Image Compression

High Performance Python Lab

Final Project

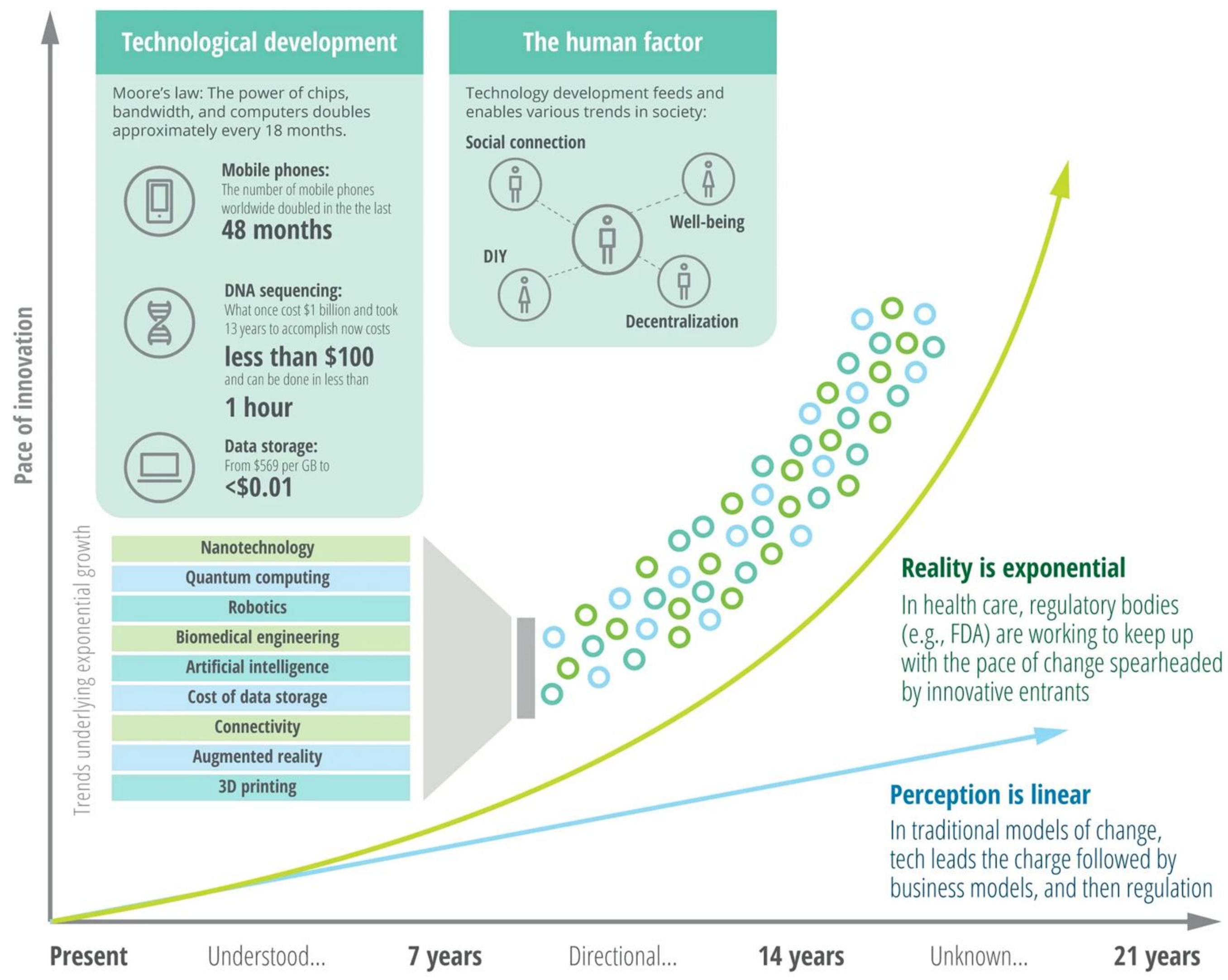
Olga Kupreeva, Iuliia Sadykova

2022

Skoltech

Modern Data problem

Exponential change will accelerate the pace of disruption



Technological development

Moore's law: The power of chips, bandwidth, and computers doubles approximately every 18 months.

Mobile phones: The number of mobile phones worldwide doubled in the the last **48 months**

DNA sequencing: What once cost $1 billion and took 13 years to accomplish now costs **less than $100** and can be done in less than **1 hour**

Data storage: From $569 per GB to **<$0.01**

The human factor

Technology development feeds and enables various trends in society:

Reality is exponential

In health care, regulatory bodies (e.g., FDA) are working to keep up with the pace of change spearheaded by innovative entrants

Perception is linear

In traditional models of change, tech leads the charge followed by business models, and then regulation

Note: All dollar amounts are given in US dollars.

Source: Deloitte analysis.

Introduction

- **What is it?**
  Image compression is a type of data compression applied to digital images, minimizing the number of bits to represent them

- **For what?**
  To *reduce space, cost and time* for storage or transmission

- **How does it work?**
  To provide superior results, algorithms may take advantage of visual perception and the statistical properties of image data

What makes compression possible?

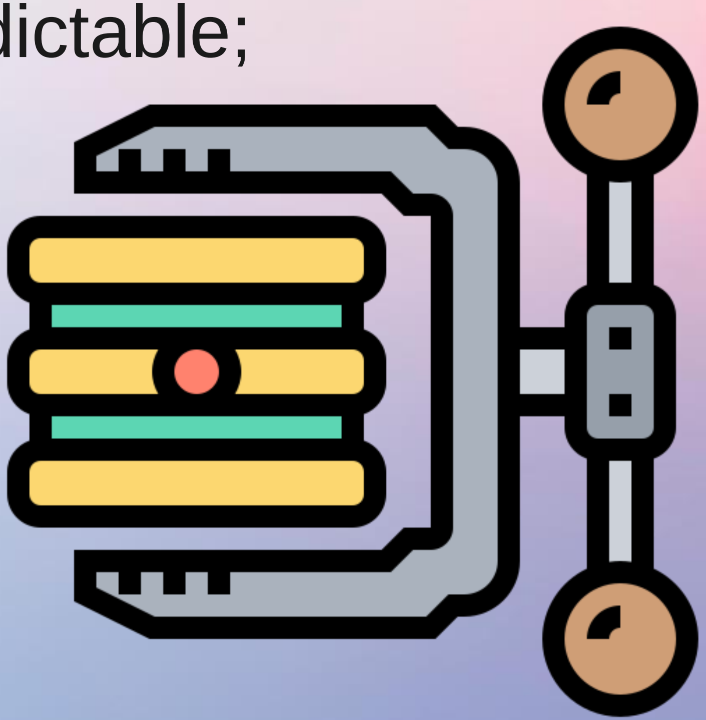
✔ Images are not noise, they are redundant and predictable;

✔ Intensities are distributed non-uniformly;

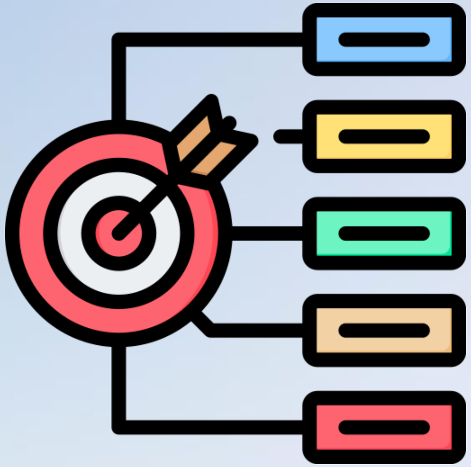
✔ Color channels are correlated;

✔ Pixel values are spatially correlated;

✔ Properties of human visual perception.



Aim and objectives



Exploring different approaches to image compression using Single-threaded and Multi-threaded Processes

- To apply Numpy library, mpi4py library module, Numba compiler and compare their parameters
- To gain three compressed images and visually evaluate them
-

Applications of lossy compression

- Data storing and/or transferring, graphical computing area
- Modern ways of people communication (messengers, social network);
- Applying machine learning, deep neural networks, virtual machines and computer vision technologies;
- Medical application (medical video, tomotherapy, ultrasound and X-ray results, phonocardiogram signals etc.)
- Materials mechanics, structure analyzing
- Satellite data

Types

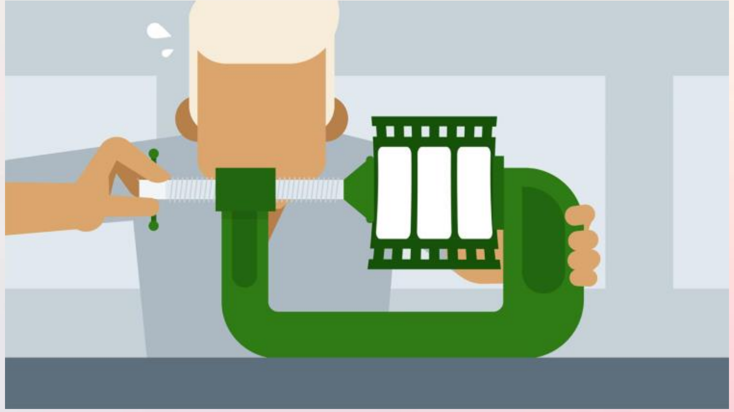


Image compression

Lossless compression

After, image can be converted back with zero error, but compression rate - low

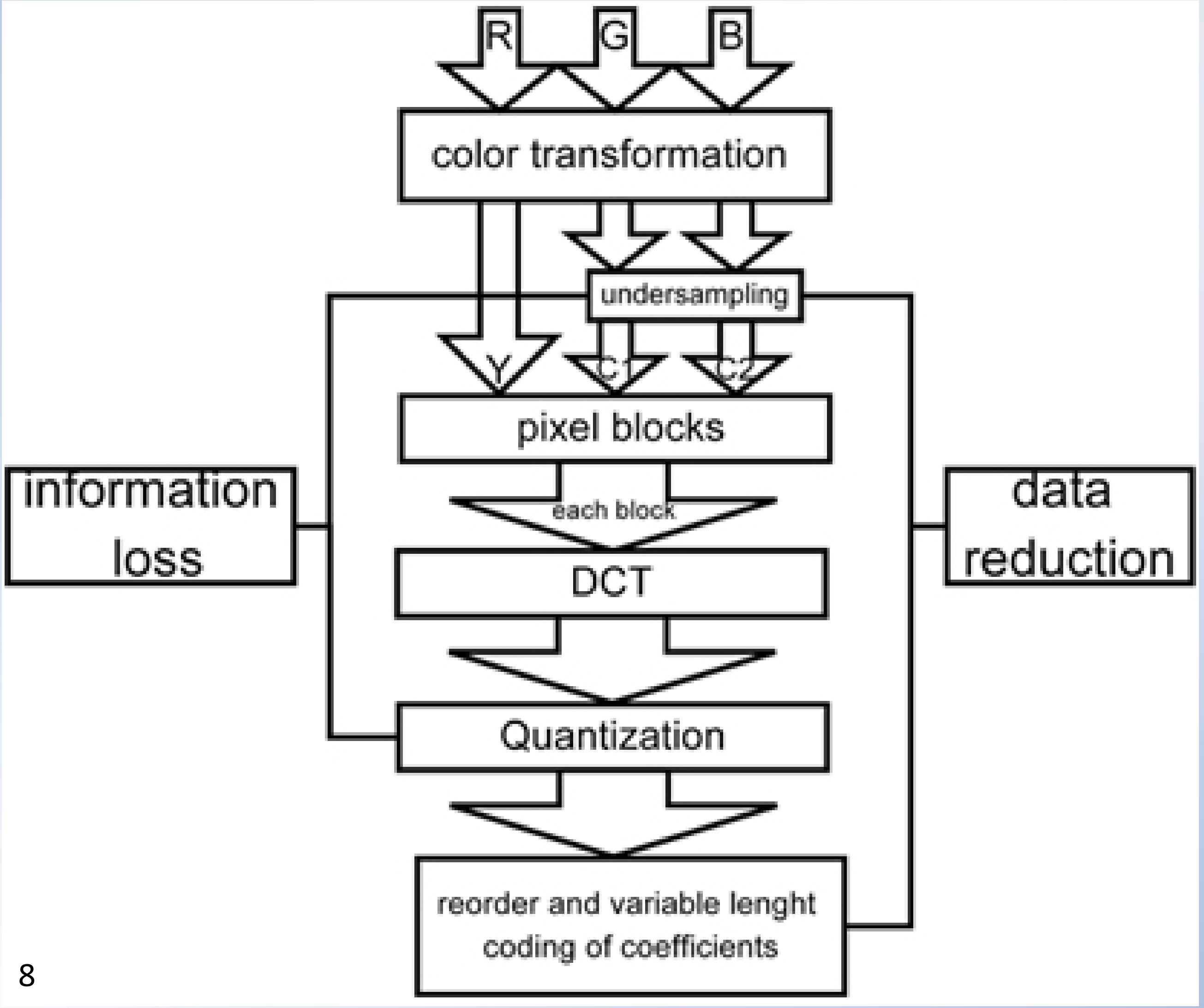
For artificially constructed images (graphics, program icons, or special cases, PNG, PCX)

Lossy compression

After, image cannot be converted back to the original without error;
The amount of error is inversely proportional to the storage space, can be controlled by the user

Algorithms, while compression ratio ↑, usually generate artifacts (JPEG, MPEG, MP3)

Discrete cosine transform

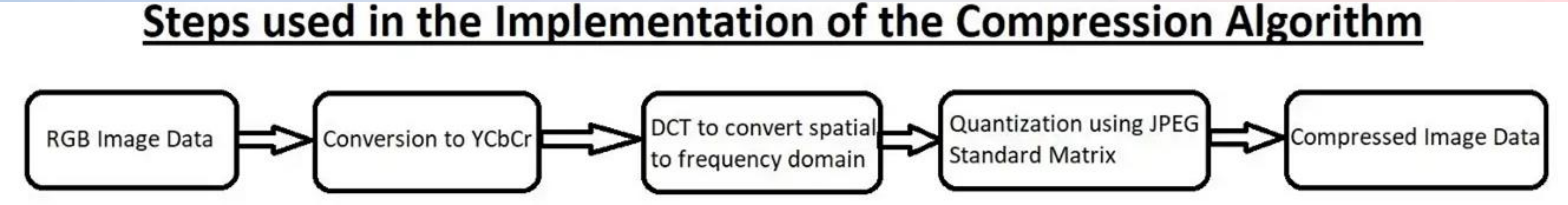


DCT - fast computing Fourier transform which maps real signal to corresponding values in frequency domain, expresses a signal as a linear combination of cosine bases; real-valued

Steps:

1. Convert RGB Image ->equivalent YCbCr format
2. Break image into N*N blocks
3. Apply DCT to every block serially
4. Apply quantization to restrict the number of values that can be saved without loss of information
5. Store subset of the quantized blocks into an array from where it can be picked up for further processing

Discrete cosine transform



Steps used in the Implementation of the Compression Algorithm
RGB Image Data
Conversion to YCbCr
DCT to convert spatial to frequency domain
Quantization using JPEG Standard Matrix
Compressed Image Data

1) Calculation of the DCT matrix:

$$T_{ij} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N} & \text{if } i > 0 \end{cases}$$

2) Compression process:

$$D = TMT^T$$

$$C_{ij} = round(\frac{D_{ij}}{Q_{ij}})$$

3) Decompression process:

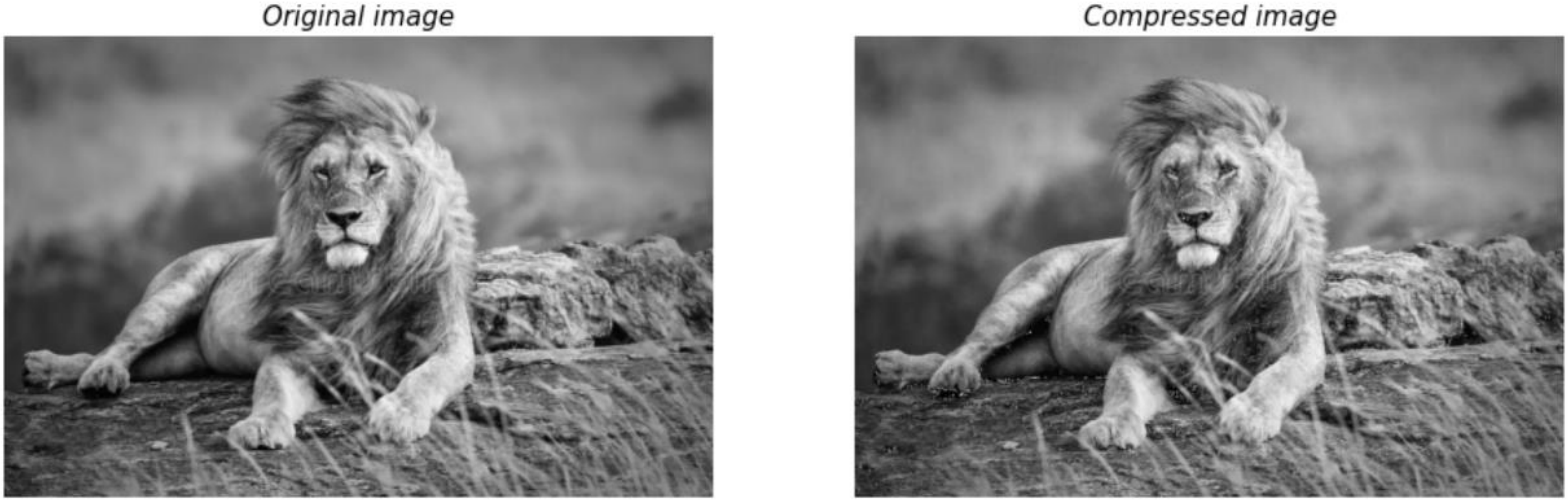
$$R_{ij} = Q_{ij} \times C_{ij}$$

$$N = round(T^T RT) + 128$$

Comparison of applied approaches

Parameter	Numpy	mpi4py	Numba
Time, s	0.6895	0.2957	0.0709

Results



Original image

Compressed image

Size Original: **150** KB Compressed: **143** KB

Δ = 7 KB

DCT formula

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N} \right] \cos\left[\frac{(2y+1)j\pi}{2N} \right]$$

$$C(u) = \left\{ \begin{array}{ll} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{array} \right\}$$