Implicit Image Compression

Encoding Pictures with Implicit Neural Representations

Lorenzo Catania[1], Dario Allegra[1]

[1]Department of Mathematics and Computer Science
University of Catania

IEEE International Conference on Image Processing, October 2024

Table of Contents

Table of Contents

Traditional data representations

Digital media signals are conventionally represented as discrete sequences of symbols. For instance:

- An **audio** is a sequence of amplitude samples.
- An **image** is a two-dimensional grid of pixels.
- A **video** is a sequence of images, to be visualized at a fixed rate.

These layouts are highly **inefficient**. Therefore, it is practical to **encode** data before storage/transmission.

Data encoding and decoding

The **encoding** of a **signal** x consists in obtaining a latent representation y by using an **encoder** function: $y = enc(x)$.
The latent representation y is, usually, not consumable as the original signal x would be, therefore a **decoder** function is used to obtain a reconstruction of the original signal: $\tilde{x} = dec(y)$.
A pair of and encoder and a decoder is referred to as **codec**.

Lossless and lossy compression

The most common motivation to encode information is to **compress** it, therefore we expect the latent representation y to be more spacially efficient than x.

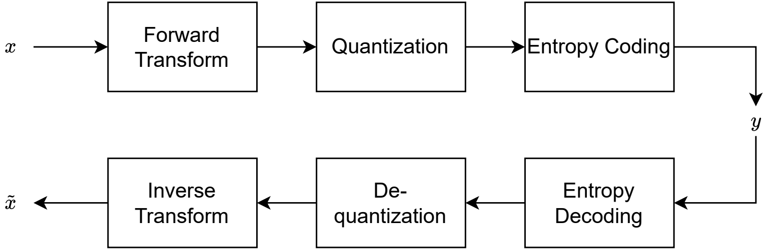
Compression techniques are distinguished in:

- **Lossless**: The original signal x can be retrieved from y.
  $\tilde{x} = dec(y) = x$.
- **Lossy**: Part of the information is lost in the encoding process, therefore the signal cannot be integrally reconstructed: $\tilde{x} \neq x$.

Lossy compression

Intuitively, the efficiency of lossless compression is limited by the need to preserve the original information. In the case of multimedia, it is acceptable to alterate the signal in order to achieve an higher compression ratio if the perceived distortion remains below a certain threshold. This process is the core of lossy data compression and is referred to as **rate-distortion optimization**.
Note that, while the rate is usually measured using standard values (e. g. size in bits), the distortion is quantified differently based on the media modality and the desired use.

Traditional Image Compression



Learned Image Compression

Due to the advances in Machine Learning, traditional compression pipelines were enhanced by replacing some steps with learned modules.

- **Pros**: Direct optimization on rate-distortion, less need for manual engineerization
- **Cons**: High computational demands, need for training on a large amount of data

Some efforts were recently put to provide a learning-based image compression standard, named JPEG AI.

Table of Contents

Functional Image Representation

Instead of thinking an image as a discrete grid of pixels, it is possible to interpret it as a function from coordinates to colours:

$$I(x, y) = (R, G, B)$$

This representations is convenient as it decouples the image's content from its original resolution. The image is rendered by evaluating the function at each desired coordinate.

Implicit Neural Representations

In practice, it is not feasible to express this function analytically. Therefore, a neural network is trained to overfit I and its parameters are the correspective latent representation of the original signal. This paradigm for data encoding is usually referred to as **Implicit Neural Representations (INRs)**.
When the purpose is to compress the signal, the network parameters are quantized and entropy coded.

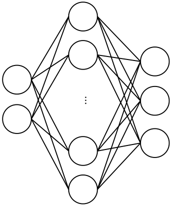
Training data - Images

Canvas Coordinates

(X, Y)
$(-1.0, -1.0)$
$(-0.9, -1.0)$
$(-1.0, -0.9)$
...
$(0.9, 1.0)$
$(1.0, 1.0)$

$\rightarrow$

Neural Network



$\rightarrow$

Pixel Color

(R, G, B)
$(0.2, 0.5, 0.3)$
$(0.2, 0.6, 0.1)$
$(0.3, 0.6, 0.2)$
...
$(1.0, 0.8, 0.2)$
$(1.0, 0.9, 0.2)$

Training data - Videos

Time-Space Coordinates

(T, X, Y)

$(-1.0, -1.0, -1.0)$

$(-1.0, -0.9, -1.0)$

$(-0.8, -1.0, -0.9)$

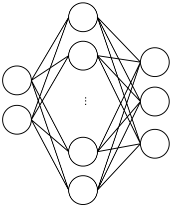
...

$(1.0, 0.9, 1.0)$

$(1.0, 1.0, 1.0)$

$\rightarrow$

Neural Network



$\rightarrow$

Pixel Color

(R, G, B)

$(0.2, 0.5, 0.3)$

$(0.2, 0.6, 0.1)$

$(0.3, 0.6, 0.2)$

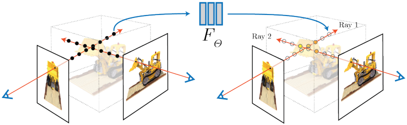
...

$(1.0, 0.8, 0.2)$

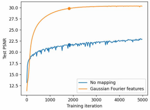
$(1.0, 0.9, 0.2)$

The overall training architecture remains **identical**.

↦ 2020: Neural Networks to approximate Radiance Fields (**NeRF**) [4]



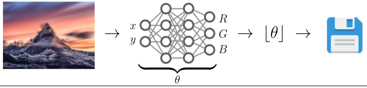
↦ 2020: Positional Encodings mitigate the spectral bias of small neural networks [8]



↦ 2020: Neural networks with periodic activation functions (**SIREN**) outperform ReLU networks [6]

Brief history

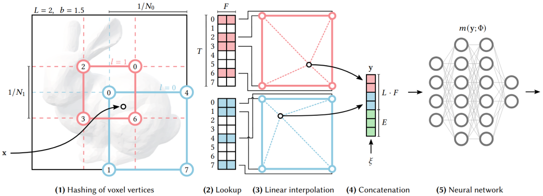
↦ 2021: First ever compression-focused INR application to images (**COIN**) [2].



↦ 2022: More complex INR-based image compression including positional encoding and meta-learning (Strumpler et al.) [7].

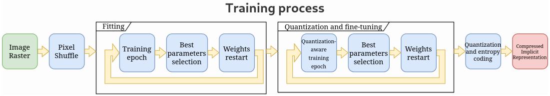
Method	Time (min:sec)	PSNR (dB)	SSIM
COIN	16:04	22.53dB	0.475
Strumpler	24:15	46.73dB	0.993
NIF	2:27	47.53dB	0.997

↦ 2022: Multi-resolution table-grids and hash encoding [5] bring ultra-fast training.

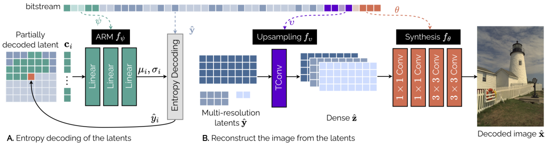


Brief history

↦ 2023: Neural Imaging Format [1] introduces faster encoding through a well-structured training process.

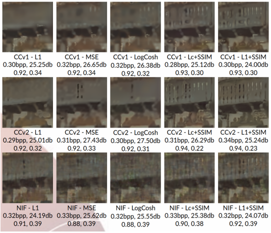


↦ 2023: COOL-CHIC [3] leverages multi-resolution two-dimensional latents to replace positional features.

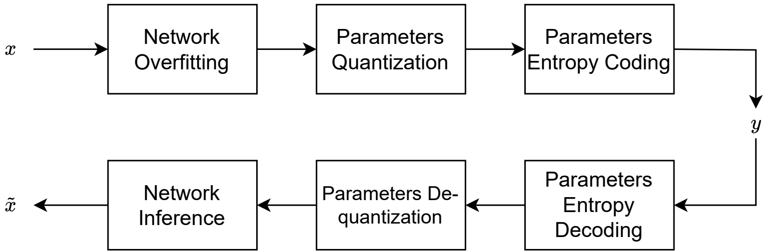


↦ 2024 (Today): Catania and Allegra explore the use of different loss functions to enhance the visual quality achieved by INR-based codecs

TP1.PC: Neural Image and Video Compression - III, Tuesday, 29 October, 14:30 - 16:00



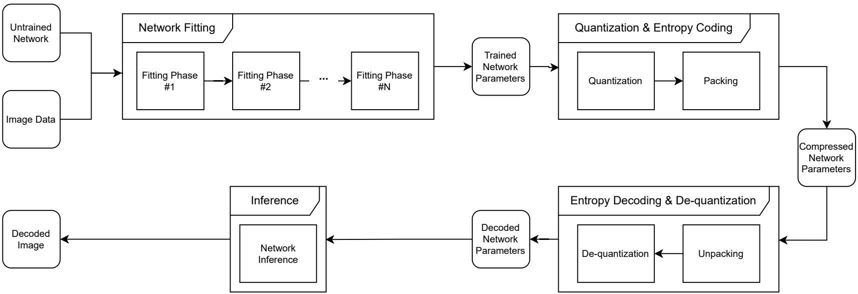
Implicit Image Compression



- The transform step, made of complex hand-crafted heuristics, is replaced by the training of a neural network.
- Inference on the decoder side can be performed at **any resolution**.

Table of Contents

Code Data Flow



Tutorial Workplan

Download the project base from
https://github.com/aegroto/icip-2024-inr-images-tutorial (*live* branch).
Implementation tasks:

- Image data loading
- SIREN and auxiliary modules (e. g. positional encoder)
- Configurable training
- Quantization and packing

Bibliography I

L. Catania and D. Allegra.
Nif: A fast implicit image compression with bottleneck layers and modulated sinusoidal activations.
In *ACM Multimedia*, 2023.

E. Dupont, A. Goliński, M. Alizadeh, Y. W. Teh, and A. Doucet.
Coin: Compression with implicit neural representations.
In *arXiv:2103.03123*, 2021.

T. Ladune, P. Philippe, F. Henry, G. Clare, and T. Leguay.
Cool-chic: Coordinate-based low complexity hierarchical image codec.
In *ICCV*, 2023.

B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng.
Nerf: Representing scenes as neural radiance fields for view synthesis.
In *ECCV*, 2020.

Bibliography II

T. Müller, A. Evans, C. Schied, and A. Keller.
Instant neural graphics primitives with a multiresolution hash encoding.
In *SIGGRAPH*, 2022.

V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein.
Implicit neural representations with periodic activation functions.
In *NeurIPS*, 2020.

Y. Strümpler, J. Postels, R. Yang, L. V. Gool, and F. Tombari.
Implicit neural representations for image compression.
In *ECCV*, 2022.

Bibliography III

M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng.
Fourier features let networks learn high frequency functions in low dimensional domains.
In *NeurIPS*, 2020.