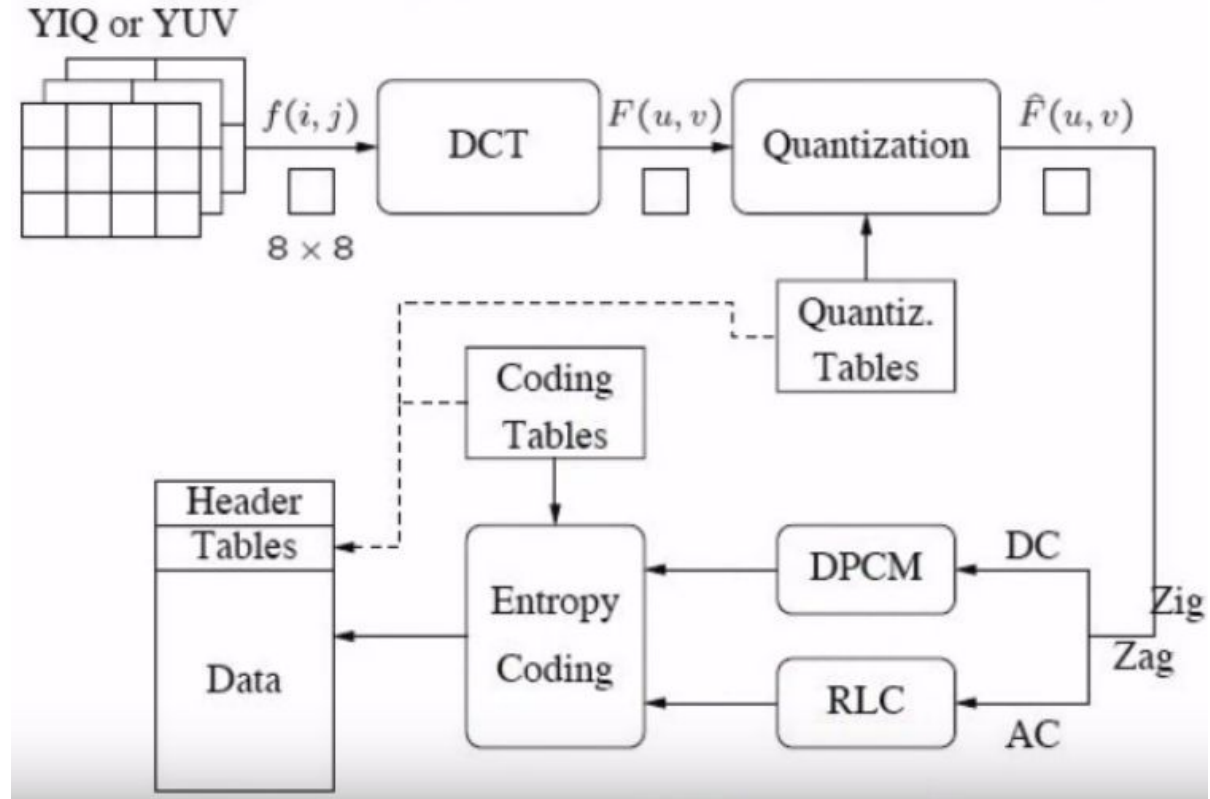


Jpeg

Algoritma

1. RGB→YUV dönüşümü yapılır.
2. Görüntü 8x8 bloklara bölünür.
3. Her bloka DFT uygulanır.
4. Niceleme (Quantization)
5. Zig zag sıralama
6. RLC (Run length coding)
7. Entropy Coding



Adım1-RGB2YUV

RGB → YUV

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = 0.492 (B-Y)$$

$$V = 0.877 (R-Y)$$

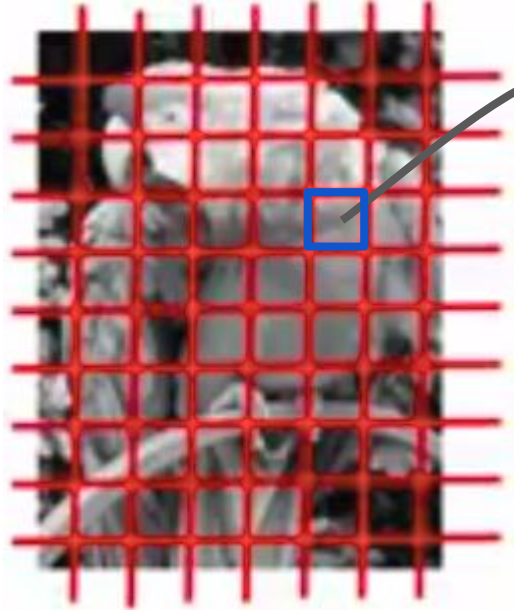
YUV → RGB

$$R = Y + 1.140V$$

$$G = Y - 0.395U - 0.581V$$

$$B = Y + 2.032U$$

Adım2-8x8 blokları elde etme



Arrow from the highlighted block in the image points to this table.

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Arrow labeled -128 points from the table above to the table below.

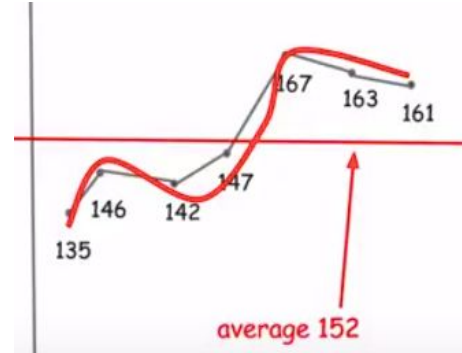
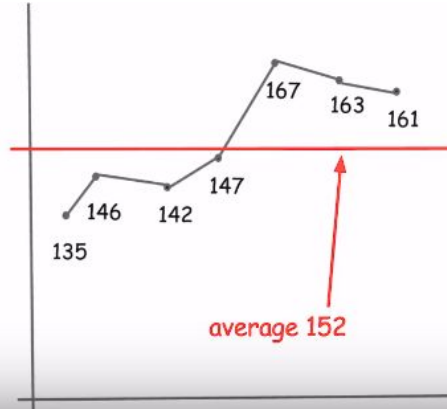
-128

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-73	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-60	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

Adım3-DCT

1D

00	25	50	48	47	42	40
10	100	240	220	210	185	50
08	120	190	220	205	201	190
85	150	148	147	152	163	165
135	146	142	147	167	163	161
84	143	152	147	167	160	161
89	135	162	142	157	150	151
101	121	151	140	152	138	131
101	115	136	161	124	89	42

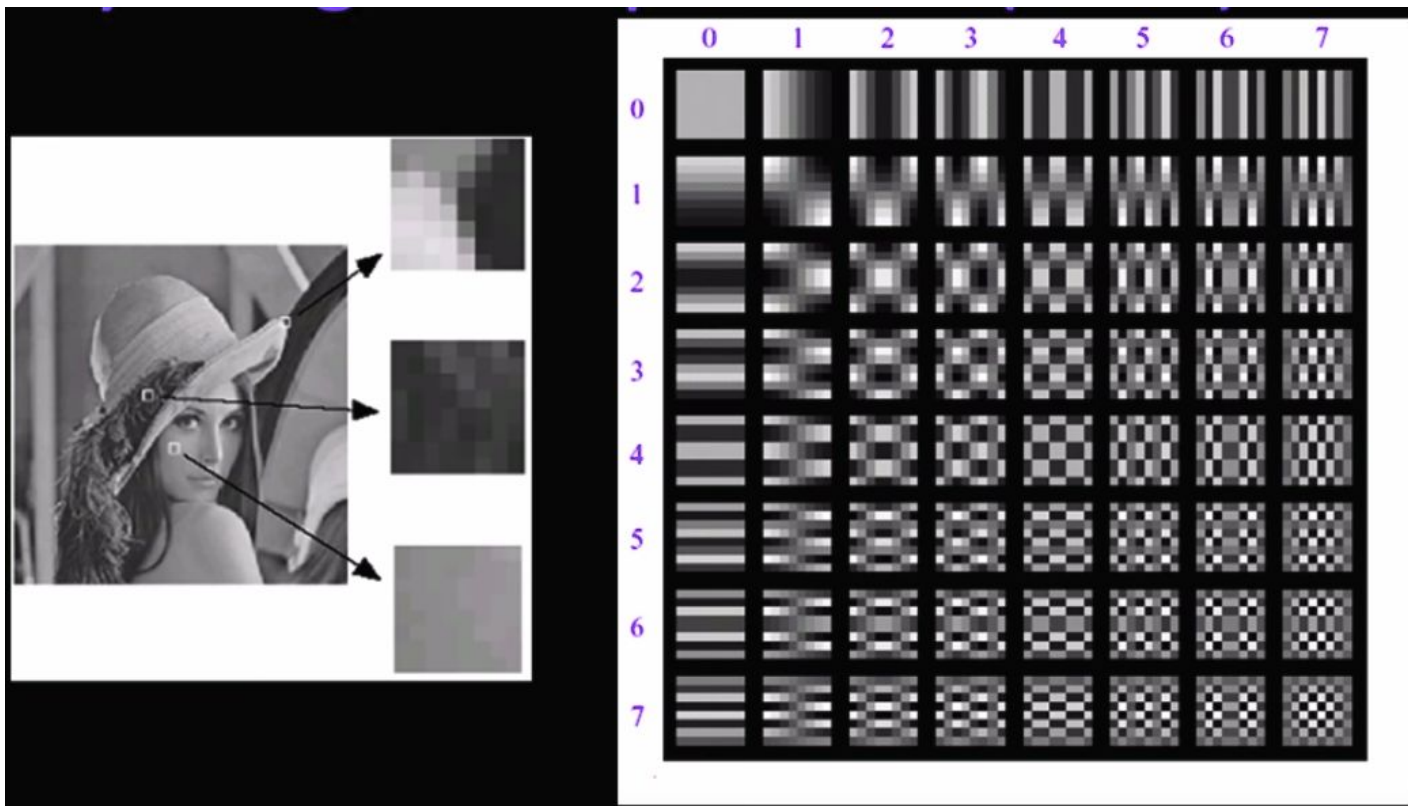


$$152 + 10 \cos(X) - 5 \sin(x)$$

135	146	142	147	167	163	161
-----	-----	-----	-----	-----	-----	-----

152	10	-5
-----	----	----

2D



Adım4-Quantization

$$\hat{c}[k_1, k_2] = \text{round}(c[k_1, k_2] / Q[k_1, k_2])$$

Quantize edilmiş DCT katsayıları

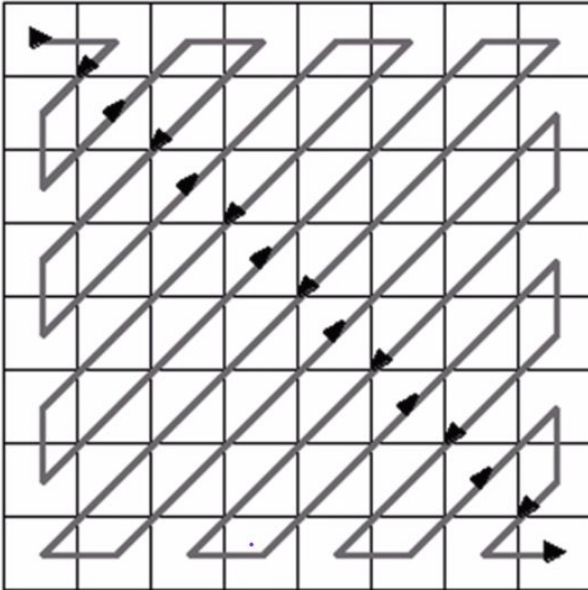
DCT katsayıları

$Q =$

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Adım5-ZigZag sıralama

DCT matrisin (1,1) noktası en düşük, (8,8) noktası en yüksek frekanslı sinüzoidin katsayısını içermektedir. Katsayıları düşükten yükseğe doğru sıralamak istediğimizde ZigZag bir ilerlemeye ihtiyaç olduğu görülür.



$$\begin{bmatrix} 100 & -60 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 13 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

100, -60, 0, 0, 0, 0, 6, 0, 0, 0, 13, 0, 0, 0, 0, 0, 0, 0, 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Adım6- RLC (Run Length Coding)

Her bir sıfır olmayan değer aşağıdaki gibi üçlü bir şekilde kodlanır.

$$[(r,s),c]$$

r (runlength)	: Değer öncesindeki sıfır sayısı
s (size)	: Değeri kodlamak için gerekli olan bit sayısı
c	: Değer
(0,0)	: Sona geldiğini ifade eder.

Adim6- RLC (Run Length Coding)

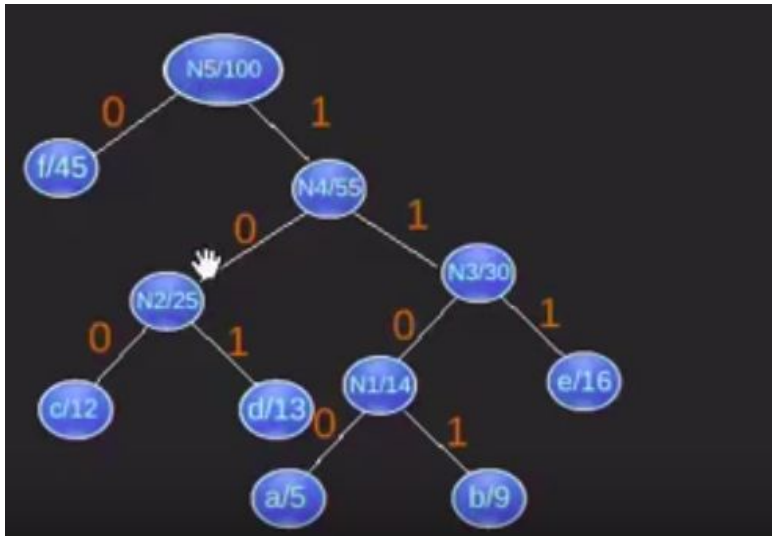
$$\begin{bmatrix} 100 & -60 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 13 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$[(0, 7), 100], [(0, 6), -60], [(4, 3), 6], [(3, 4), 13], [(8, 1), -1], [(0, 0)]$

Adım7- Entropy Coding (Huffman)

Amaç: minimum mesaj uzunluğu. Frekansı yüksek olan sembollere düşük bit dizisi ata.

Örnek) aabddccadecedcaeddcbbbebdbffdbfcbea...



Character	Code	Frequency
a	1100	5
b	1101	9
c	100	12
d	101	13
e	111	16
f	0	45

Örnek

Orjinal



89k

JPG



12k