot(131);imshow(I\_RGB);title('RGB');

% Taking 8x8

I\_cropped=imcrop(I\_RGB,[150 500 7 7]);

subplot(132);imshow(I\_cropped);title('Cropped image');



I\_cropped(:,:,1) =

215 215 215 215 215 215 215 215

215 215 215 215 215 215 215 215

215 215 215 215 215 215 215 215

214 214 214 214 214 214 214 214

214 214 214 214 214 214 214 214

214 214 214 214 214 214 214 214

214 214 214 214 214 214 214 214

214 214 214 214 214 214 214 214

I\_cropped(:,:,2) =

118 118 118 118 118 118 118 118

118 118 118 118 118 118 118 118

118 118 118 118 118 118 118 118

119 119 119 119 119 119 119 119

119 119 119 119 119 119 119 119

119 119 119 119 119 119 119 119

119 119 119 119 119 119 119 119

119 119 119 119 119 119 119 119

I\_cropped(:,:,3) =

149 149 149 149 149 149 149 149

149 149 149 149 149 149 149 149

149 149 149 149 149 149 149 149

149 149 149 149 149 149 149 149

149 149 149 149 149 149 149 149

149 149 149 149 149 149 149 149

149 149 149 149 149 149 149 149

149 149 149 149 149 149 149 149

%conversion of RGB image to YCbCr

I\_ycbcr = rgb2ycbcr(I\_cropped);

subplot(133);imshow(I\_ycbcr);title('YCbCr');

y=I\_ycbcr(:,:,1); %all rows and colums in 1st plane i.e Y

cb=I\_ycbcr(:,:,2); %all rows and colums in 2nd plane i.e Cb

cr=I\_ycbcr(:,:,3); %all rows and colums in 3rd plane i.e Cr

figure();subplot(131);imshow(y);title('Luma');

subplot(132);imshow(cb);title('Cb');

subplot(133);imshow(cr);title('Cr');



y =

8×8 uint8 matrix

145 145 145 145 145 145 145 145

145 145 145 145 145 145 145 145

145 145 145 145 145 145 145 145

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

cb =

8×8 uint8 matrix

127 127 127 127 127 127 127 127

127 127 127 127 127 127 127 127

127 127 127 127 127 127 127 127

127 127 127 127 127 127 127 127

127 127 127 127 127 127 127 127

127 127 127 127 127 127 127 127

127 127 127 127 127 127 127 127

127 127 127 127 127 127 127 127

cr =

8×8 uint8 matrix

168 168 168 168 168 168 168 168

168 168 168 168 168 168 168 168

168 168 168 168 168 168 168 168

168 168 168 168 168 168 168 168

168 168 168 168 168 168 168 168

168 168 168 168 168 168 168 168

168 168 168 168 168 168 168 168

168 168 168 168 168 168 168 168

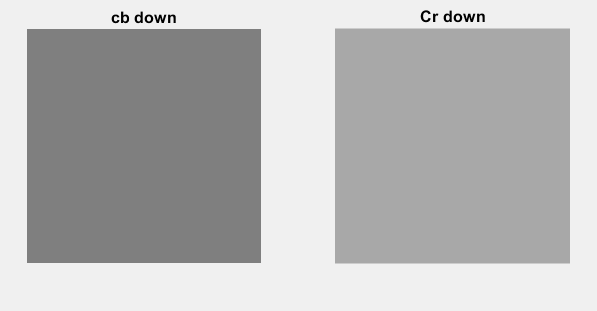
%chroma sub-sampling

cb\_down=imresize(cb,0.5,'bilinear');

cr\_down=imresize(cr,0.5,'bilinear');

figure();subplot(121);imshow(cb\_down);title('cb down');

subplot(122);imshow(cr\_down);title('Cr down');



cb\_down =

4×4 uint8 matrix

127 127 127 127

127 127 127 127

127 127 127 127

127 127 127 127

cr\_down =

4×4 uint8 matrix

168 168 168 168

168 168 168 168

168 168 168 168

168 168 168 168

%converting uint8 to double

y=double(y);

cb\_down=double(cb\_down);

cr\_down=double(cr\_down);

%Shifting the blocks to be centered around 0 rather than 128:

for i = 1:height(y)

for j = 1:width(y)

y(i,j) = y(i,j)-128;

end

end

for i = 1:height(cb\_down)

for j = 1:width(cb\_down)

cb\_down(i,j) = cb\_down(i,j)-128;

end

end

for i = 1:height(cr\_down)

for j = 1:width(cr\_down)

cr\_down(i,j) = cr\_down(i,j)-128;

end

end

figure();subplot(131);imshow(y);title('Shifted Luma');

subplot(132);imshow(cb);title('Shifted Cb');

subplot(133);imshow(cr);title('Shifted Cr');



y =

17 17 17 17 17 17 17 17

17 17 17 17 17 17 17 17

17 17 17 17 17 17 17 17

18 18 18 18 18 18 18 18

18 18 18 18 18 18 18 18

18 18 18 18 18 18 18 18

18 18 18 18 18 18 18 18

18 18 18 18 18 18 18 18

cb\_down =

-1 -1 -1 -1

-1 -1 -1 -1

-1 -1 -1 -1

-1 -1 -1 -1

cr\_down =

40 40 40 40

40 40 40 40

40 40 40 40

40 40 40 40

% zero padding

% Convert the luminance height and width to multiple of 8

if rem(size(y,1),8)~=0

y=[y;zeros(8-rem(size(y,1),8),size(y,2))];

end

if rem(size(y,2),8)~=0

y=[y zeros(size(y,1),8-rem(size(y,2),8))];

end

% Convert the chrominance height and width to multiple of 8

if rem(size(cb\_down,1),8)~=0

cb\_down=[cb\_down;zeros(8-rem(size(cb\_down,1),8),size(cb\_down,2))];

end

if rem(size(cb\_down,2),8)~=0

cb\_down=[cb\_down zeros(size(cb\_down,1),8-rem(size(cb\_down,2),8))];

end

if rem(size(cr\_down,1),8)~=0

cr\_down=[cr\_down;zeros(8-rem(size(cr\_down,1),8),size(cr\_down,2))];

end

if rem(size(cr\_down,2),8)~=0

cr\_down=[cr\_down zeros(size(cr\_down,1),8-rem(size(cr\_down,2),8))];

end

y =

17 17 17 17 17 17 17 17

17 17 17 17 17 17 17 17

17 17 17 17 17 17 17 17

18 18 18 18 18 18 18 18

18 18 18 18 18 18 18 18

18 18 18 18 18 18 18 18

18 18 18 18 18 18 18 18

18 18 18 18 18 18 18 18

cb\_down =

-1 -1 -1 -1 0 0 0 0

-1 -1 -1 -1 0 0 0 0

-1 -1 -1 -1 0 0 0 0

-1 -1 -1 -1 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

cr\_down =

40 40 40 40 0 0 0 0

40 40 40 40 0 0 0 0

40 40 40 40 0 0 0 0

40 40 40 40 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

%dividing the image into blocks and applying dct2 transform

y\_dct = blockproc(y,[8 8],@(blkStruct) dct2(blkStruct.data))

cb\_down\_dct = blockproc(cb\_down,[8 8],@(blkStruct) dct2(blkStruct.data))

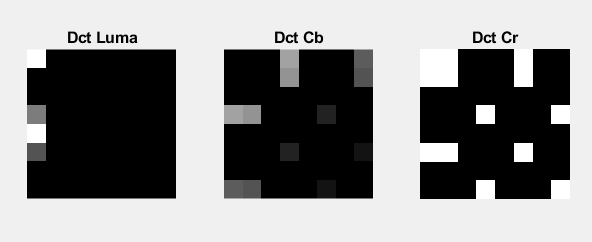
cr\_down\_dct = blockproc(cr\_down,[8 8],@(blkStruct) dct2(blkStruct.data))

figure();

subplot(131);imshow(y\_dct);title('Dct Luma');

subplot(132);imshow(cb\_down\_dct);title('Dct Cb');

subplot(133);imshow(cr\_down\_dct);title('Dct Cr');



y\_dct =

141.0000 0 0 0 0 0 0 0

-3.3486 0 0 0 0 0 0 0

-1.3066 0 0 0 0 0 0 0

0.4871 0 0 0 0 0 0 0

1.0000 0 0 0 0 0 0 0

0.3254 0 0 0 0 0 0 0

-0.5412 0 0 0 0 0 0 0

-0.6661 0 0 0 0 0 0 0

cb\_down\_dct =

-2.0000 -1.8123 0 0.6364 0 -0.4252 0 0.3605

-1.8123 -1.6421 0 0.5766 0 -0.3853 0 0.3266

0 0 0 0 0 0 0 0

0.6364 0.5766 0 -0.2025 0 0.1353 0 -0.1147

0 0 0 0 0 0 0 0

-0.4252 -0.3853 0 0.1353 0 -0.0904 0 0.0766

0 0 0 0 0 0 0 0

0.3605 0.3266 0 -0.1147 0 0.0766 0 -0.0650

cr\_down\_dct =

80.0000 72.4902 0 -25.4552 0 17.0086 0 -14.4192

72.4902 65.6854 0 -23.0656 0 15.4120 0 -13.0656

0 0 0 0 0 0 0 0

-25.4552 -23.0656 0 8.0996 0 -5.4120 0 4.5880

0 0 0 0 0 0 0 0

17.0086 15.4120 0 -5.4120 0 3.6162 0 -3.0656

0 0 0 0 0 0 0 0

-14.4192 -13.0656 0 4.5880 0 -3.0656 0 2.5989

% Initialization of quantization matrices for chrominance and luminance

% Quant luminance

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 ] ;

% Quant chrominance

Q\_c = [ 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 ] ;

%Quantization of Luminance and Chrominance:

quant\_y = @(blockStruct) round((blockStruct.data)./Q\_y);

quant\_c = @(blockStruct) round((blockStruct.data)./Q\_c);

y\_dct\_quant = blockproc(y\_dct, [8 8], quant\_y)

cb\_down\_dct\_quant = blockproc(cb\_down\_dct, [8 8], quant\_c)

cr\_down\_dct\_quant = blockproc(cr\_down\_dct, [8 8], quant\_c)

figure();

subplot(131);imshow(y\_dct\_quant);title('Dct Luma quantized');

subplot(132);imshow(cb\_down\_dct\_quant);title('Dct Cb quantized');

subplot(133);imshow(cr\_down\_dct\_quant);title('Dct Cr quantized');



y\_dct\_quant =

9 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

cb\_down\_dct\_quant =

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

cr\_down\_dct\_quant =

5 4 0 -1 0 0 0 0

4 3 0 0 0 0 0 0

0 0 0 0 0 0 0 0

-1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

%Zig-zag coding

function output = zigzag(in)

% initializing the variables

%----------------------------------

h = 1;

v = 1;

vmin = 1;

hmin = 1;

vmax = size(in, 1);

hmax = size(in, 2);

i = 1;

output = zeros(1, vmax \* hmax);

%----------------------------------

while ((v <= vmax) && (h <= hmax))

if (mod(h + v, 2) == 0) % going up

if (v == vmin)

output(i) = in(v, h); % if we got to the first line

if (h == hmax)

v = v + 1;

else

h = h + 1;

end

i = i + 1;

elseif ((h == hmax) && (v < vmax)) % if we got to the last column

output(i) = in(v, h);

v = v + 1;

i = i + 1;

elseif ((v > vmin) && (h < hmax)) % all other cases

output(i) = in(v, h);

v = v - 1;

h = h + 1;

i = i + 1;

end

else % going down

if ((v == vmax) && (h <= hmax)) % if we got to the last line

output(i) = in(v, h);

h = h + 1;

i = i + 1;

elseif (h == hmin) % if we got to the first column

output(i) = in(v, h);

if (v == vmax)

h = h + 1;

else

v = v + 1;

end

i = i + 1;

elseif ((v < vmax) && (h > hmin)) % all other cases

output(i) = in(v, h);

v = v + 1;

h = h - 1;

i = i + 1;

end

end

if ((v == vmax) && (h == hmax)) % bottom right element

output(i) = in(v, h);

break

end

end

zigzag\_y=zigzag(y\_dct\_quant)

zigzag\_cb=zigzag(cb\_down\_dct\_quant);

zigzag\_cr=zigzag(cr\_down\_dct\_quant);

zigzag\_y =

Columns 1 through 17

9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 18 through 34

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 35 through 51

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 52 through 64

0 0 0 0 0 0 0 0 0 0 0 0 0

zigzag\_cb =

Columns 1 through 17

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 18 through 34

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 35 through 51

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 52 through 64

0 0 0 0 0 0 0 0 0 0 0 0 0

zigzag\_cr =

Columns 1 through 17

5 4 4 0 3 0 -1 0 0 -1 0 0 0 0 0 0 0

Columns 18 through 34

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 35 through 51

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 52 through 64

0 0 0 0 0 0 0 0 0 0 0 0 0

%Run Length Encoding

function Output=rle(Input)

L=length(Input);

j=1;

k=1;

i=1;

while i<2\*L

comp=1;

for j=j:L

if j==L

break

end

if Input(j)==Input(j+1)

comp=comp+1;

else

break

end

end

Output(k+1)=comp;

Output(k)=Input(j);

if j==L && Input(j-1)==Input(j)

break

end

i=i+1;

k=k+2;

j=j+1;

if j==L

if mod(L,2)==0

Output(k+1)=1;

Output(k)=Input(j);

else

Output(k+1)=1;

Output(k)=Input(j);

end

break

end

end

y\_rle=rle(zigzag\_y);

cb\_rle=rle(zigzag\_cb);

cr\_rle=rle(zigzag\_cr);

y\_rle =

9 1 0 63

cb\_rle =

0 64

cr\_rle =

Columns 1 through 17

5 1 4 2 0 1 3 1 0 1 -1 1 0 2 -1 1 0

Column 18

54

%%%%%%%%%%%%%%%%%%%%%%%%%% Decompression %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Inverse run-length encoding

function Output=irle(Input)

L=length(Input);

s=1;

k=1;

i=1;

while i<=L

while s<=Input(i+1)

Output(k)=Input(i);

s=s+1;

k=k+1;

end

i=i+2;

s=1;

end

y\_irle=irle(y\_rle);

cb\_irle=irle(cb\_rle);

cr\_irle=irle(cr\_rle);

y\_irle =

Columns 1 through 17

9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 18 through 34

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 35 through 51

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 52 through 64

0 0 0 0 0 0 0 0 0 0 0 0 0

cb\_irle =

Columns 1 through 17

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 18 through 34

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 35 through 51

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 52 through 64

0 0 0 0 0 0 0 0 0 0 0 0 0

cr\_irle =

Columns 1 through 17

5 4 4 0 3 0 -1 0 0 -1 0 0 0 0 0 0 0

Columns 18 through 34

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 35 through 51

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 52 through 64

0 0 0 0 0 0 0 0 0 0 0 0 0

%Inverse zigzag encoding

function output = izigzag(in, vmax, hmax)

% initializing the variables

%----------------------------------

h = 1;

v = 1;

vmin = 1;

hmin = 1;

output = zeros(vmax, hmax);

i = 1;

%----------------------------------

while ((v <= vmax) && (h <= hmax))

if (mod(h + v, 2) == 0) % going up

if (v == vmin)

output(v, h) = in(i);

if (h == hmax)

v = v + 1;

else

h = h + 1;

end

i = i + 1;

elseif ((h == hmax) && (v < vmax))

output(v, h) = in(i);

v = v + 1;

i = i + 1;

elseif ((v > vmin) && (h < hmax))

output(v, h) = in(i);

v = v - 1;

h = h + 1;

i = i + 1;

end

else % going down

if ((v == vmax) && (h <= hmax))

output(v, h) = in(i);

h = h + 1;

i = i + 1;

elseif (h == hmin)

output(v, h) = in(i);

if (v == vmax)

h = h + 1;

else

v = v + 1;

end

i = i + 1;

elseif ((v < vmax) && (h > hmin))

output(v, h) = in(i);

v = v + 1;

h = h - 1;

i = i + 1;

end

end

if ((v == vmax) && (h == hmax))

output(v, h) = in(i);

break

end

end

y\_izigzag=izigzag(y\_irle,size(y\_dct\_quant,1),size(y\_dct\_quant,2));

cb\_izigzag=izigzag(cb\_irle,size(cb\_down\_dct\_quant,1),size(cb\_down\_dct\_quant,2));

cr\_izigzag=izigzag(cr\_irle,size(cr\_down\_dct\_quant,1),size(cr\_down\_dct\_quant,2));

y\_izigzag =

9 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

cb\_izigzag =

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

cr\_izigzag =

5 4 0 -1 0 0 0 0

4 3 0 0 0 0 0 0

0 0 0 0 0 0 0 0

-1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

%Dequantization

y\_dequant = blockproc(y\_izigzag,[8 8],@(blockStruct) round((blockStruct.data).\*Q\_y));

cb\_dequant = blockproc(cb\_izigzag,[8 8],@(blockStruct) round((blockStruct.data).\*Q\_c));

cr\_dequant = blockproc(cr\_izigzag,[8 8],@(blockStruct) round((blockStruct.data).\*Q\_c));

figure();

subplot(131);imshow(y\_dequant);title('dequantized y');

subplot(132);imshow(cb\_dequant);title('dequantized cb');

subplot(133);imshow(cr\_dequant);title('dequantized cr');



y\_dequant =

144 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

cb\_dequant =

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

cr\_dequant =

85 72 0 -47 0 0 0 0

72 63 0 0 0 0 0 0

0 0 0 0 0 0 0 0

-47 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

% Applying idct

y\_idct = blockproc(y\_dequant,[8 8],@(blockStruct) idct2(blockStruct.data));

cb\_idct = blockproc(cb\_dequant,[8 8],@(blockStruct) idct2(blockStruct.data));

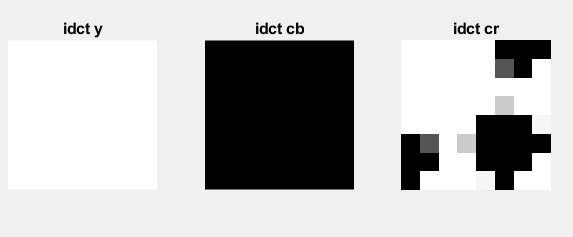
cr\_idct = blockproc(cr\_dequant,[8 8],@(blockStruct) idct2(blockStruct.data));

figure();

subplot(131);imshow(y\_idct);title('idct y');

subplot(132);imshow(cb\_idct);title('idct cb');

subplot(133);imshow(cr\_idct);title('idct cr');



%Shifting the blocks back

for i = 1:height(y\_idct)

for j = 1:width(y\_idct)

y\_idct(i,j) = y\_idct(i,j)+128;

end

end

for i = 1:height(cb\_idct)

for j = 1:width(cb\_idct)

cb\_idct(i,j) = cb\_idct(i,j)+128;

end

end

for i = 1:height(cr\_idct)

for j = 1:width(cr\_idct)

cr\_idct(i,j) = cr\_idct(i,j)+128;

end

end

y\_idct =

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

cb\_idct =

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

cr\_idct =

164.9257 169.2479 168.0023 154.3128 134.0874 120.3979 119.1523 123.4744

169.2479 173.9212 173.3245 160.4827 141.1749 128.3331 127.7364 132.4097

168.0023 173.3245 173.9266 162.6513 145.0390 133.7636 134.3658 139.6879

154.3128 160.4827 162.6513 153.4226 138.0256 128.7969 130.9654 137.1353

134.0874 141.1749 145.0390 138.0256 125.0263 118.0129 121.8770 128.9645

120.3979 128.3331 133.7636 128.7969 118.0129 113.0461 118.4766 126.4119

119.1523 127.7364 134.3658 130.9654 121.8770 118.4766 125.1061 133.6901

123.4744 132.4097 139.6879 137.1353 128.9645 126.4119 133.6901 142.6254

% Chroma up sampling

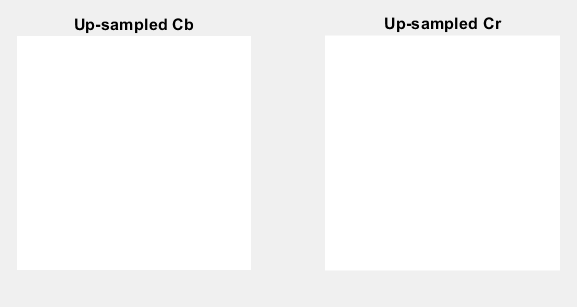
cb\_up\_sampling = imresize(cb\_idct,2,'bilinear');

cr\_up\_sampling = imresize(cr\_idct,2,'bilinear');

figure();

subplot(121);imshow(cb\_up\_sampling);title('Up-sampled Cb');

subplot(122);imshow(cr\_up\_sampling);title('Up-sampled Cr');



cb\_up\_sampling =

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128 128 128 128 128 128 128 128 128

cr\_up\_sampling =

Columns 1 through 10

164.9257 166.0063 168.1674 168.9365 168.3137 164.5799 157.7352 149.2564 139.1437 130.6650

166.0063 167.1088 169.3137 170.1454 169.6037 165.9634 159.2246 150.8563 140.8583 132.4899

168.1674 169.3137 171.6065 172.5631 172.1837 168.7305 162.2036 154.0559 144.2873 136.1396

168.9365 170.1454 172.5631 173.6978 173.5492 170.3624 164.1374 156.3038 146.8619 139.0284

168.3137 169.6037 172.1837 173.5492 173.7005 170.8593 165.0258 157.6001 148.5820 141.1562

164.5799 165.9634 168.7305 170.3624 170.8593 168.4168 163.0350 156.0795 147.5502 140.5947

157.7352 159.2246 162.2036 164.1374 165.0258 163.0350 158.1648 151.7420 143.7666 137.3438

149.2564 150.8563 154.0559 156.3038 157.6001 156.0795 151.7420 145.8739 138.4751 132.6070

139.1437 140.8583 144.2873 146.8619 148.5820 147.5502 143.7666 138.4751 131.6758 126.3843

130.6650 132.4899 136.1396 139.0284 141.1562 140.5947 137.3438 132.6070 126.3843 121.6475

123.8203 125.7511 129.6127 132.8033 135.3227 135.2129 132.4736 128.2696 122.6007 118.3967

120.0865 122.1108 126.1596 129.6165 132.4816 132.7704 130.4828 126.7490 121.5690 117.8351

119.4637 121.5692 125.7801 129.4680 132.6328 133.2672 131.3713 128.0452 123.2891 119.9630

120.2328 122.4008 126.7367 130.6026 133.9984 134.8992 133.3050 130.2931 125.8636 122.8518

122.3939 124.6058 129.0295 133.0204 136.5784 137.6663 136.2840 133.4928 129.2927 126.5015

123.4744 125.7083 130.1759 134.2292 137.8684 139.0498 137.7735 135.0926 131.0072 128.3263

Columns 11 through 16

123.8203 120.0865 119.4637 120.2328 122.3939 123.4744

125.7511 122.1108 121.5692 122.4008 124.6058 125.7083

129.6127 126.1596 125.7801 126.7367 129.0295 130.1759

132.8033 129.6165 129.4680 130.6026 133.0204 134.2292

135.3227 132.4816 132.6328 133.9984 136.5784 137.8684

135.2129 132.7704 133.2672 134.8992 137.6663 139.0498

132.4736 130.4828 131.3713 133.3050 136.2840 137.7735

128.2696 126.7490 128.0452 130.2931 133.4928 135.0926

122.6007 121.5690 123.2891 125.8636 129.2927 131.0072

118.3967 117.8351 119.9630 122.8518 126.5015 128.3263

115.6574 115.5476 118.0670 121.2576 125.1192 127.0500

115.5476 115.8363 118.7014 122.1584 126.2071 128.2315

118.0670 118.7014 121.8663 125.5542 129.7651 131.8706

121.2576 122.1584 125.5542 129.4200 133.7560 135.9239

125.1192 126.2071 129.7651 133.7560 138.1797 140.3916

127.0500 128.2315 131.8706 135.9239 140.3916 142.6254

%Reconstructing Luminance and Chrominance of same size:

y\_reconstruct = y\_idct(1:height(y), 1:width(y));

cb\_up\_reconstruct = cb\_up\_sampling(1:height(cb), 1:width(cb));

cr\_up\_reconstruct = cr\_up\_sampling(1:height(cr), 1:width(cr));

y\_reconstruct =

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

cb\_up\_reconstruct =

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

cr\_up\_reconstruct =

164.9257 166.0063 168.1674 168.9365 168.3137 164.5799 157.7352 149.2564

166.0063 167.1088 169.3137 170.1454 169.6037 165.9634 159.2246 150.8563

168.1674 169.3137 171.6065 172.5631 172.1837 168.7305 162.2036 154.0559

168.9365 170.1454 172.5631 173.6978 173.5492 170.3624 164.1374 156.3038

168.3137 169.6037 172.1837 173.5492 173.7005 170.8593 165.0258 157.6001

164.5799 165.9634 168.7305 170.3624 170.8593 168.4168 163.0350 156.0795

157.7352 159.2246 162.2036 164.1374 165.0258 163.0350 158.1648 151.7420

149.2564 150.8563 154.0559 156.3038 157.6001 156.0795 151.7420 145.8739

% YCbCr reconstruction

YCbCr\_reconstruct(:,:,1)=y\_reconstruct;

YCbCr\_reconstruct(:,:,2)=cb\_up\_reconstruct;

YCbCr\_reconstruct(:,:,3)=cr\_up\_reconstruct;

YCbCr\_reconstruct =uint8(YCbCr\_reconstruct);

figure();

imshow(YCbCr\_reconstruct);

title('Reconstructed YCbCr');

YCbCr\_reconstruct(:,:,1) =

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

146 146 146 146 146 146 146 146

YCbCr\_reconstruct(:,:,2) =

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

128 128 128 128 128 128 128 128

YCbCr\_reconstruct(:,:,3) =

165 166 168 169 168 165 158 149

166 167 169 170 170 166 159 151

168 169 172 173 172 169 162 154

169 170 173 174 174 170 164 156

168 170 172 174 174 171 165 158

165 166 169 170 171 168 163 156

158 159 162 164 165 163 158 152

149 151 154 156 158 156 152 146

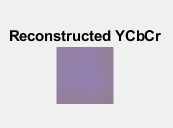
% Convert ycbcr back to RGB

I\_reconstructed=uint8(ycbcr2rgb(YCbCr\_reconstruct));

figure();

imshow(I\_reconstructed);

title('Reconstructed Image');



I\_reconstructed

8×8×3 uint8 array

I\_reconstructed(:,:,1) =

210 212 215 217 215 210 199 185

212 214 217 218 218 212 201 188

215 217 222 223 222 217 206 193

217 218 223 225 225 218 209 196

215 218 222 225 225 220 210 199

210 212 217 218 220 215 207 196

199 201 206 209 210 207 199 190

185 188 193 196 199 196 190 180

I\_reconstructed(:,:,2) =

121 120 119 118 119 121 127 134

120 120 118 117 117 120 126 133

119 118 116 115 116 118 124 130

118 117 115 114 114 117 122 129

119 117 116 114 114 116 121 127

121 120 118 117 116 119 123 129

127 126 124 122 121 123 127 132

134 133 130 129 127 129 132 137

I\_reconstructed(:,:,3) =

151 151 151 151 151 151 151 151

151 151 151 151 151 151 151 151

151 151 151 151 151 151 151 151

151 151 151 151 151 151 151 151

151 151 151 151 151 151 151 151

151 151 151 151 151 151 151 151

151 151 151 151 151 151 151 151

151 151 151 151 151 151 151 151

% Now taking the whole image

clear; close all; clc;

%Reading the image

I\_RGB = imread('ankit.jpg');

figure();

subplot(121);imshow(I\_RGB);title('RGB');

%conversion of RGB image to YCbCr

I\_ycbcr = rgb2ycbcr(I\_RGB);

subplot(122);imshow(I\_ycbcr);title('YCbCr');

y=I\_ycbcr(:,:,1); %all rows and colums in 1st plane i.e Y

cb=I\_ycbcr(:,:,2); %all rows and colums in 2nd plane i.e Cb

cr=I\_ycbcr(:,:,3); %all rows and colums in 3rd plane i.e Cr

figure();subplot(131);imshow(y);title('Luma');

subplot(132);imshow(cb);title('Cb');

subplot(133);imshow(cr);title('Cr');

%chroma sub-sampling

cb\_down=imresize(cb,0.5,'bilinear');

cr\_down=imresize(cr,0.5,'bilinear');

figure();subplot(121);imshow(cb\_down);title('cb down');

subplot(122);imshow(cr\_down);title('Cr down');

%converting uint8 to double

y=double(y);

cb\_down=double(cb\_down);

cr\_down=double(cr\_down);

%Shifting the blocks to be centered around 0 rather than 128:

for i = 1:height(y)

for j = 1:width(y)

y(i,j) = y(i,j)-128;

end

end

for i = 1:height(cb\_down)

for j = 1:width(cb\_down)

cb\_down(i,j) = cb\_down(i,j)-128;

end

end

for i = 1:height(cr\_down)

for j = 1:width(cr\_down)

cr\_down(i,j) = cr\_down(i,j)-128;

end

end

y;

cb\_down;

cr\_down;

figure();subplot(131);imshow(y);title('Shifted Luma');

subplot(132);imshow(cb);title('Shifted Cb');

subplot(133);imshow(cr);title('Shifted Cr');

% zero padding

% Convert the luminance height and width to multiple of 8

if rem(size(y,1),8)~=0

y=[y;zeros(8-rem(size(y,1),8),size(y,2))];

end

if rem(size(y,2),8)~=0

y=[y zeros(size(y,1),8-rem(size(y,2),8))];

end

% Convert the chrominance height and width to multiple of 8

if rem(size(cb\_down,1),8)~=0

cb\_down=[cb\_down;zeros(8-rem(size(cb\_down,1),8),size(cb\_down,2))];

end

if rem(size(cb\_down,2),8)~=0

cb\_down=[cb\_down zeros(size(cb\_down,1),8-rem(size(cb\_down,2),8))];

end

if rem(size(cr\_down,1),8)~=0

cr\_down=[cr\_down;zeros(8-rem(size(cr\_down,1),8),size(cr\_down,2))];

end

if rem(size(cr\_down,2),8)~=0

cr\_down=[cr\_down zeros(size(cr\_down,1),8-rem(size(cr\_down,2),8))];

end

%dividing the image into blocks and applying dct2 transform

y\_dct = blockproc(y,[8 8],@(blkStruct) dct2(blkStruct.data));

cb\_down\_dct = blockproc(cb\_down,[8 8],@(blkStruct) dct2(blkStruct.data));

cr\_down\_dct = blockproc(cr\_down,[8 8],@(blkStruct) dct2(blkStruct.data));

figure();

subplot(131);imshow(y\_dct);title('Dct Luma');

subplot(132);imshow(cb\_down\_dct);title('Dct Cb');

subplot(133);imshow(cr\_down\_dct);title('Dct Cr');

% Initialization of quantization matrices for chrominance and luminance

% Quant luminance

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 ] ;

% Quant chrominance

Q\_c = [ 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 ] ;

%Quantization of Luminance and Chrominance:

quant\_y = @(blockStruct) round((blockStruct.data)./Q\_y);

quant\_c = @(blockStruct) round((blockStruct.data)./Q\_c);

y\_dct\_quant = blockproc(y\_dct, [8 8], quant\_y);

cb\_down\_dct\_quant = blockproc(cb\_down\_dct, [8 8], quant\_c);

cr\_down\_dct\_quant = blockproc(cr\_down\_dct, [8 8], quant\_c);

figure();

subplot(131);imshow(y\_dct\_quant);title('Dct Luma quantized');

subplot(132);imshow(cb\_down\_dct\_quant);title('Dct Cb quantized');

subplot(133);imshow(cr\_down\_dct\_quant);title('Dct Cr quantized');

%Dequantization

y\_dequant = blockproc(y\_dct\_quant,[8 8],@(blockStruct) round((blockStruct.data).\*Q\_y));

cb\_dequant = blockproc(cb\_down\_dct\_quant,[8 8],@(blockStruct) round((blockStruct.data).\*Q\_c));

cr\_dequant = blockproc(cr\_down\_dct\_quant,[8 8],@(blockStruct) round((blockStruct.data).\*Q\_c));

figure();

subplot(131);imshow(y\_dequant);title('dequantized y');

subplot(132);imshow(cb\_dequant);title('dequantized cb');

subplot(133);imshow(cr\_dequant);title('dequantized cr');

% Applying idct

y\_idct = blockproc(y\_dequant,[8 8],@(blockStruct) idct2(blockStruct.data));

cb\_idct = blockproc(cb\_dequant,[8 8],@(blockStruct) idct2(blockStruct.data));

cr\_idct = blockproc(cr\_dequant,[8 8],@(blockStruct) idct2(blockStruct.data));

figure();

subplot(131);imshow(y\_idct);title('idct y');

subplot(132);imshow(cb\_idct);title('idct cb');

subplot(133);imshow(cr\_idct);title('idct cr');

%Shifting the blocks back

for i = 1:height(y\_idct)

for j = 1:width(y\_idct)

y\_idct(i,j) = y\_idct(i,j)+128;

end

end

for i = 1:height(cb\_idct)

for j = 1:width(cb\_idct)

cb\_idct(i,j) = cb\_idct(i,j)+128;

end

end

for i = 1:height(cr\_idct)

for j = 1:width(cr\_idct)

cr\_idct(i,j) = cr\_idct(i,j)+128;

end

end

% Chroma up sampling

cb\_up\_sampling = imresize(cb\_idct,2,'bilinear');

cr\_up\_sampling = imresize(cr\_idct,2,'bilinear');

figure();

subplot(121);imshow(cb\_up\_sampling);title('Up-sampled Cb');

subplot(122);imshow(cr\_up\_sampling);title('Up-sampled Cr');

%Reconstructing Luminance and Chrominance of same size:

y\_reconstruct = y\_idct(1:height(y), 1:width(y));

cb\_up\_reconstruct = cb\_up\_sampling(1:height(cb), 1:width(cb));

cr\_up\_reconstruct = cr\_up\_sampling(1:height(cr), 1:width(cr));

% YCbCr reconstruction

YCbCr\_reconstruct(:,:,1)=y\_reconstruct;

YCbCr\_reconstruct(:,:,2)=cb\_up\_reconstruct;

YCbCr\_reconstruct(:,:,3)=cr\_up\_reconstruct;

YCbCr\_reconstruct =uint8(YCbCr\_reconstruct);

figure();

imshow(YCbCr\_reconstruct);

title('Reconstructed YCbCr');

% Convert ycbcr back to RGB

I\_reconstructed=uint8(ycbcr2rgb(YCbCr\_reconstruct));

figure();

imshow(I\_reconstructed);

title('Reconstructed Image');





