

МЭС-2016

Презентация доклада

*Повышение эффективности тензорной аппроксимации
для сжатия изображений
с помощью заранее обученного словаря*

проф. Чобану М.К., асп. Макаров Д.В.

НИУ “МЭИ”

г. Москва, 2016 г.

Тензор

Обозначения

$T(i_1, i_2, i_3, \dots, i_d)$ - тензор размера $n_1 \times n_2 \times n_3 \times \dots \times n_d$

d – размерность тензора,

i_k – индексы, могут принимать n_k целочисленных значений

Скаляр
(тензор размерности 0)

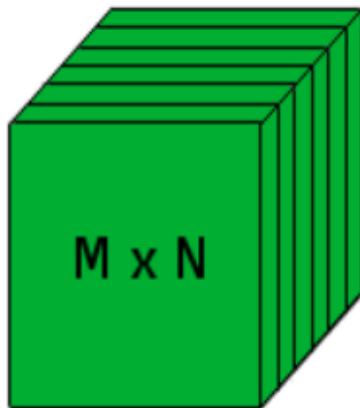
1

Тензор

Вектор – $T(i_1)$, $i_1 = 1 \dots n_1$

1 | | | | n_1

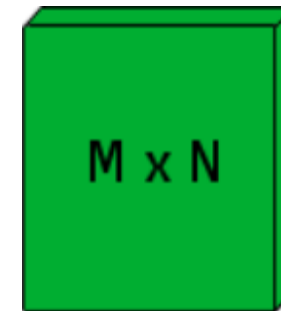
Видео



$T(i_1, i_2, i_3)$ размера $M \times N \times n_3$

Изображение

(в оттенках серого)



$T(i_1, i_2)$ размера $M \times N$

Тензорная аппроксимация

Diagram illustrating the decomposition of a matrix \mathbf{X} (size $I \times J$) into rank-1 components and then into matrices \mathbf{A} , \mathbf{D} , and \mathbf{B}^T .

Matrix \mathbf{X} (size $I \times J$) is approximated by the sum of rank-1 components:

$$\mathbf{X} \cong \lambda_1 \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix} + \dots + \lambda_R \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix}$$

The first component is shown with vectors \mathbf{a}_1 (vertical) and \mathbf{b}_1 (horizontal). The R -th component is shown with vectors \mathbf{a}_R (vertical) and \mathbf{b}_R (horizontal).

This decomposition is equivalent to the matrix product:

$$\mathbf{X} \cong \mathbf{A} \mathbf{D} \mathbf{B}^T$$

where:

- \mathbf{A} (size $I \times R$) is a matrix with columns \mathbf{a}_r .
- \mathbf{D} (size $R \times R$) is a diagonal matrix.
- \mathbf{B}^T (size $R \times J$) is a matrix with rows \mathbf{b}_r .

Diagram illustrating the decomposition of a 3D tensor \mathcal{X} (size $I \times J \times K$) into rank-1 components and then into matrices \mathbf{A} , \mathcal{D} , and \mathbf{B}^T .

Tensor \mathcal{X} (size $I \times J \times K$) is approximated by the sum of rank-1 components:

$$\mathcal{X} \cong \lambda_1 \begin{matrix} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix} + \dots + \lambda_R \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix}$$

The first component is shown with vectors \mathbf{a}_1 (vertical), \mathbf{b}_1 (horizontal), and \mathbf{c}_1 (depth). The R -th component is shown with vectors \mathbf{a}_R (vertical), \mathbf{b}_R (horizontal), and \mathbf{c}_R (depth).

This decomposition is equivalent to the tensor product:

$$\mathcal{X} \cong \mathbf{A} \mathcal{D} \mathbf{B}^T$$

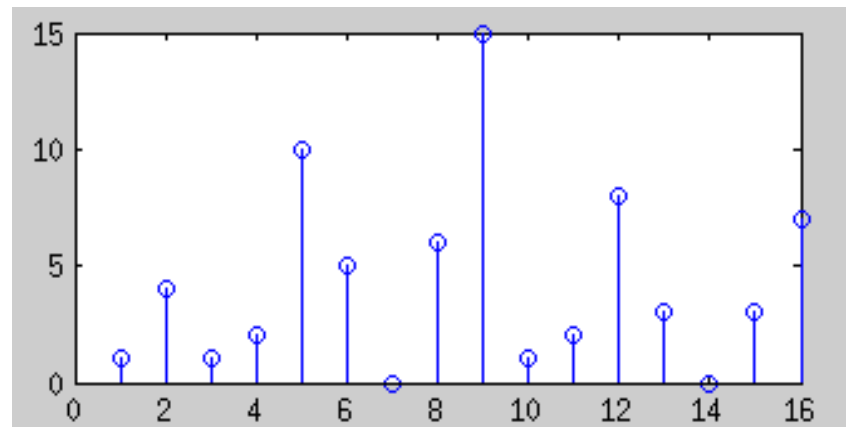
where:

- \mathbf{A} (size $I \times R$) is a matrix with columns \mathbf{a}_r .
- \mathcal{D} (size $R \times R \times R$) is a diagonal tensor.
- \mathbf{B}^T (size $R \times J$) is a matrix with rows \mathbf{b}_r .
- \mathbf{C} (size $K \times R$) is a matrix with rows \mathbf{c}_r .

Разреженное представление

Исходный сигнал (изображение)

1	10	15	3
4	5	1	0
1	0	2	3
2	6	8	7

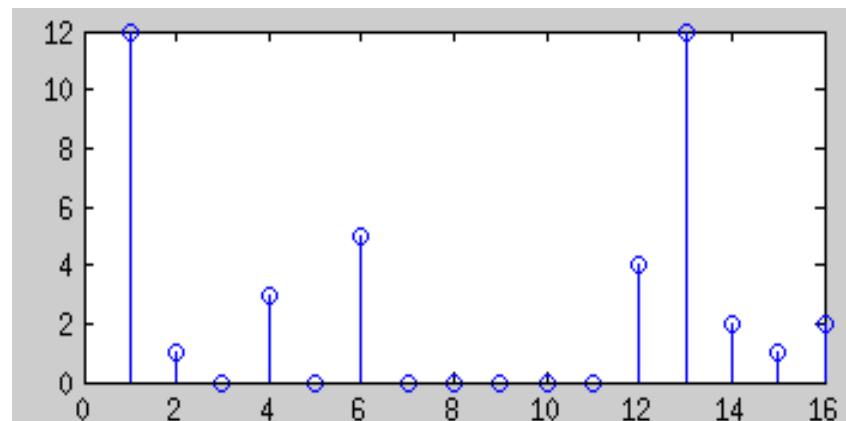


Энтропия: 3.3278



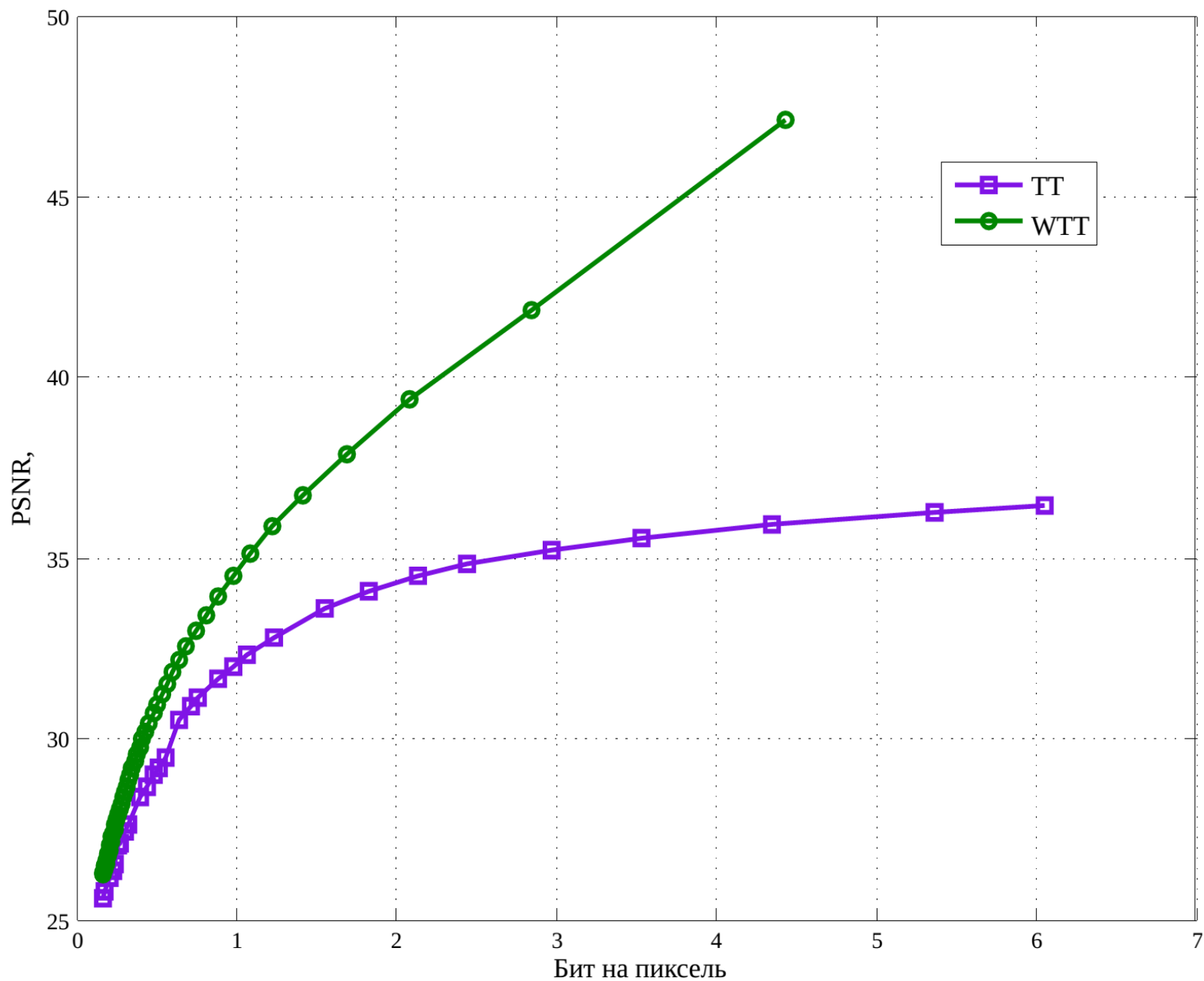
Проекция на более подходящий базис

12	1	0	3
0	5	0	0
0	0	0	4
12	2	1	2



Энтропия: 2.3968

Сравнение эффективности тензорной аппроксимации и разреженного представления

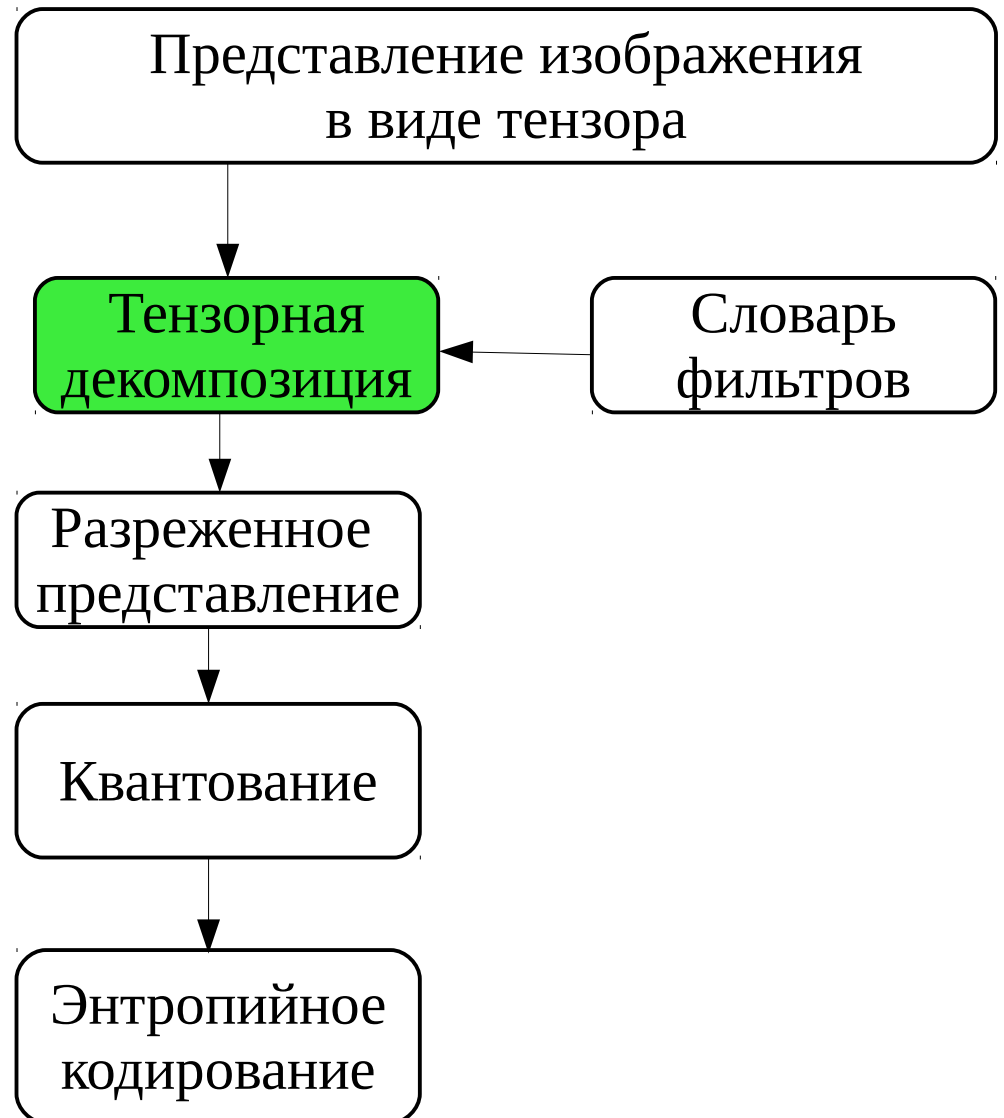


Схемы сжатия изображений

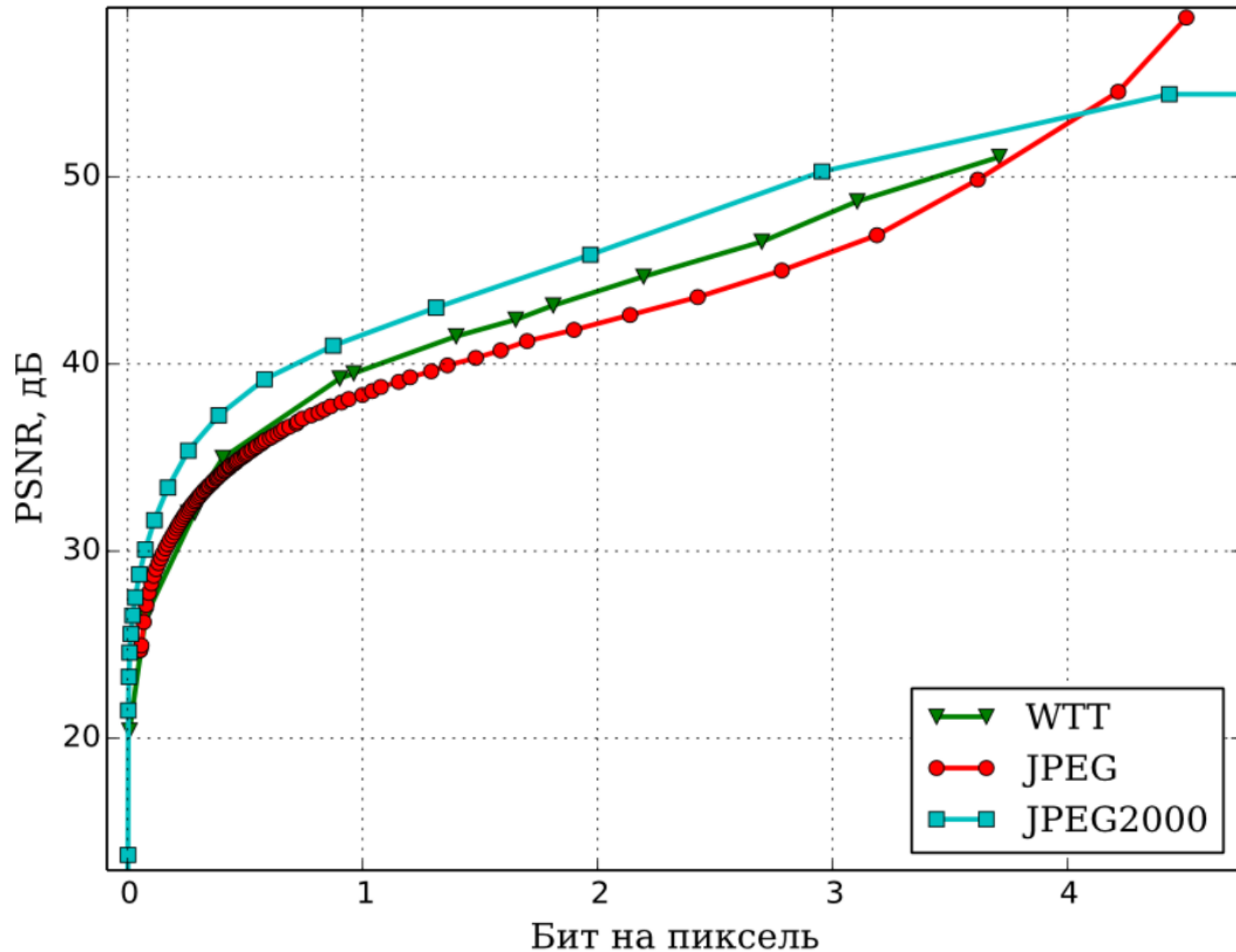
С использованием
тензорной декомпозиции



С использованием
разреженного представления

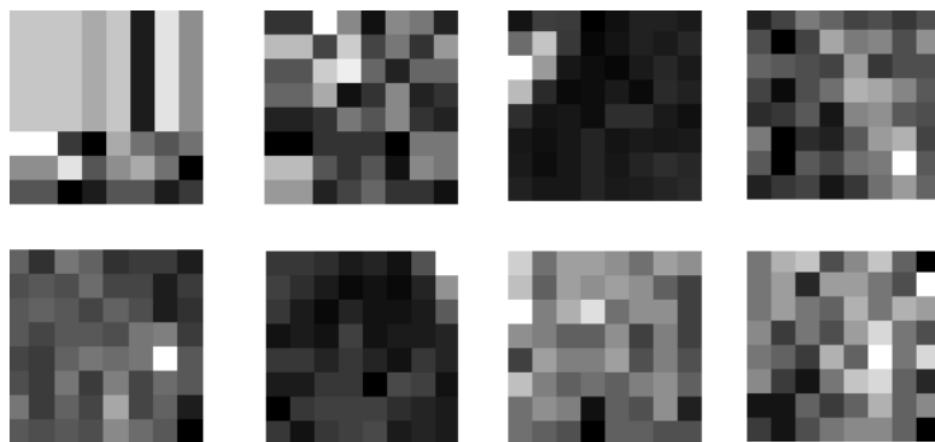


Метод разреженного представления с универсальным словарем



Сравнение с JPEG и JPEG2000

Эффективность фильтра словаря





Блоки, на коротких
эффективно работает
один из фильтров WTT

В таблице представлена статистика применения нескольких
фильтров на наборе 8192 блоков

Номер фильтра	Кол-во блоков, на которых фильтр работает как минимум не хуже ДКП	Кол-во блоков, на которых фильтр работает как минимум на 30% лучше ДКП	Кол-во блоков, на которых фильтр работает хуже ДКП
3823	2967	691	4534
4006	3147	635	4510
3264	3061	604	4527
3947	2694	602	4896
3821	2770	596	4826

Методы выбора фильтров из словаря

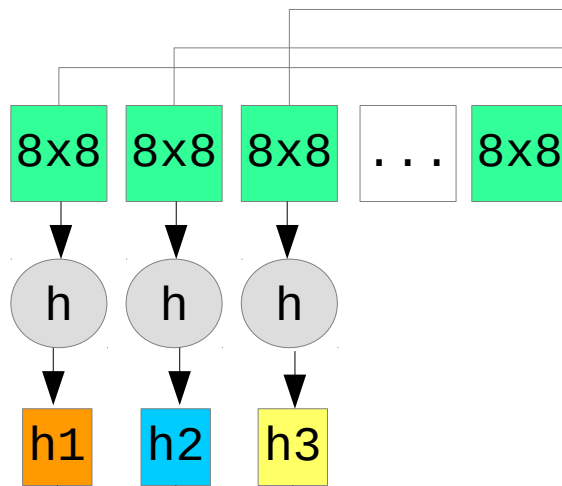
1. Перебор элементов словаря

-  Безошибочен
-  Низкая скорость сжатия

Методы выбора фильтров из словаря

2. Выбор фильтра по хеш-значению блока

Блоки для обучения словаря:



Соответствующие значения хеш-функции:

Фильтр, эффективный для данных блоков

Словарь:



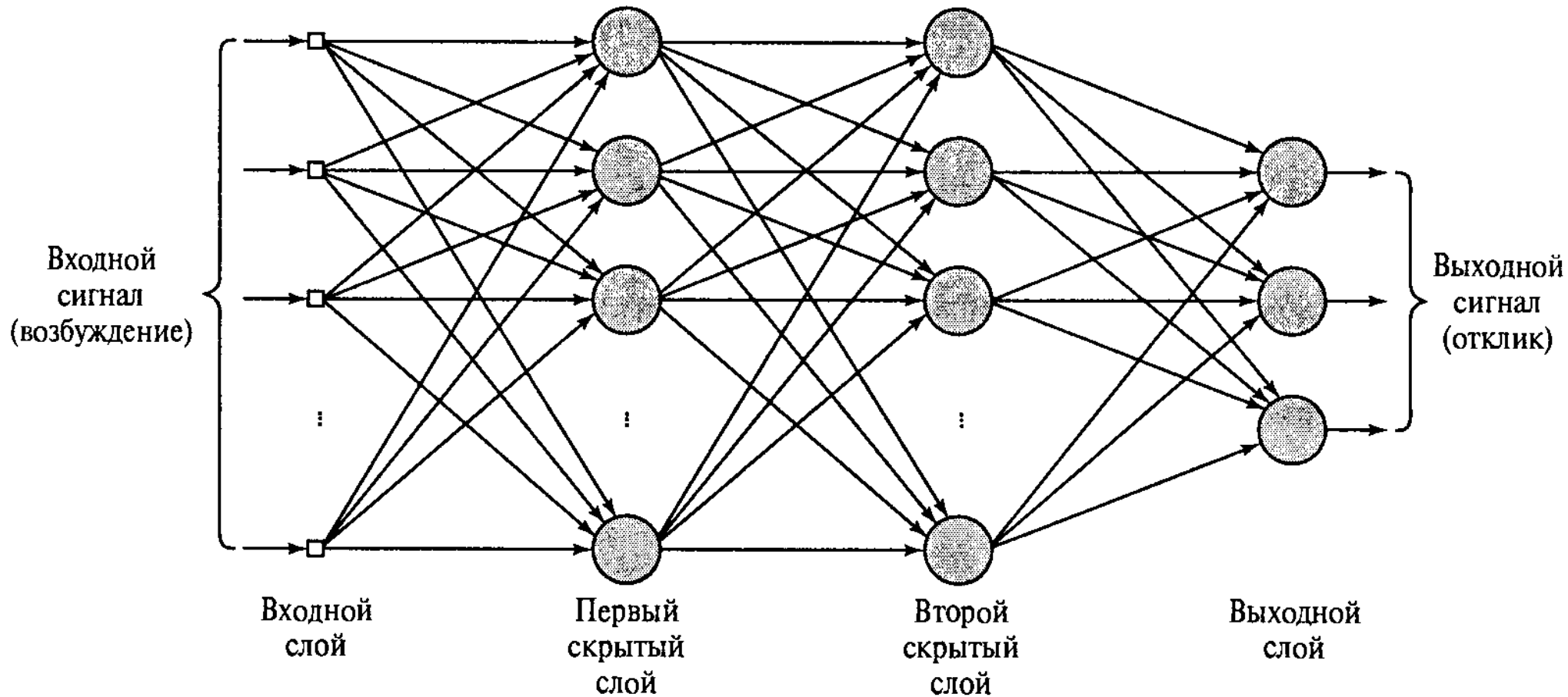
Методы выбора фильтров из словаря

2. Выбор фильтра по хеш-значению блока

- + Очень быстро выбирается фильтр
- Возможны коллизии хеш-функции
- Увеличивается объем словаря
- Может найти фильтр только для блока хеш-значение которого было добавлено в словарь или для похожего блока

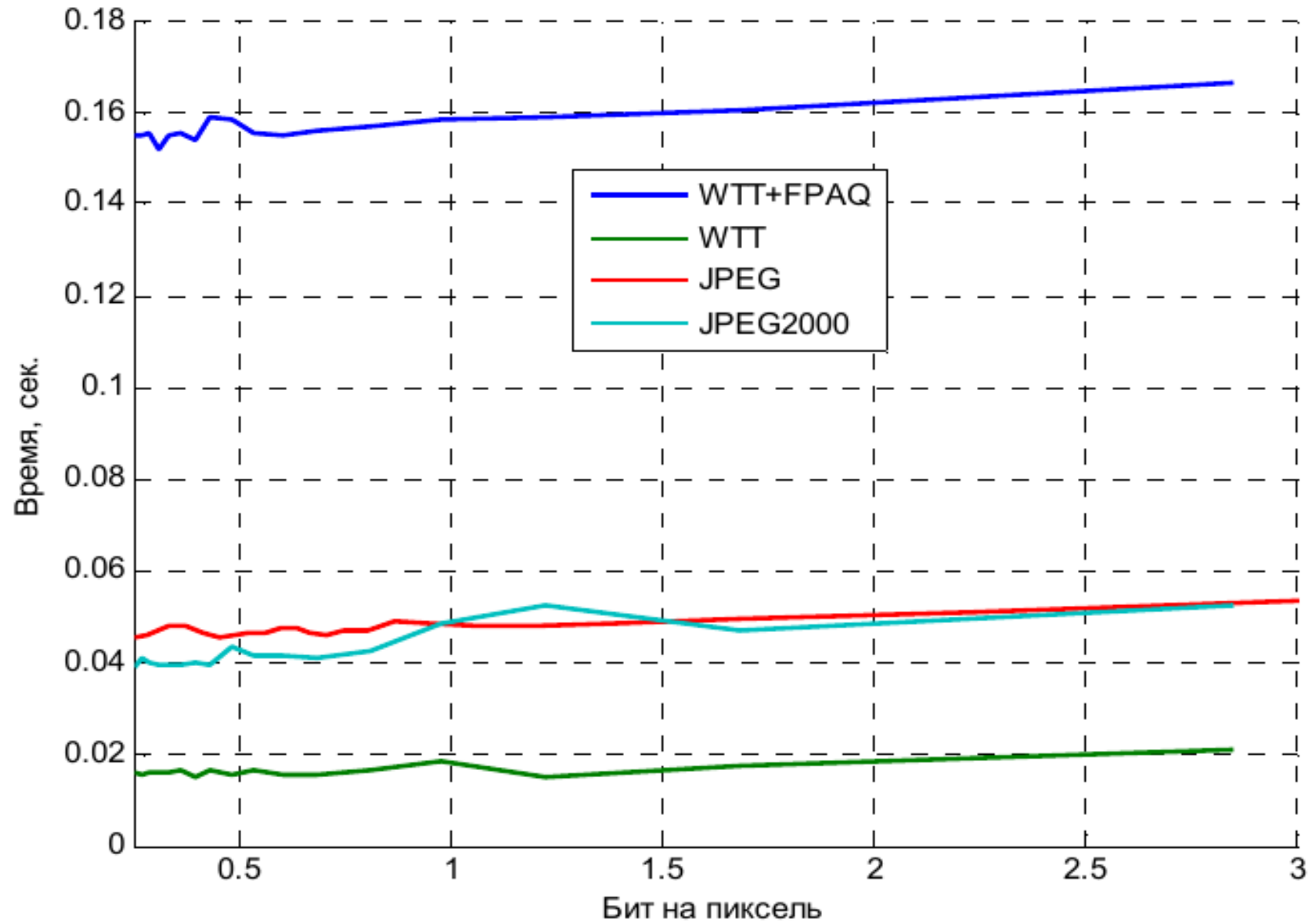
Дальнейшая доработка

3. Методы машинного обучения



Спасибо за внимание

Временные характеристики



Методы Тензорной Аппроксимации

Каноническая аппроксимация

Наиболее эффективно снижает объем данных

Нет стабильного алгоритма для вычисления

Аппроксимация Такера

Близка к канонической по снижению объема данных

Эффективна для тензоров размерности не более 3

Существует эффективный алгоритм для расчета

Аппроксимация Tensor-Train

Близка к канонической по снижению объема данных

Нет ограничения на размерность данных

Существует эффективный алгоритм для расчета

Можно использовать совместно с декомпозицией Такера

Метод Tensor-Train (ТТ)

Метод ТТ позволяет представить исходных тензор A в виде нескольких связанных между собой тензоров G размерности 3.

$$A(i_1, \dots, i_d) = \sum_{\alpha_0, \dots, \alpha_{d-1}, \alpha_d} G_1(\alpha_0, i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \dots G_d(\alpha_{d-1}, i_d, \alpha_d).$$

A – исходный тензор

G_k – тензор размера $r_{k-1} \times n_k \times r_k$

r – ранги аппроксимации (зависят от данных)

α – индексы для связи тензоров

Эффект достигается за счет выявления закономерностей среди входных данных

В результате такой аппроксимации количество элементов равно:

$$O((d-2) \cdot n \cdot r^2 + 2 \cdot n \cdot r)$$

Число элементов после канонической аппроксимации оценивается:

$$O(d \cdot n \cdot r)$$