

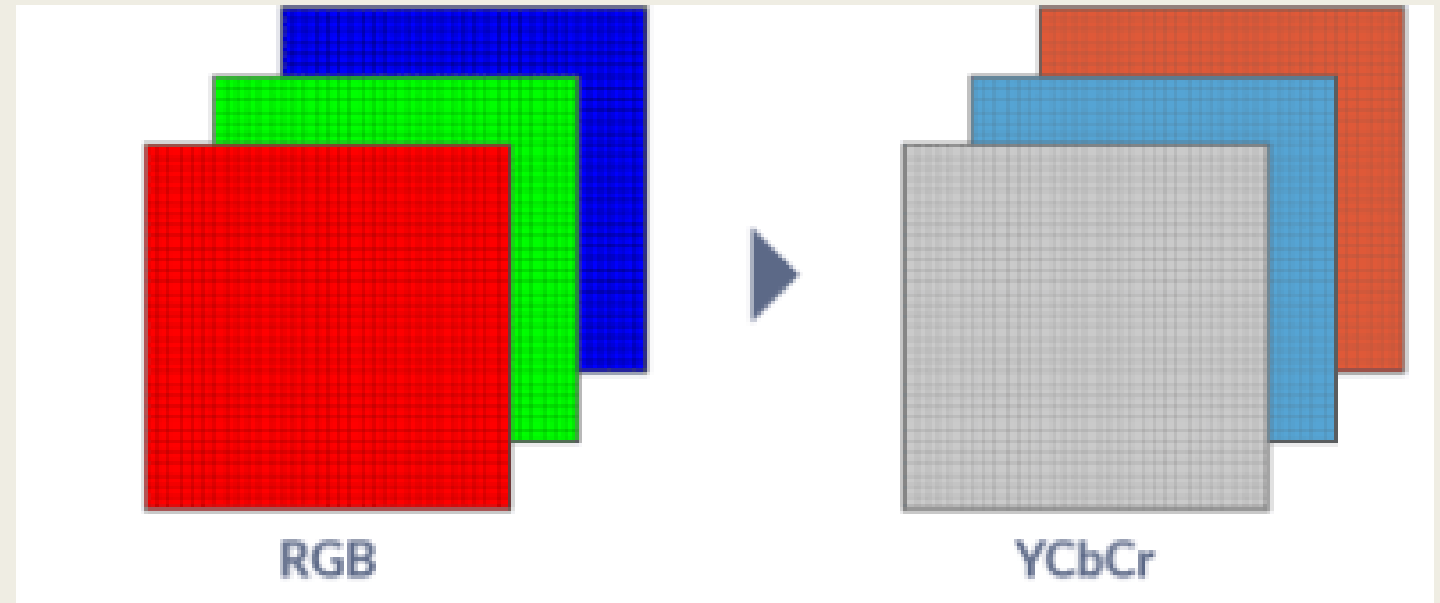
# МОДУЛЬ АРХИВАЦИИ JPEG ФАЙЛОВ

Автор: Кулаков Владислав Сергеевич

Научный руководитель: Ананичев Дмитрий Сергеевич

# Преобразование изображения в JPEG

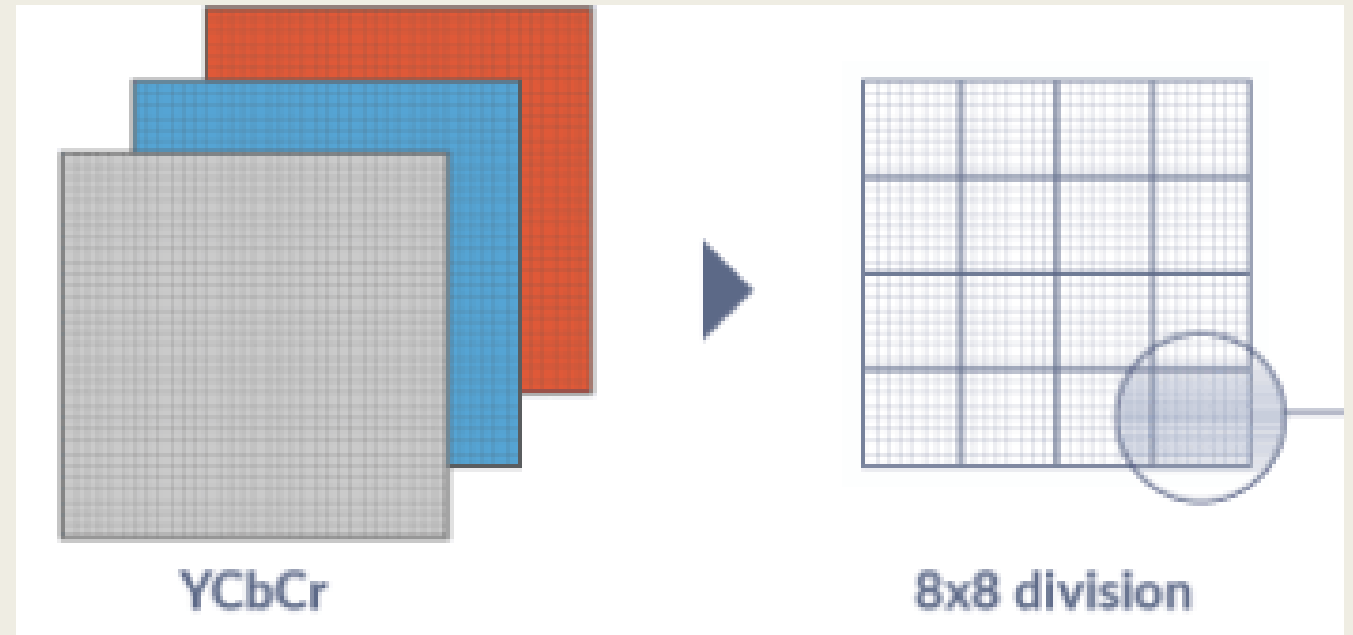
- Шаг 1. Преобразование цветового пространства



$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

# Преобразование изображения в JPEG

- **Шаг 1.** Преобразование цветового пространства
- **Шаг 2.** MCU, разбивка на матрицы 8x8



# Преобразование изображения в JPEG

- **Шаг 1.** Преобразование цветового пространства
- **Шаг 2.** MCU, разбивка на матрицы 8x8
- **Шаг 3.** Дискретное косинусное преобразование

-414	-33	-59	36	57	-51	-16	-12
6	-36	40	19	27	1	-5	3
-47	13	81	-35	-50	20	8	-18
-50	21	35	-20	2	34	39	12
8	-3	8	-5	-30	-15	40	36
-7	17	-16	7	-7	16	4	6
20	-29	-1	-27	-1	8	-43	-21
18	26	-11	-45	31	47	-36	4

$$F(u, v) = \frac{c(u, v)}{4} * \sum_{x=0}^7 \sum_{y=0}^7 \left( f(x, y) \times \cos\left(\frac{2 * x + 1}{16} u \pi\right) \times \cos\left(\frac{2 * y + 1}{16} v \pi\right) \right)$$

$$c(u, v) = \begin{cases} \left(\frac{1}{2}\right), u = 0 \text{ и } v = 0 \\ 1, u \neq 0 \text{ или } v \neq 0 \end{cases}$$

# Преобразование изображения в JPEG

- **Шаг 1.** Преобразование цветового пространства
- **Шаг 2.** MCU, разбивка на матрицы 8x8
- **Шаг 3.** Дискретное косинусное преобразование
- **Шаг 4.** Квантование

-414	-33	-59	36	57	-51	-16	-12
6	-36	40	19	27	1	-5	3
-47	13	81	-35	-50	20	8	-18
-50	21	35	-20	2	34	39	12
8	-3	8	-5	-30	-15	40	36
-7	17	-16	7	-7	16	4	6
20	-29	-1	-27	-1	8	-43	-21
18	26	-11	-45	31	47	-36	4

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

# Преобразование изображения в JPEG

- **Шаг 1.** Преобразование цветового пространства
- **Шаг 2.** MCU, разбивка на матрицы 8x8
- **Шаг 3.** Дискретное косинусное преобразование
- **Шаг 4.** Квантование

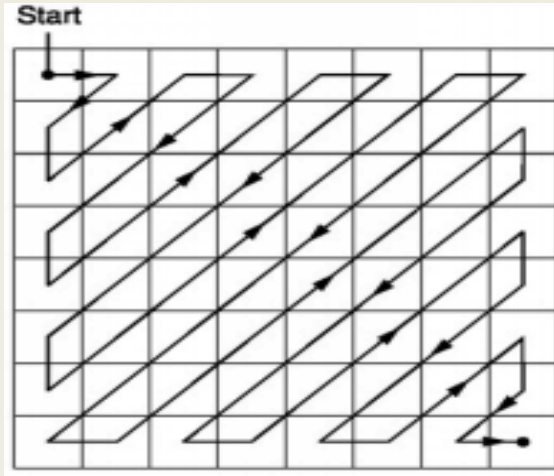
$$\begin{bmatrix} -414 & -33 & -59 & 36 & 57 & -51 & -16 & -12 \\ 6 & -36 & 40 & 19 & 27 & 1 & -5 & 3 \\ -47 & 13 & 81 & -35 & -50 & 20 & 8 & -18 \\ -50 & 21 & 35 & -20 & 2 & 34 & 39 & 12 \\ 8 & -3 & 8 & -5 & -30 & -15 & 40 & 36 \\ -7 & 17 & -16 & 7 & -7 & 16 & 4 & 6 \\ 20 & -29 & -1 & -27 & -1 & 8 & -43 & -21 \\ 18 & 26 & -11 & -45 & 31 & 47 & -36 & 4 \end{bmatrix} \div \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$\begin{bmatrix} -25 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 1 & -3 & 3 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Сжатие квантованных матриц

- Шаг 1. ZigZag обход

-25	<b>-3</b>	-6	<b>2</b>	2	<b>-1</b>	0	0
<b>1</b>	-3	<b>3</b>	1	<b>1</b>	0	0	0
-3	<b>1</b>	5	<b>-1</b>	-1	0	0	0
<b>-4</b>	1	<b>2</b>	-1	0	0	0	0
0	<b>0</b>	0	0	0	0	0	0
<b>0</b>	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0


$$[-25, -3, 1, -3, -3, -6, 2, 3, 1, -4, 0, 1, 5, 1, 2, -1, 1, -1, 2, 0, 0, 0, 0, 0, -1, -1, 0, 0, \dots]$$

# Сжатие квантованных матриц

- Шаг 1. ZigZag обход
- Шаг 2. Обрезка и свёртка нулей

$[-25, -3, 1, -3, -3, -6, 2, 3, 1, -4, \mathbf{0, 1}, 5, 1, 2, -1, 1, -1, 2, \mathbf{0, 0, 0, 0, 0, -1}, -1, \mathbf{0, 0}, \dots]$

$[(0, -25), -3, 1, -3, -3, -6, 2, 3, 1, -4, (1, 1), 5, 1, 2, -1, 1, -1, 2, (5, -1), -1, \mathbf{END} = (0, 0)]$



# Сжатие квантованных матриц

- Шаг 1. Зигзаг обход
- Шаг 2. Обрезка и свёртка нулей
- Шаг 3. Категории

Значение	Категория	Код
0	0	–
-1, 1	1	0, 1
-3, -2, 2, 3	2	00, 01, 10, 11
-7, -6, -5, -4, 4, 5, 6, 7	3	000, 001, 010, 011, 100, 101, 110, 111
⋮	⋮	⋮
-32767 ... -16384, 16384 ... 32767	15	...

# Сжатие квантованных матриц

- Шаг 1. ZigZag обход
- Шаг 2. Обрезка и свёртка нулей
- Шаг 3. Категории
- Шаг 4. Кодирование Хаффмана

(Нули, Категория)	Частота использования	Код Хаффмана
(0, 1)	2527816	00
(0, 2)	1865317	01
(0, 3)	1033833	100
(1, 1)	648828	101
(0, 4)	363132	1100
(1, 2)	253353	11011
(2, 1)	256078	11100
(0, 0)	161914	11010
⋮	⋮	⋮

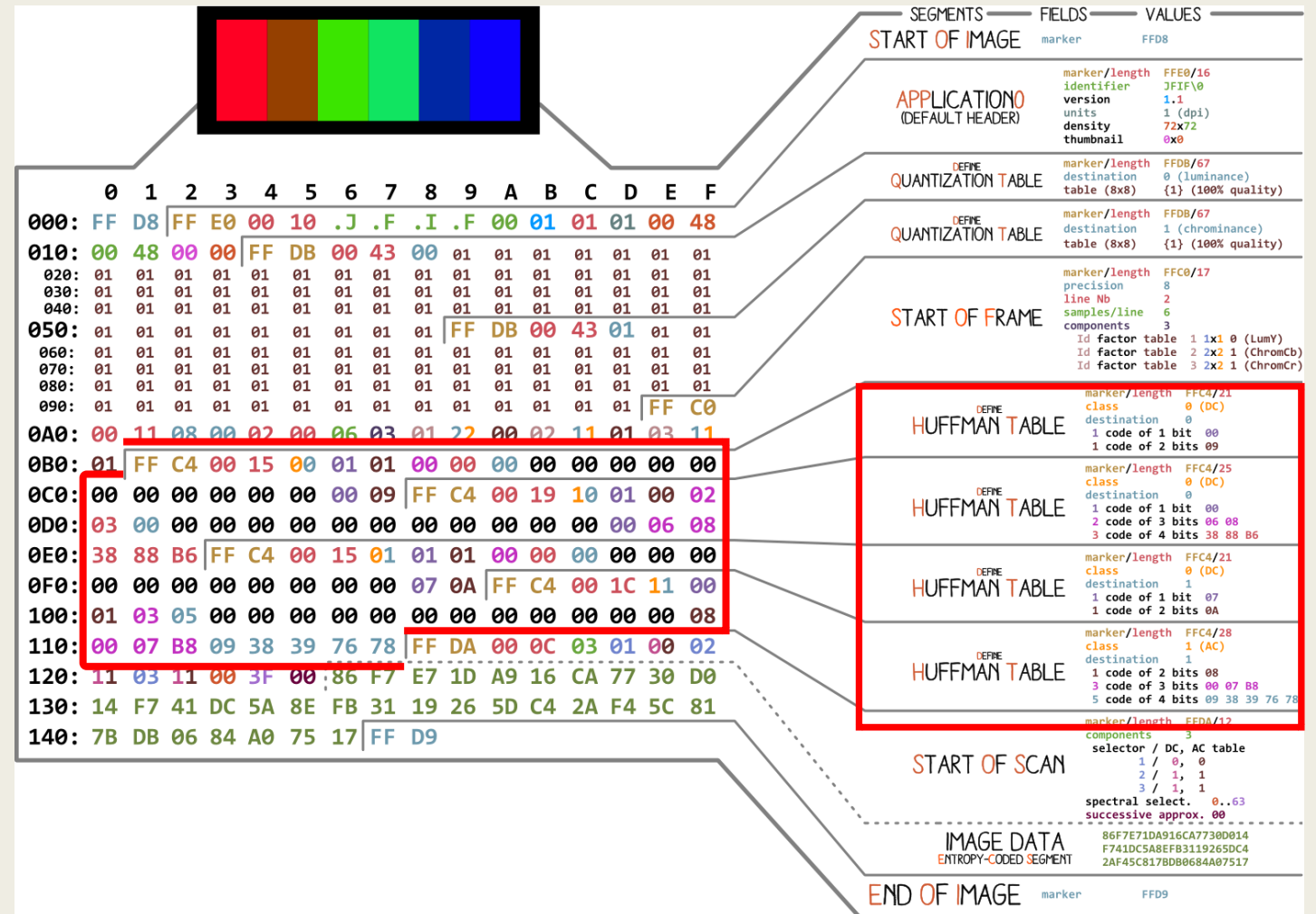
# Сжатие квантованных матриц

- Шаг 1. ZigZag обход
- Шаг 2. Обрезка и свёртка нулей
- Шаг 3. Категории
- Шаг 4. Кодирование Хаффмана

(Нули, Категория)	Частота использования	Код Хаффмана
(0, 1)	2527816	00
(0, 2)	1865317	01
(0, 3)	1033833	100
(1, 1)	648828	101
(0, 4)	363132	1100
(1, 2)	253353	11011
(2, 1)	256078	11100
(0, 0)	161914	11010
⋮	⋮	⋮

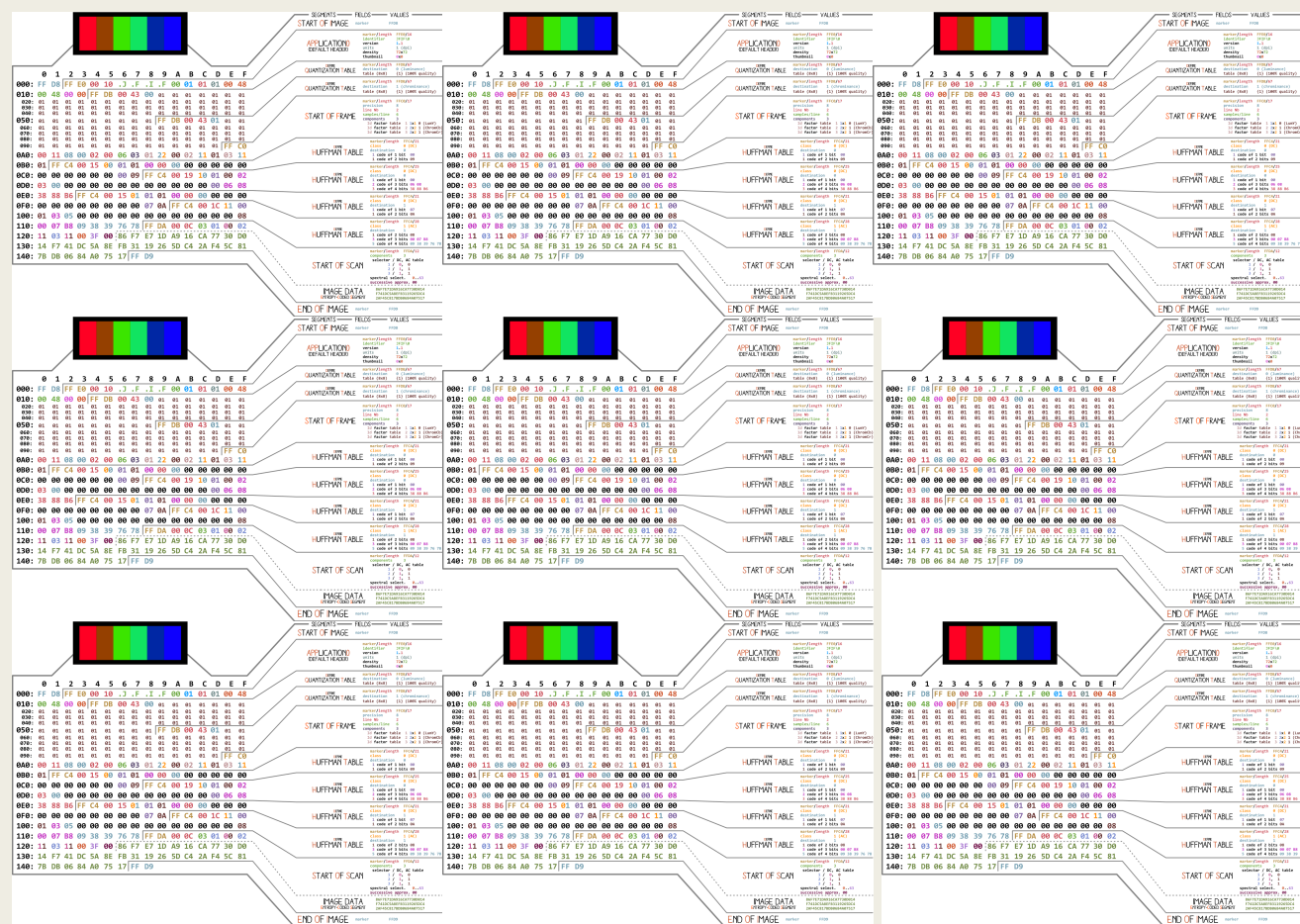
# Сжатие квантованных матриц

- Цель: Сжать в архив JPEG файлы



# Сжатие квантованных матриц

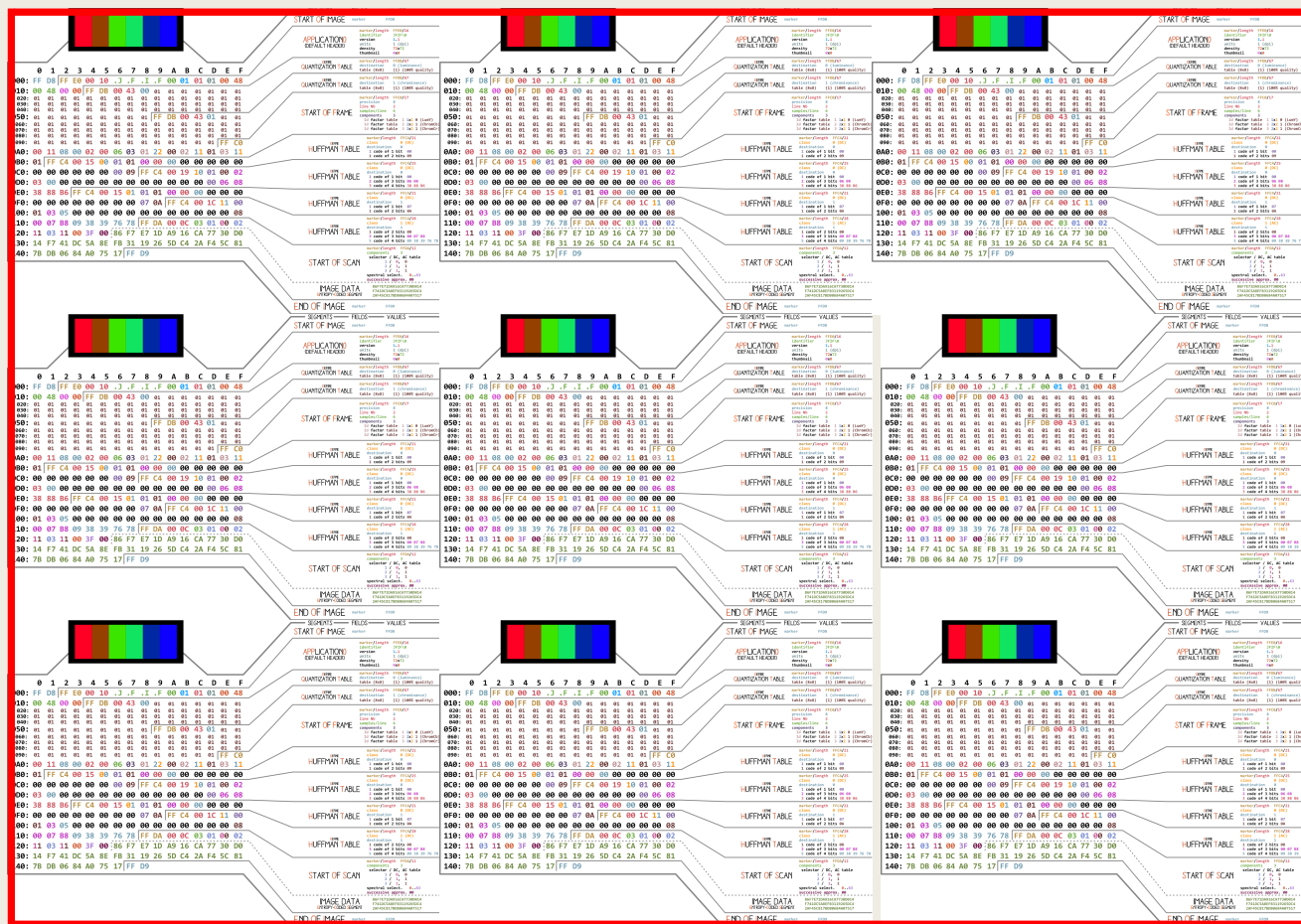
- **Цель:** Сжать в архив JPEG файлы



# Сжатие квантованных матриц

- **Цель:** Сжать в архив JPEG файлы

<pre> DEFINE HUFFMAN TABLE </pre>	<pre> marker/length  FFC4/25 class          0 (DC) destination    1 1 code of 1 bit  00 1 code of 2 bits 09 </pre>
<pre> DEFINE HUFFMAN TABLE </pre>	<pre> marker/length  FFC4/25 class          0 (DC) destination    1 1 code of 1 bit  00 2 code of 3 bits 06 08 3 code of 4 bits 38 88 86 </pre>
<pre> DEFINE HUFFMAN TABLE </pre>	<pre> marker/length  FFC4/21 class          0 (DC) destination    1 1 code of 1 bit  07 1 code of 2 bits 0A </pre>
<pre> DEFINE HUFFMAN TABLE </pre>	<pre> marker/length  FFC4/28 class          1 (AC) destination    1 1 code of 2 bits 08 3 code of 3 bits 00 07 88 5 code of 4 bits 09 38 39 76 78 </pre>



# Сжатие квантованных матриц

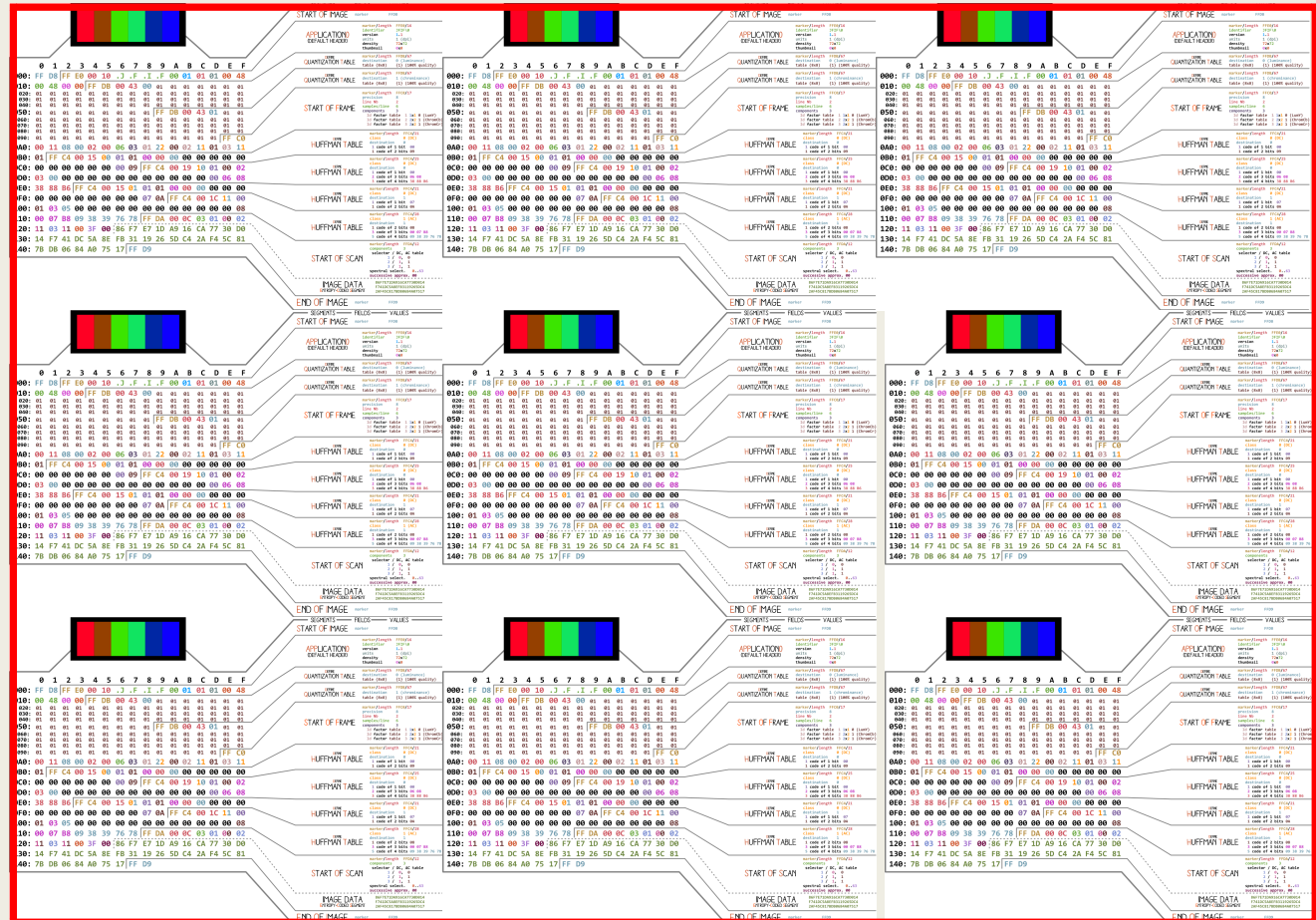
- Цель: Сжать в архив JPEG файлы

```
DEFINE HUFFMAN TABLE
marker/length FFC4/21
class 0 (DC)
destination 0
1 code of 1 bit 00
1 code of 2 bits 09

DEFINE HUFFMAN TABLE
marker/length FFC4/25
class 0 (DC)
destination 0
1 code of 1 bit 00
2 code of 3 bits 06 08
3 code of 4 bits 38 88 B6

DEFINE HUFFMAN TABLE
marker/length FFC4/21
class 0 (DC)
destination 1
1 code of 1 bit 07
1 code of 2 bits 0A

DEFINE HUFFMAN TABLE
marker/length FFC4/28
class 1 (AC)
destination 1
1 code of 2 bits 08
3 code of 3 bits 00 07 B8
5 code of 4 bits 09 38 39 76 78
```



# Алгоритм

- **Шаг 1.** Декодируем матрицы.



4.jpg



5.jpg



6.jpg



7.jpg



# Алгоритм

- Шаг 1. Декодируем матрицы.
- Шаг 2. Собираем статистику частот

```
Y·DC
{0: 17049, 1: 32848, 2: 58746, 3: 86073, 4: 101565, 5: 101208, 6: 83807, 7: 59047, 8: 22584, 9: 1704, 10: 202, 11: 47}
Y·AC
{0: 319094, 1: 7907755, 2: 3351441, 3: 1246854, 4: 719877, 5: 453947, 6: 227426, 7: 64284, 8: 8007, 9: 215, 17: 3069482, 18: 766981, 19: 113900, 20: 29962, 21: 10553, 22: 3288, 23: 627, 24: 53, 25: 1, 33: 1374065, 34: 229008, 35: 15556, 36: 1879, 37: 338, 38: 69, 39: 7, 40: 1, 49: 665077, 50: 80769, 51: 2784, 52: 190, 53: 31, 54: 6, 65: 339275, 66: 32441, 67: 585, 68: 24, 69: 3, 81: 178093, 82: 13664, 83: 119, 84: 3, 97: 95212, 98: 5925, 99: 36, 113: 52237, 114: 2725, 115: 11, 129: 29587, 130: 1268, 131: 1, 145: 16780, 146: 594, 147: 1, 161: 9725, 162: 314, 177: 5593, 178: 137, 193: 3386, 194: 77, 209: 2028, 210: 40, 225: 1245, 226: 26, 240: 1384, 241: 759, 242: 11}
Cb·DC
{0: 53044, 1: 28843, 2: 41789, 3: 55439, 4: 46442, 5: 35193, 6: 16777, 7: 4398, 8: 491, 9: 24}
Cb·AC
{0: 265283, 1: 1005665, 2: 369690, 3: 170325, 4: 73041, 5: 19159, 6: 2760, 7: 186, 17: 440680, 18: 68017, 19: 16156, 20: 2865, 21: 359, 22: 31, 23: 1, 33: 244801, 34: 15806, 35: 1738, 36: 151, 37: 12, 49: 170450, 50: 6560, 51: 351, 52: 21, 65: 119802, 66: 2747, 67: 52, 68: 1, 81: 96177, 82: 1583, 83: 8, 97: 73748, 98: 918, 99: 6, 113: 55957, 114: 402, 115: 1, 129: 48053, 130: 265, 145: 39474, 146: 250, 147: 1, 161: 35178, 162: 190, 163: 1, 177: 26899, 178: 111, 193: 24635, 194: 87, 209: 21295, 210: 89, 225: 15097, 226: 58, 240: 56458, 241: 10646, 242: 12}
Cr·DC
{0: 63328, 1: 27345, 2: 40475, 3: 53018, 4: 43303, 5: 31353, 6: 17093, 7: 6052, 8: 471, 9: 2}
Cr·AC
{0: 267738, 1: 881349, 2: 302113, 3: 128970, 4: 53266, 5: 14290, 6: 1698, 17: 396842, 18: 56861, 19: 12717, 20: 2161, 21: 281, 22: 6, 33: 225609, 34: 12980, 35: 1307, 36: 81, 49: 160805, 50: 5775, 51: 277, 52: 10, 65: 114899, 66: 2370, 67: 42, 81: 94263, 82: 1423, 83: 26, 97: 73105, 98: 881, 99: 5, 113: 56571, 114: 377, 115: 1, 129: 49167, 130: 266, 145: 40509, 146: 271, 147: 1, 161: 36609, 162: 192, 163: 1, 177: 28179, 178: 100, 193: 25336, 194: 76, 209: 22280, 210: 110, 225: 15622, 226: 64, 240: 58982, 241: 11353, 242: 23}
```

# Алгоритм

- **Шаг 1.** Декодируем матрицы.
- **Шаг 2.** Собираем статистику частот
- **Шаг 3.** Генерируем коды Хаффмана

```
Expanded Form of Codes:
Codes of length 02 bits:
  00 = 01 (Total Len = 3)
  01 = 02 (Total Len = 4)
Codes of length 03 bits:
  100 = 03 (Total Len = 6)
  101 = 11 (Total Len = 4)
Codes of length 04 bits:
  1100 = 04 (Total Len = 8)
Codes of length 05 bits:
  11010 = 00 (EOB) (Total Len = 5)
  11011 = 12 (Total Len = 7)
  11100 = 21 (Total Len = 6)
Codes of length 06 bits:
  111010 = 05 (Total Len = 11)
  111011 = 13 (Total Len = 9)
  111100 = 31 (Total Len = 7)
Codes of length 07 bits:
  1111010 = 06 (Total Len = 13)
  1111011 = 22 (Total Len = 9)
  1111100 = 41 (Total Len = 8)
Codes of length 08 bits:
  11111010 = 51 (Total Len = 9)
  11111011 = 61 (Total Len = 9)
Codes of length 09 bits:
  111111000 = 07 (Total Len = 16)
  111111001 = 14 (Total Len = 13)
  111111010 = 23 (Total Len = 12)
  111111011 = 32 (Total Len = 11)
```

# Алгоритм

- **Шаг 1.** Декодируем матрицы.
- **Шаг 2.** Собираем статистику частот
- **Шаг 3.** Генерируем коды Хаффмана
- **Шаг 4.** Кодировем матрицы



Union.jpg

# Сравнение алгоритма

Тест (описание)	Размер файлов (Б-байт)	Алгоритм <i>compress.py</i>	WinRAR rar (32 Б) обычный	WinRAR rar (1024 Б) максимальный	WinRAR zip
0 (один файл)	5 321 980	5 315 973 (0.9989)	5 322 131	5 322 131	5 316 599
1 (много разных файлов)	362 227 195	352 042 512 (0.9719)	357 456 809	357 452 689	358 807 393
2 (много больших фото)	3 339 347 585	3 316 346 218 (0.9931)	3 334 708038	3 334 705 146	3 329 845467
3 (серия из 4 фото)	13 395 043	13 072 305 (0.9759)	13 218 699	13 218 638	13 218 668

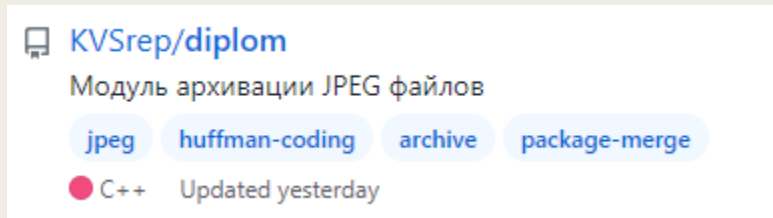
# Серийная съёмка

Исходный размер (Б)	Сжатие алгоритмом	Сжатие WinRAR .rar	Сжатие .zip	Сжатие .7z	Сжатие алгоритмом + .7z
13 395 043	13 072 305 (0.976)	13 218 638 (0.987)	13 218 668 (0.987)	13 116 576 (0.979)	12 900 644 (0.963)



# Исходники

- Исходный код реализованного алгоритма выложен на GitHub.



<https://github.com/KVSrep/diplom>