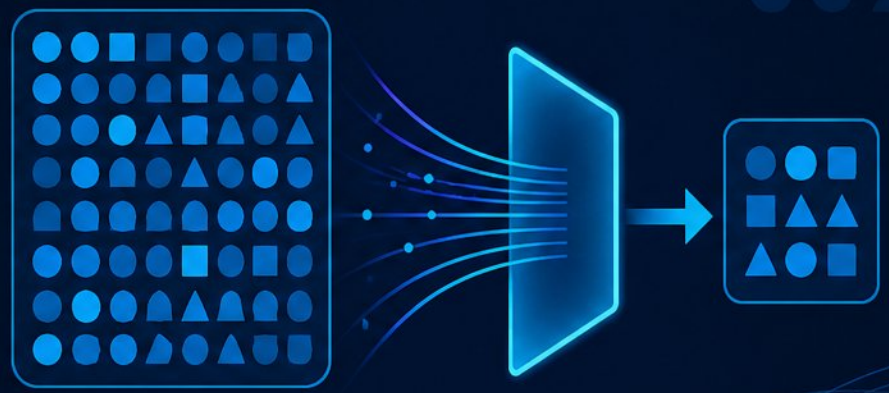
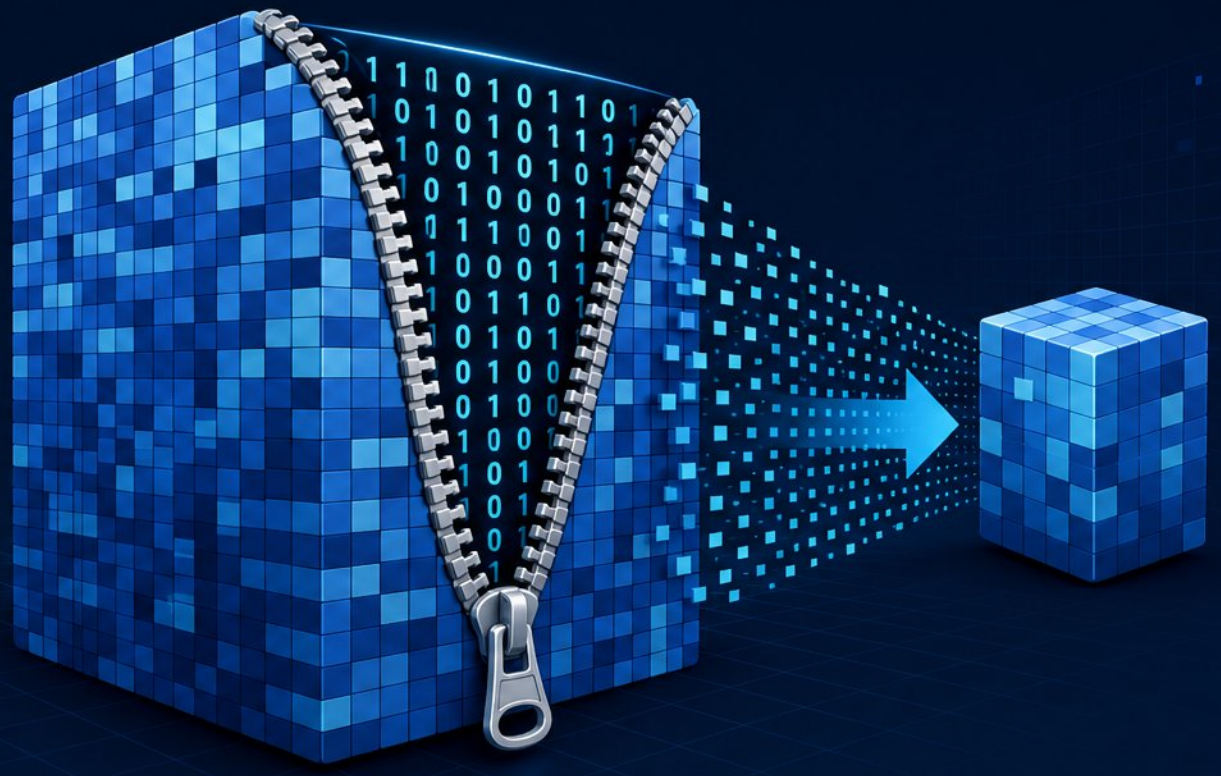
Data Compression:
Finding Patterns to Store Less

- What this covers: why files shrink, and the core ideas behind data compression
- For technical beginners who work with files, data, or storage
- The big question: How can a file get smaller without losing its meaning?
- Roadmap: patterns, repetition, two families of compression, real-world examples



What Data Compression Actually Is

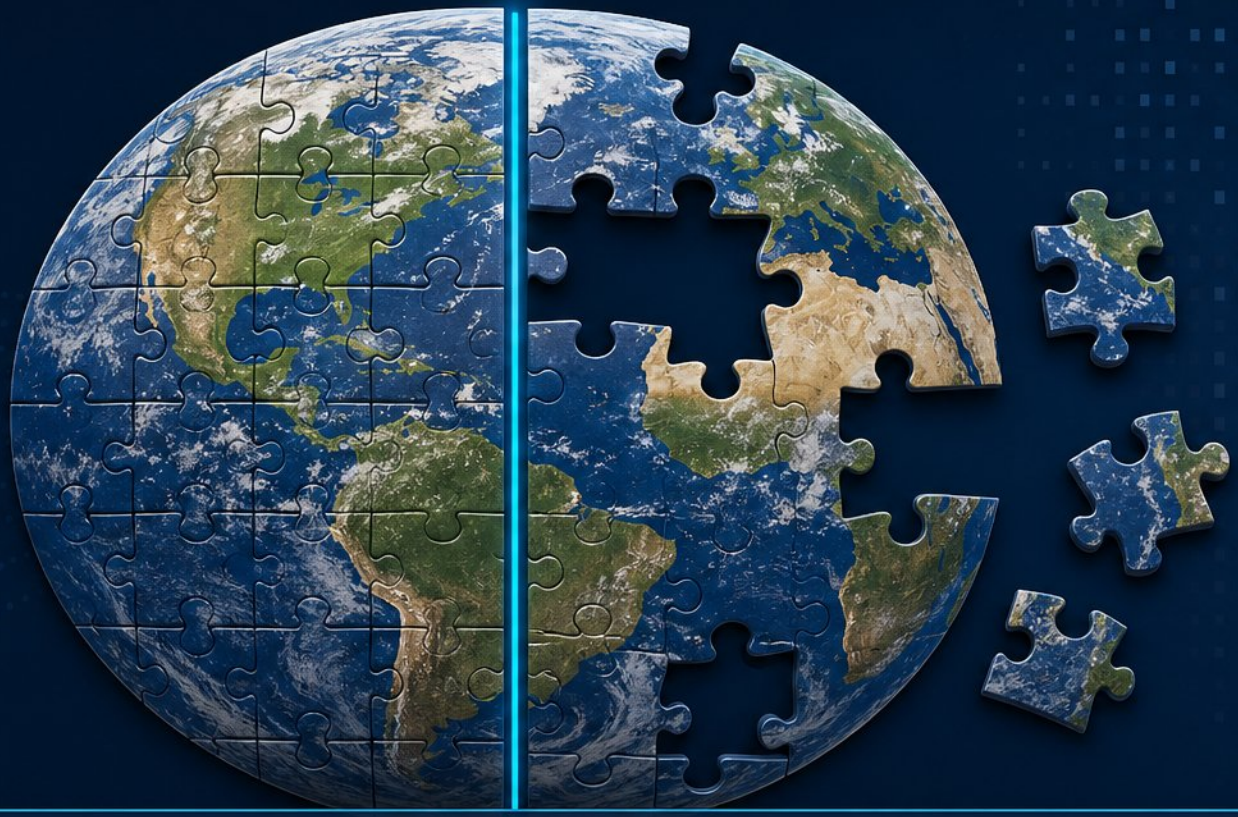
- Compression: encoding information with fewer bits than the original
- Works because real-world data contains patterns, not pure randomness
- Trades processing time and memory for storage or bandwidth savings
- Everyday examples: ZIP files, website images, streaming video, phone photos



Patterns: The Hidden Structure That Makes Compression Work

- Real-world data is full of repetition: characters, words, colors, and silence.
- Predictable, redundant data compresses well; truly random data does not.
- Compression algorithms find these patterns and describe them with shorter codes.
- Run-Length Encoding: replace 'AAAA' with 'A4' to save space instantly.

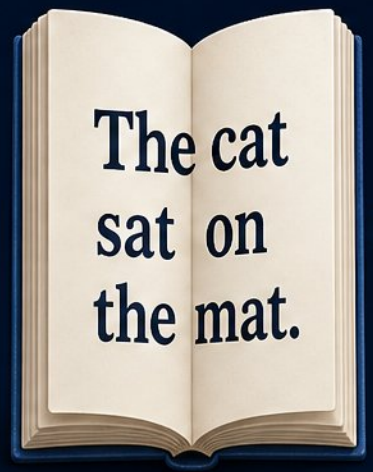
Two Fundamental Approaches: Lossless and Lossy Compression



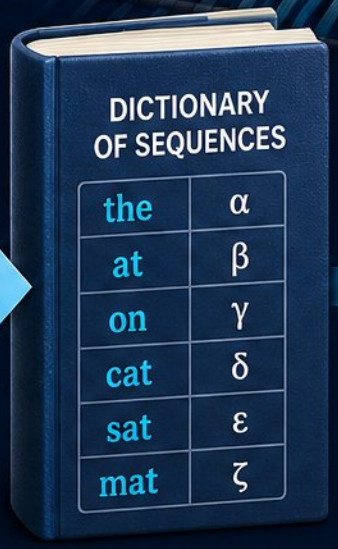
- **Lossless:** perfect reconstruction, bit-for-bit identical to original
- **Lossy:** permanently discards some data for much smaller files
- Use lossless for text, code, documents, spreadsheets, archives
- Use lossy for photos, audio, video, streaming—where perception forgives

How Lossless Compression Exploits Repetition

- **Dictionary-based:** build a table of repeated sequences, replace them with shorter back-references (LZ77).
- **Statistical:** assign shorter codes to frequent symbols, longer codes to rare ones (Huffman).
- **Combined approach:** modern tools like ZIP use LZ77 to find duplicates, then Huffman to encode the results.
- **Step-by-step:** "The cat sat on the mat" → LZ77 spots repeated "at" and "the" → Huffman shrinks frequent letters.



Original Text
(Long & Redundant)



Dictionary / Codes
(Shorter References)

Compressed Output
(Brief Symbol Stream)

Why Some Files Shrink Dramatically—and Others Barely at All

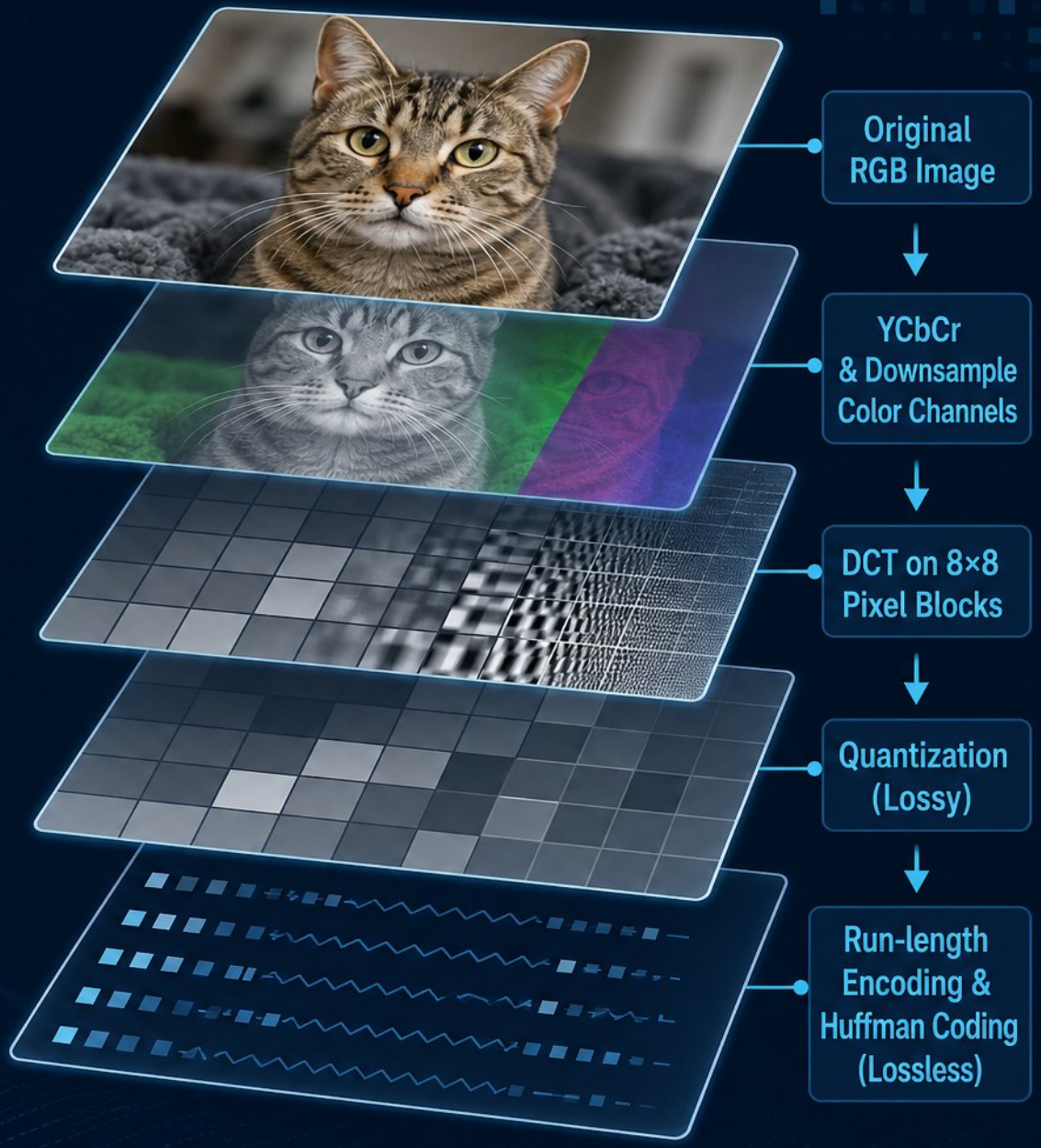
- Highly repetitive content compresses a lot: text files, BMP images, uncompressed WAV audio.
- Already-compressed or random-like data compresses poorly: JPEG, MP3, encrypted files.
- Attempting to recompress an already compressed file often makes it larger.
- A practical rule: compress before you encrypt, and avoid double compression.

Lossy Compression: Trading Detail for Size

- Lossy compression permanently discards data to achieve much smaller file sizes
- Targets information human senses are less likely to notice (high-frequency detail, quiet sounds)
- In images: subtle color variations and fine textures are simplified
- In audio: sounds masked by louder ones are removed using psychoacoustic models
- Quality slider controls the trade-off: smaller files vs. higher fidelity

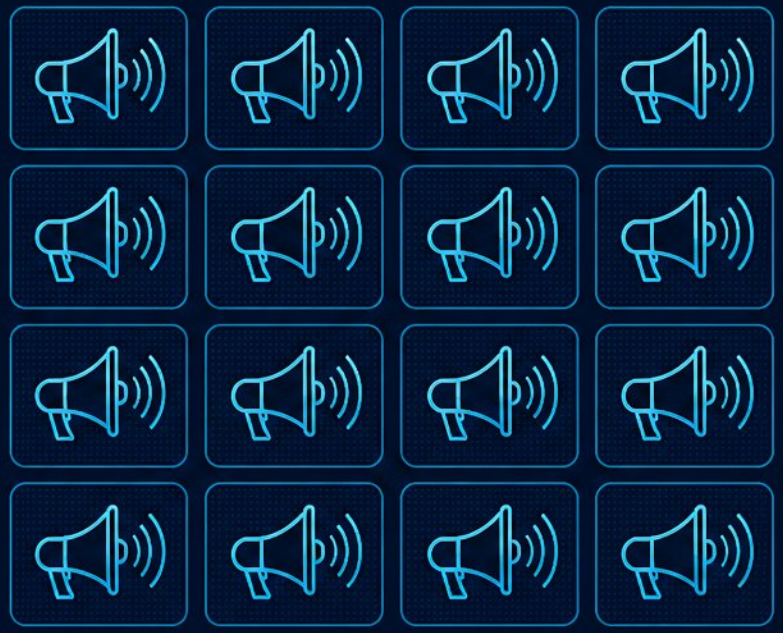
Inside JPEG: How Photos Become Smaller

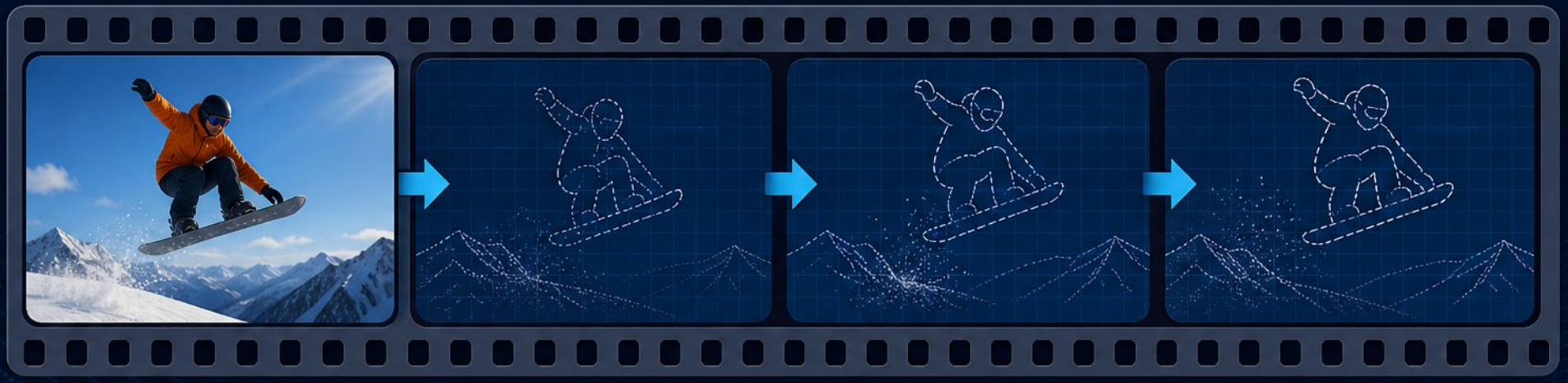
- Convert from RGB to YCbCr to separate brightness from color
- Downsample color channels—our eyes are less sensitive to color detail
- Apply Discrete Cosine Transform (DCT) to 8×8 pixel blocks
- Quantization: the lossy step that discards high-frequency data
- Run-length encoding and Huffman coding compress the remaining data losslessly



Inside MP3 and Video: Perceptual Tricks in Audio and Motion

- MP3 and AAC discard sounds masked by louder ones using a model of human hearing.
- Video codecs store most frames as differences from a reference frame, not full images.
- I-frames are full images; P-frames and B-frames store only what changed.
- This temporal redundancy is massive, making 4K streaming possible.





The Trade-Off: Compression Speed vs. File Size vs. Fidelity

- Higher compression ratios demand more CPU time and computing power.
- Lossless formats are safe for editing; lossy files degrade with each re-save.
- Best practice: archive editable masters as lossless, deliver finals as lossy.
- Modern web formats like WebP and AVIF balance quality and small file sizes.

Compression in Everyday Tools and Formats

- **ZIP and 7z:** lossless archiving for documents, code, and folders
- **PNG:** lossless for screenshots, logos, and sharp graphics
- **JPEG, HEIC, WebP, AVIF:** lossy for photos and web images
- **MP3, AAC, Opus:** lossy audio for streaming and portable music
- **FLAC and ALAC:** lossless audio for archiving and mastering

Choosing the Right Format for the Job

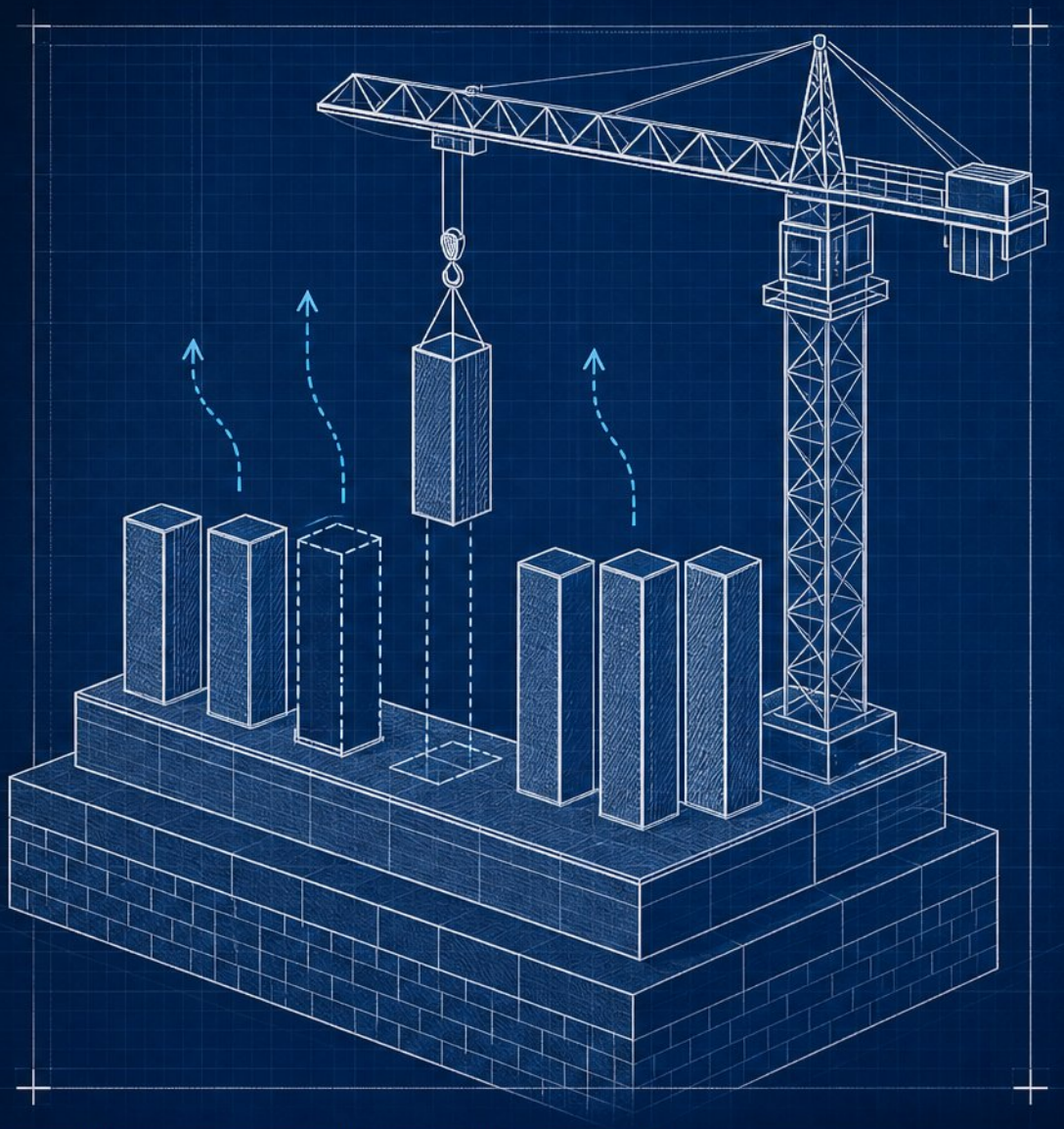
- **Photographs:** use lossy JPEG, WebP, or AVIF for the best file size to quality balance.
- **Logos and screenshots:** use lossless PNG or lossless WebP to keep edges sharp.
- **Audio masters:** FLAC or WAV.
  Audio delivery: AAC or Opus.
  Video delivery: H.264, H.265, or AV1.
- **Documents and code:** lossless archives like ZIP, GZIP, or 7z.
- **Always ask:** will this be edited again, or is this the final distribution form?

Common Mistakes and Misconceptions

- Saving a JPEG as a PNG makes it larger, not smaller—PNG cannot exploit perceptual redundancy.
- Zipping already-compressed files (JPEG, MP3, MP4) saves almost nothing.
- Re-saving a lossy file repeatedly causes generation loss—each save degrades quality.
- Quality 100 on JPEG produces enormous files for negligible gain over quality 90.

The Pattern-Based Mental Model

- Compression relies on real-world data having redundancy and predictable patterns
- Shannon entropy defines the theoretical limit: random data cannot be compressed
- Lossless methods reduce redundancy; lossy methods permanently discard information itself
- Lossy compression removes information, eliminating redundancy — further compression is impossible



Key Takeaways and When to Apply What You Learned

- Compression works by finding patterns and eliminating redundancy.
- Lossless = perfect reconstruction; lossy = smaller files, some fidelity lost.
- Use lossless for documents, code, archives; lossy for photos, video, streaming.
- Don’t expect magic on already-compressed (JPEG, MP3) or encrypted data.
- Keep lossless masters for editing; export lossy versions only for final distribution.

PersonWise.

PersonWise • AI digital-human course platform

Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/e5f662a1/public