Universitat de Barcelona
Facultat de Matemàtiques i Informàtica

File and image compression

Author: Quim Lagunas Rebollar

Degree in Computer Engineering

Supervisor: Joan Canals Gil

Academic year: 2025–2026

Index

1. Abstract, Resumen and Resum

Abstract

Data compression plays a fundamental role in modern computing systems, enabling efficient storage and transmission of digital information. Given the wide variety of available compression codecs and configurations, objective and reproducible performance evaluation is essential to understand their behavior across different types of data.

This project presents the design and implementation of a benchmarking framework for the evaluation of lossless and lossy compression techniques, with a particular focus on general-purpose file compression and digital image compression. Representative datasets were selected to cover heterogeneous data types, including mixed file collections and image datasets. Relevant performance metrics, such as compression ratio and throughput, were defined to enable consistent comparison across codecs and compression levels.

The proposed methodology was applied to a set of widely used compression codecs, and extensive experimental evaluations were conducted. For file compression, the analysis highlights how codec performance varies significantly depending on the nature of the data. For image compression, both lossless and lossy configurations were evaluated, revealing trade-offs between compression efficiency and processing performance.

The results are analyzed and visualized to identify trends, strengths, and limitations of the evaluated codecs under comparable conditions. The findings provide practical insights into the suitability of different compression approaches depending on the data characteristics and application requirements. This work contributes a systematic and extensible benchmarking approach that can be used as a basis for future comparative studies in data compression.

Resumen

La compresión de datos juega un papel fundamental en los sistemas informáticos modernos, permitiendo el almacenamiento y transmisión eficientes de la información digital. Dada la gran variedad de códecs y configuraciones de compresión disponibles, la evaluación objetiva y reproducible del rendimiento es esencial para entender su comportamiento en distintos tipos de datos.

Este proyecto presenta el diseño y la implementación de un marco de referencia para la evaluación de técnicas de compresión sin pérdidas y pérdidas, con un enfoque particular en la compresión de archivos de uso general y la compresión de imágenes digitales. Se seleccionan conjuntos de datos representativos para cubrir tipos de datos heterogéneos, incluyendo colecciones de archivos mixtos y conjuntos

de datos de imágenes. Se definen métricas de rendimiento relevantes, como la relación de compresión y el rendimiento, para permitir una comparación coherente entre códecs y niveles de compresión.

La metodología propuesta se ha aplicado a un conjunto de códecs de compresión ampliamente utilizados y se han llevado a cabo amplias evaluaciones experimentales. Para la compresión de archivos, el análisis destaca cómo el rendimiento del códec varía significativamente según la naturaleza de los datos. Para la compresión de imágenes, se han evaluado las configuraciones sin pérdidas y pérdidas, revelando los compromisos entre la eficiencia de la compresión y el rendimiento del procesamiento.

Los resultados se han analizado y visualizado para identificar tendencias, puntos fuertes y limitaciones de los códecs evaluados en condiciones comparables. Los resultados proporcionan información práctica sobre la adecuación de distintos enfoques de compresión en función de las características de los datos y los requisitos de la aplicación. Este trabajo aporta un enfoque de benchmarking sistemático y extensible que puede utilizarse como base para futuros estudios comparativos en compresión de datos.

Resum

La compressió de dades juga un paper fonamental en els sistemes informàtics moderns, permetent l'emmagatzematge i la transmissió eficients de la informació digital. Donada la gran varietat de còdecs i configuracions de compressió disponibles, l'avaluació objectiva i reproduïble del rendiment és essencial per entendre el seu comportament en diferents tipus de dades.

Aquest projecte presenta el disseny i la implementació d'un marc de referència per a l'avaluació de tècniques de compressió sense pèrdues i amb pèrdues, amb un enfocament particular en la compressió d'arxius d'ús general i la compressió d'imatges digitals. Es seleccionen conjunts de dades representatius per cobrir tipus de dades heterogenis, incloent col·leccions d'arxius mixtes i conjunts de dades d'imatges. Es defineixen mètriques de rendiment rellevants, com ara la relació de compressió i el rendiment, per permetre una comparació coherent entre còdecs i nivells de compressió.

La metodologia proposada s'ha aplicat a un conjunt de còdecs de compressió àmpliament utilitzats i s'han dut a terme àmplies avaluacions experimentals. Per a la compressió d'arxius, l'anàlisi destaca com el rendiment del còdec varia significativament segons la naturalesa de les dades. Per a la compressió d'imatges, s'han avaluat les configuracions sense pèrdues i amb pèrdues, revelant els compromisos entre l'eficiència de la compressió i el rendiment del processament.

Els resultats s'han analitzat i visualitzat per identificar tendències, punts forts i limitacions dels còdecs avaluats en condicions comparables. Els resultats proporcionen informació pràctica sobre l'adequació de diferents enfocaments de compressió en funció de les característiques de les dades i els requisits de l'aplicació. Aquest treball aporta un enfocament de benchmarking sistemàtic i extensible que es pot utilitzar com a base per a futurs estudis comparatius en compressió de dades.

Keywords: data compression, file compression, benchmarking, lossless compression, lossy compression, image compression, performance evaluation

2. Introduction and motivations

Data compression is a fundamental component of modern computing systems, with a direct impact on storage efficiency, network performance, and computational cost. It is present in almost every layer of today's digital infrastructure, from file storage and software distribution to web communication, multimedia delivery, and large-scale data processing. Despite its ubiquity, compression often operates transparently to end users and developers, remaining largely invisible unless performance issues arise.

One of the primary motivations for this project is the contrast between the widespread daily use of compression and the limited understanding of its practical implications. Compressed formats are routinely used to store files, transmit data over networks, and deliver images and web content, yet the choice of compression algorithm, configuration, or format is frequently made by convention rather than by informed analysis. As a result, decisions that significantly affect performance, storage requirements, and resource consumption are often taken without a clear understanding of the trade-offs involved.

Although many compression algorithms are well established and have been studied for decades, the compression ecosystem remains highly fragmented. A large number of codecs coexist, each designed with different goals in mind, such as maximizing compression ratio, minimizing latency, reducing memory usage, or ensuring broad compatibility. In addition, modern codecs expose multiple configuration parameters, including compression levels, window sizes, block sizes, and optional features such as dictionary training. This diversity raises a natural question: why are there so many different compression algorithms, and under what conditions does each one perform best?

This project is further motivated by the observation that, even for experienced practitioners, reliable and up-to-date comparative information is often scarce. Many existing benchmarks are outdated, focus on synthetic data, or evaluate codecs in isolation without consistent methodology. Consequently, real-world decisions are frequently based on incomplete data, anecdotal evidence, or default settings rather than systematic evaluation.

The motivation of this work is therefore twofold. First, to highlight the practical importance of compression and make its performance characteristics more visible and understandable. Second, to provide an experimental, data-driven comparison of widely used compression codecs, focusing on realistic datasets and reproducible measurements. Rather than proposing new algorithms, this project aims to clarify existing trade-offs and provide concrete insights that can support informed codec and configuration selection in real-world scenarios.

By systematically evaluating file and image compression techniques under comparable conditions, this work seeks to bridge the gap between the pervasive use of compression in everyday computing and the limited practical understanding of its behavior and implications.

3. Objectives

The main objective of this project is to design, implement, and evaluate a comprehensive benchmarking framework for lossless and lossy data compression, with a particular focus on file-based datasets and digital images, in order to analyze the performance of different compression codecs under comparable conditions.

To achieve this general objective, the following specific objectives were defined:

1. Study and characterize data compression techniques, with emphasis on widely used lossless and lossy codecs applicable to general-purpose files and image data.
2. Select representative datasets for file and image compression, ensuring diversity in data types, content, and statistical properties.
3. Define relevant evaluation metrics for compression performance, including compression ratio and throughput, allowing objective comparison between codecs.
4. Design and implement an automated benchmarking methodology capable of evaluating multiple codecs and compression levels in a reproducible and systematic manner.
5. Perform experimental evaluations of file compression codecs across heterogeneous datasets, analyzing their behavior with respect to different data types.
6. Conduct lossless and lossy image compression experiments, studying the impact of compression parameters on compression efficiency and performance.
7. Analyze and visualize the obtained results, enabling clear interpretation of trends, trade-offs, and limitations across codecs and data types.
8. Compare and discuss the performance of the evaluated codecs, identifying strengths and weaknesses depending on the nature of the data and the compression configuration.
9. Synthesize the experimental findings into coherent conclusions that highlight the main contributions and practical implications of the study.

4. Planification

This section describes the planning of the Final Degree Project, including the tasks involved, their temporal distribution, and the estimated effort devoted to each phase. In accordance with the regulations, both the initial planning defined at the start of the project and the final adjusted planning actually followed are presented. The planning is summarized using a Gantt diagram.

At the start of the project, an initial planning was defined to guide the development of the project. The work was structured on a weekly basis, with milestones associated with the different sections of the project. The initial planning was as follows:

- 20/10 - 27/10: Initial project state and preliminary analysis.
- 27/10 - 03/11: Current state analysis and refinement of objectives.
- 03/11 - 10/11: Dataset selection and definition of evaluation metrics.
- 10/11 - 17/11: Methodology definition and preparation of file-based datasets.
- 17/11 - 24/11: File compression experiments.
- 24/11 - 08/12: Continuation of file compression experiments.
- 08/12 - 15/12: Image compression experiments.
- 15/12 - 22/12: Extension of image compression experiments.
- 22/12 - 29/12: Study of preprocessing transformations.
- 29/12 - 02/01: Compatibility analysis between formats and codecs.
- 02/01 - 09/01: Synthesis of results and conclusions.
- 09/01 - 12/01: Writing of the introduction and abstract.

This initial plan assumed that the experimental phases for file and image compression could be completed within the allocated time and that preprocessing transformations could be analyzed as an independent final step.

During the execution of the project, significant deviations from the initial planning occurred. In particular, the file compression benchmarking and, more notably, the image compression experiments required substantially more time than originally anticipated. This was due to the complexity of implementing multiple codecs, handling different data types, debugging failures, and validating the obtained results.

As a consequence:

- The experimental phases for file and image compression were extended well beyond their initially planned time slots.
- Several planned tasks had to be compressed into the final two weeks of the project.
- The analysis of preprocessing transformations was ultimately dropped, as priority was given to completing the core experimental evaluation and ensuring the correctness and clarity of the main results.

The final effective planning was therefore the already mentioned but with these changes:

- 17/11 - 22/12: File compression experiments.
- 22/12 - 09/01: Image compression experiments.
- 09/01 - 15/01: Compatibility analysis, synthesis of results, conclusions, and final writing (introduction, abstract, and full document revision).

The total duration of the project was approximately **13 weeks**. The estimated time effort distribution is summarized below:

- File compression experiments: ~35%
- Image compression experiments: ~35%
- Result analysis and synthesis: ~15%
- Writing and final revision: ~15%

This redistribution reflects the experimental complexity of the project and the need to prioritize core contributions over secondary planned extensions.

Figure 1 presents the Gantt diagram corresponding to the initial planning defined at the start of the project, showing the expected schedule and duration of each task.

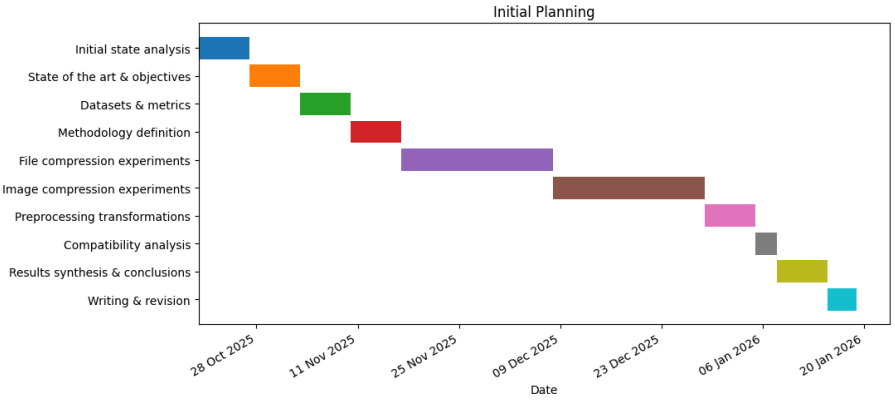

Figure 1. Gantt chart of the Initial Planning

Figure 2 presents the Gantt diagram corresponding to the final planning actually followed, illustrating the extension of the file and image compression experimental phases, the removal of preprocessing transformations, and the concentration of analysis and writing tasks toward the end of the project.

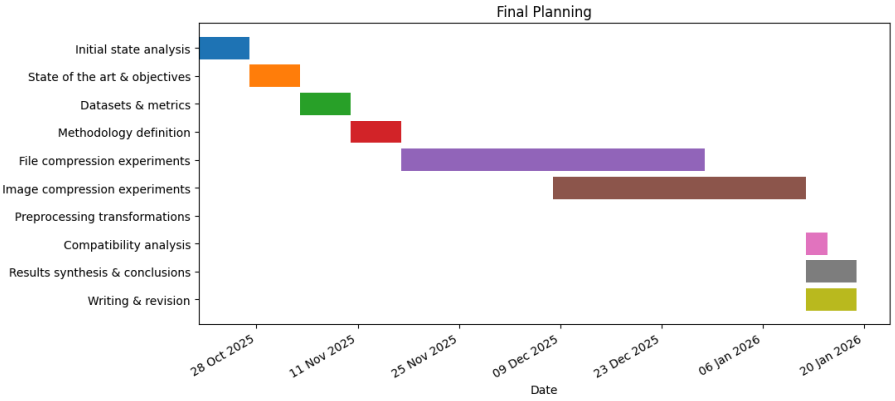

Figure 2. Gantt chart of the Final Planning

5. Current state of art

File and image compression is a mature yet continually evolving technological field. Its importance lies in the constant demand to reduce data size for efficient storage, transmission, and processing. In storage systems, smaller files lower disk usage and I/O overhead. In data transmission, especially over the web and mobile networks, compression directly improves latency and bandwidth efficiency. In computation-intensive environments, reducing data size also decreases memory load and energy consumption, making compression a key component of modern performance optimization.

The current landscape of data and image compression is characterized by a relatively small set of mature, widely deployed algorithms that position themselves differently in terms of compression efficiency, speed, memory usage, and compatibility. Rather than attempting an exhaustive survey of all existing codecs, this project focuses on a representative subset of algorithms that are actively used in real-world systems, and are well documented.

Lossless compression algorithms aim to reduce file size without losing any information. Many widely used formats are based on the LZ (Lempel-Ziv) family of algorithms, first introduced in the late 1970s. The Lempel-Ziv method identifies and replaces repeated data patterns with shorter references to earlier occurrences, enabling efficient encoding of redundancy. This principle has influenced nearly every modern lossless codec.

For lossless compression of general files (text, structured data, and binaries), the algorithms selected are Deflate, Zstandard (zstd), Brotli, and LZMA. These codecs are not only influenced by LZ algorithms but also belong to the same broad family of dictionary-based compression schemes but differ significantly in design goals and practical use.

Deflate (RFC 1951) is included primarily as a baseline and legacy reference. Defined in the early 1990s and still used in formats such as ZIP and gzip. Deflate uses a combination of the LZ77 algorithm and Huffman coding, with efficiency comparable to the best currently available general-purpose compression methods [1]. Its own specification and long-standing usage emphasize portability and simplicity rather than optimal compression ratio or speed by modern standards. While it is generally outperformed by newer codecs, its ubiquity makes it a useful comparison point when evaluating gains offered by more recent approaches.

Zstandard (zstd) is a fast compression algorithm, providing high compression ratios [2]. According to its official documentation, zstd is designed to scale across a wide range of compression levels, from very fast modes competing with LZ4 to high-density modes approaching LZMA, while keeping decompression speed

consistently high. Zstd also highlights advanced features such as trained dictionaries, large window sizes, and fine-grained parameter control, positioning itself as a modern replacement for Deflate in many applications.

Brotli (RFC 7932) defines a lossless compressed data format that compresses data using a combination of the LZ77 algorithm and Huffman coding [3]. It presents itself as a codec optimized for distribution of data over the web, particularly text-based resources, and emphasizes strong compression density at moderate decoding cost, achieved through a combination of a large static dictionary and entropy coding. Brotli is commonly used in HTTP content encoding, where its design favors smaller payload sizes at the expense of slower compression compared to zstd. In contrast to zstd, Brotli offers fewer tunable parameters, reflecting its goal of predictable performance in web delivery scenarios.

LZMA (Lempel-Ziv-Markov chain-Algorithm), as implemented in the LZMA SDK and popularized by 7-Zip, is a general-purpose compression algorithm with high compression ratio and fast decompression [4]. LZMA is based on LZ77 and range coding algorithms [4]. Its documentation describes large dictionaries, sophisticated match finding, and range coding as key features that enable superior density, especially for large or homogeneous datasets. However, these benefits come with clearly acknowledged disadvantages: high memory consumption and slow compression and decompression times. For this reason, LZMA is typically positioned for archival use rather than latency-sensitive workloads.

For image compression, the project evaluates both lossy and lossless formats, focusing on codecs that define themselves as successors or alternatives to established standards.

JPEG is included as a reference standard, not because it represents current best performance, but because it remains the most widely supported lossy image format, and is still the most dominant still image format around [5]. Its standard documentation emphasizes broad compatibility and simplicity, while modern analyses consistently identify limitations in compression efficiency and artifact handling at low bitrates.

PNG (RFC 2083, or Portable Network Graphics) plays a similar role for lossless images; it is designed to work well in online viewing applications [6]. It is described as a robust, patent-free format with predictable behavior and wide support, particularly suitable for graphics, UI elements, and images requiring exact reconstruction. However, its compression efficiency is generally surpassed by newer lossless image codecs.

WebP, developed by Google, explicitly positions itself as a modern replacement for JPEG and PNG, it is a modern image format that provides superior lossless and lossy compression for images on the web [7]. According to its official documentation,

WebP achieves smaller file sizes than JPEG at equivalent visual quality and denser lossless compression than PNG in many cases, while maintaining reasonable encoding and decoding complexity.

AVIF (AV1 Image File), based on AV1 intra-frame coding, is an image format designed to offer superior image compression and quality compared to traditional formats [8]. The Alliance for Open Media describes AVIF as delivering “significant compression gains” over JPEG and WebP, especially for photographic content. These gains, however, are accompanied by higher computational cost, which is explicitly acknowledged in its design trade-offs.

JPEG XL presents itself as a next-generation image coding system designed to unify lossy and lossless compression while also enabling efficient transcoding from legacy JPEG. Its standard and accompanying white papers emphasize a balance between compression efficiency, fast decoding, advanced features (HDR, wide color gamut), and long-term archival suitability. JPEG XL is particularly optimised for responsive web environments [9].

Compatibility and licensing strongly influence real-world adoption. JPEG and PNG remain universally supported; WebP and AVIF are now compatible with roughly 90-95 % of browsers. JPEG XL’s adoption uncertainty makes it promising in research contexts but unreliable for broad deployment.

The emerging practice of using trained dictionaries, especially in zstd and brotli, further improves compression for repetitive, domain-specific datasets such as JSON logs or configuration files.

Modern compression ecosystems emphasize optimized command-line tools (e.g., zstd, brotli, xz, cwebp, avifenc, cjxl) that allow fine-tuning of compression levels and resource usage. Multi-threading and adjustable presets enable the same codec to target distinct use cases, from fast on-the-fly compression to archival storage.

Current research and industrial trends include:

- Reducing encoding latency while maintaining or improving compression ratios.
- Introducing domain-specific compression, such as JSON-aware or structured-text optimizations.
- Standardizing perceptual quality metrics (SSIM, PSNR, LPIPS) for fair image evaluation.

- Enhancing interoperability, HDR support, and metadata preservation in next-generation image formats.

As a summary, the current landscape reveals coexistence between legacy codecs (stable and universally supported) and modern alternatives that offer higher efficiency but face uneven adoption. For general data, zstd and brotli deliver the best compromise between speed, ratio, and resource use. For images, AVIF and WebP currently lead the transition to more advanced and compact formats. Ultimately, codec selection depends on content type, system constraints, and compatibility requirements.

6. Reproducible methodology

The project followed a rigorous and fully reproducible methodology to ensure that all experiments can be repeated and independently validated. This methodology defines how measurements are collected, how environments are controlled, and how experiments are structured across both file and image compression tasks.

1. Experimental Environment

All experiments were executed on a fixed and documented hardware setup to avoid inconsistencies caused by hardware variability. This includes:

- CPU model, core count, and frequency
- RAM capacity and memory speed
- Storage type (e.g., NVMe SSD)
- Operating system version
- Compiler and library versions used to build the codecs (where applicable)

Hyper-threading, CPU frequency scaling (boost/turbo), and power-saving features were disabled when possible, to avoid fluctuations in timing results.

All codecs were invoked using command-line tools with explicit parameters, ensuring that their behavior is transparent and reproducible.

2. Experimental Procedure

Each experiment followed a standardized procedure:

• 2.1. Fixed input datasets

Datasets were prepared in advance, stored in unmodified form, and never altered during testing.

• 2.2. Warm-up and repetitions

Each measurement was repeated five times, preceded by a warm-up run to mitigate cold-cache effects. Median values were used as the primary metric.

• 2.3. Cache and I/O control

- Disk caches may be cleared between runs when supported by the system.
- Input and output files were explicitly separated to avoid OS-level buffering artifacts.

- Only local storage was used (no network filesystems).

• 2.4. Time and memory measurement

Execution time was measured using stable, high-resolution timers.

3. Parameter Selection and Documentation

For each codec, the methodology includes:

- Testing multiple compression levels (low/medium/high)
- Controlling window size and block size.
- Running identical parameter sets across all files of the same category

4. Image Compression Experiments

Image compression experiments follow a controlled pipeline:

- All images are normalized into a reference format (usually PNG lossless).
- Experiments test multiple target sizes (e.g., 50 KB, 100 KB, 200 KB) by adjusting codec quality parameters.
- Both objective metrics (PSNR, SSIM, LPIPS) and controlled visual inspection are performed.
- Subsampling modes (4:4:4 vs. 4:2:0) are compared when supported by the codec.

Different image categories (natural, UI/text, illustrations, textures) are evaluated separately to capture domain-specific behavior.

5. Result Logging and Data Integrity

All results are saved in:

- Structured logs (CSV/JSON)
- Scripts for reproduction (shell scripts or Python notebooks)
- Separate folders per experiment (e.g., “zstd-levels”, “avif-quality-curve”, “brotli-dictionary-tests”)

Any anomalous results (e.g., unexpected spikes in runtime or corrupted output) are marked and re-tested.

6. Reproducibility Assets

The final project deliverables include:

- A dataset manifest listing all inputs, checksums, and licenses

Together, these components ensure that every experiment can be independently rerun, producing identical or statistically consistent results.

7. Datasets and sample selection

A rigorous and representative dataset design is essential to ensure that compression experiments yield meaningful and generalizable results. The datasets used in this project cover a diverse range of file types and image contents, enabling a balanced evaluation across both textual and visual domains.

1. File datasets (general-purpose, lossless)

For general-purpose compression, the selected datasets must reflect the variability found in real-world data. Three main categories were included:

1. Plain text and source code - Collections of programming language source files (e.g., C, Python, JavaScript) and literary or technical text corpora. These datasets are ideal for evaluating compression of highly structured, low-entropy content.

2. Structured data (JSON, CSV, logs) - Files containing hierarchical or tabular data, representing common web and analytical formats. Both homogeneous and heterogeneous collections will be used to test the impact of trained dictionaries on compression efficiency.

3. Binary and mixed folders - Compiled libraries, executables, and mixed-content project directories. These files present challenges for dictionary-based and block-size dependent codecs due to their non-textual entropy patterns.

Publicly available repositories such as Silesia Corpus, Canterbury Corpus, or open-source project archives (e.g., datasets from GitHub or Debian mirrors) will serve as the foundation. Additionally, a small custom corpus will be assembled to represent realistic application logs and configuration files. Each dataset will be documented with metadata describing size, structure, and licensing.

2. Image datasets (lossless and lossy)

Image compression experiments require controlled diversity in both content and complexity. The selected image sets will be organized into four main content categories:

1. Natural photographs - High-resolution camera images covering varied lighting, color, and texture conditions. Public datasets such as Kodak PhotoCD, Tecnick, and DIV2K provide suitable baselines.

2. User interfaces and infographics - Screenshots, UI components, and text-overlaid vector graphics. These are sensitive to color sub-sampling and

edge sharpness, offering a good basis for evaluating perceptual degradation in lossy codecs.

3. Textures and repeating patterns - Synthetic and tiled images representative of 3D assets or web design resources, used to test codecs' efficiency on repetitive high-frequency content.

4. Illustrations and flat-color graphics - Drawings, icons, and synthetic artwork with large uniform color areas, ideal for evaluating codecs' behavior in low-entropy but high-fidelity requirements.

Each subset include multiple resolutions to measure how scaling affects compression efficiency and visual quality. When applicable, images have been stored in uncompressed or high-quality PNG form to prevent pre-existing artifacts from biasing results.

3. Curation criteria

Dataset curation will follow three main principles:

- Representativeness: ensure coverage of common real-world scenarios while avoiding bias toward a single content type or format.
- Diversity: include varying sizes, structures, and complexity levels to observe codec behavior under different conditions.
- Transparency and reproducibility: all datasets will be either publicly available or clearly documented with source, license, and selection rationale.

Licensing will be verified for all included materials to guarantee redistribution or fair-use compliance. Whenever necessary, derived datasets will be annotated with the applied transformations (e.g., cropping, format conversion).

4. Deliverables

The final outcome of this stage will be:

- A catalogue of datasets used in the study, including metadata such as file count, average size, entropy estimates, and licensing information.
- A data inventory appendix in the final report, ensuring that all compression tests are reproducible by third parties.
- A documentation table mapping each dataset to the specific experiments in which it is used (e.g., text vs. image compression, lossy vs. lossless, dictionary vs. baseline).

Name: Silesia Compression Corpus [10]
License: Public domain / free for research use
Data type: Mixed files (text, binaries, databases, image-like binary data)
Total size: 211,938,580 bytes (~212 MB)
Number of files: 12
File list:

- dickens (text)
- mozilla (tarred executables)
- mr (image binary data)
- nci (database)
- ooffice (DLL)
- osdb (MySQL sample database)
- reymont (PDF binary data)
- samba (tarred source code)
- sao (binary data)
- webster (HTML text)
- xml (XML files)
- x-ray (image binary data)

Reason for selection: A standard and widely cited benchmark for evaluating general-purpose lossless compression algorithms across heterogeneous data types.
Representativeness: The corpus exhibits a wide range of entropy characteristics and is highly representative of real-world workloads, including natural language text, executables, structured documents, databases, and scientific binary data.
Transformation applied: Files were decompressed from their original .bz2 format prior to benchmarking.
Intended use: Benchmarking lossless file compression and decompression performance.

Name: Canterbury Corpus [11]
License: Public domain
Data type: Mixed files (text, binaries)
Total size: 2,810,784 bytes (~3 MB)
Number of files: 11
File list:

- alice29.txt (text)
- asyoulik.txt (text)
- cp.html (HTML source)
- fields.c (C source)
- grammar.lsp (LISP source)
- kennedy.xls (Excel spreadsheet)
- lcet10.txt (text)
- plrabn12.txt (text)

- ptt5 (CCITT test set)
- sum (SPARC executable)
- xargs.1 (manual)

Reason for selection: A classical benchmark corpus designed to evaluate compression performance on small and medium-sized files.
Representativeness: Representative of legacy and small-file workloads, including source code, plain text, formatted documents, and binaries.
Transformation applied: None.
Intended use: Evaluation of compression efficiency and overhead for small inputs.

Name: Kodak PhotoCD Image Dataset [12]
License: Research and educational use
Data type: Natural color photographs
Total size: 15,394,305 bytes (~15 MB)
Number of files: 24 images
File list:

- kodim01.png
- kodim02.png
- kodim03.png
- …
- kodim22.png
- kodim23.png
- kodim24.png

Resolution: 768 x 512 pixels
Reason for selection: The Kodak dataset is a historical and widely cited benchmark in image compression research.
Representativeness: Represents natural photographic content with smooth gradients, textures, and color variation.
Transformation applied: None.
Intended use: Benchmarking lossless and lossy image compression algorithms and evaluating rate-distortion behavior.

Name: Tecnick Image Dataset [13]
License: Free for research use
Data type: High-quality photographs
Total size: ~1.0 GB
Number of files: 100 images
File list:

- img_2448x2448_3x16bit_SRC_RGB_almonds.png
- img_2448x2448_3x16bit_SRC_RGB_apples.png
- img_2448x2448_3x16bit_SRC_RGB_baloons.png

- …
- img_2448x2448_3x16bit_SRC_RGB_tools_a.png
- img_2448x2448_3x16bit_SRC_RGB_tools_b.png
- img_2448x2448_3x16bit_SRC_RGB_wood_game.png

Resolution: 2448 x 2448 pixels
Reason for selection: The Tecnick dataset provides high-resolution images specifically curated for compression and codec evaluation.
Representativeness: Highly representative of modern photographic workloads, including high-frequency detail and large image sizes.
Transformation applied: Files were decompressed from their original .tar.bz2 format prior to benchmarking.
Intended use: Evaluation of compression scalability, encoding/decoding performance, and quality metrics at high resolutions.

8. General-purpose files

This section presents and analyzes the experimental results obtained for lossless compression of general-purpose files using the Silesia Compression Corpus. The corpus comprises twelve heterogeneous datasets, including natural language text, structured data, executables, source code, scientific databases, XML collections, and medical images. This diversity allows the evaluation of codec behavior across a wide range of entropy profiles and data characteristics.

All results reported in this section were generated using the custom tool compression_CLI.py, following the reproducible methodology described previously. Each file was compressed independently using multiple codecs and compression levels, and metrics including compression ratio, compression throughput, and decompression throughput were recorded.

Across the full Silesia corpus, the results confirm a clear stratification between codecs optimized for speed, codecs optimized for compression density, and legacy formats optimized for interoperability.

Zstandard (zstd) consistently delivers the highest compression and decompression throughput across nearly all files and compression levels. Even at moderate compression levels, zstd achieves compression ratios close to those of Brotli while maintaining significantly higher encoding speed and extremely fast decoding. This behavior is stable across text-heavy datasets (e.g., dickens, reymont), mixed datasets (e.g., samba, mozilla), and structured datasets (e.g., xml).

Brotli achieves higher compression ratios than zstd at comparable mid-range compression levels, particularly on highly compressible text and structured data. However, this improvement comes at a noticeable cost in compression speed. At higher compression levels, Brotli's encoding throughput decreases sharply, making such configurations unsuitable for latency-sensitive workflows.

LZMA consistently produces the highest compression ratios across almost all datasets, including difficult-to-compress binaries and XML collections. This confirms its suitability for archival use. However, its compression throughput is orders of magnitude lower than that of the other codecs, and even decompression is significantly slower. These characteristics make LZMA impractical for interactive or high-throughput systems.

Deflate (gzip) remains the weakest codec in terms of compression ratio, particularly on text-heavy and structured datasets, where modern codecs exploit redundancy more effectively. Nevertheless, gzip maintains reasonable and predictable performance, with relatively fast encoding and decoding and universal tool support.

The results also show that codec performance is strongly influenced by the nature of the input data.

Textual datasets exhibit high redundancy and benefit substantially from modern compression techniques. On these files:

- LZMA achieves the highest compression ratios, often significantly outperforming all other codecs.
- Brotli typically ranks second in compression ratio, particularly effective on large, homogeneous text.
- Zstd provides slightly lower compression ratios than Brotli but with vastly superior compression speed.
- Gzip consistently yields the lowest compression ratios.

This confirms that deeper modeling and larger dictionaries provide tangible benefits for text, but only when encoding cost is acceptable.

For executables and mixed binary datasets, compression ratios are lower and differences between codecs narrow:

- All codecs achieve relatively modest compression gains.
- Zstd and gzip often perform similarly in ratio, with zstd maintaining a clear speed advantage.
- LZMA still achieves higher compression, but the marginal gain over zstd or Brotli is smaller than in text datasets.

This suggests diminishing returns for very high-cost compression on already dense binary data.

The medical image datasets behave differently from text or executables:

- Compression ratios are relatively low across all codecs.
- Differences between codecs are less pronounced, as these files already exhibit high entropy.
- Zstd and gzip offer the best balance of modest compression and high speed.

These results reinforce the idea that lossless general-purpose compressors are inherently limited on image-like binary data, motivating the separate evaluation of image-specific codecs in later sections.

A central outcome of the experiments is the clear visualization of trade-offs between compression ratio and throughput. Figure 3 summarizes this relationship by positioning each codec according to its average compression ratio and encoding speed across the Silesia corpus at representative compression levels.

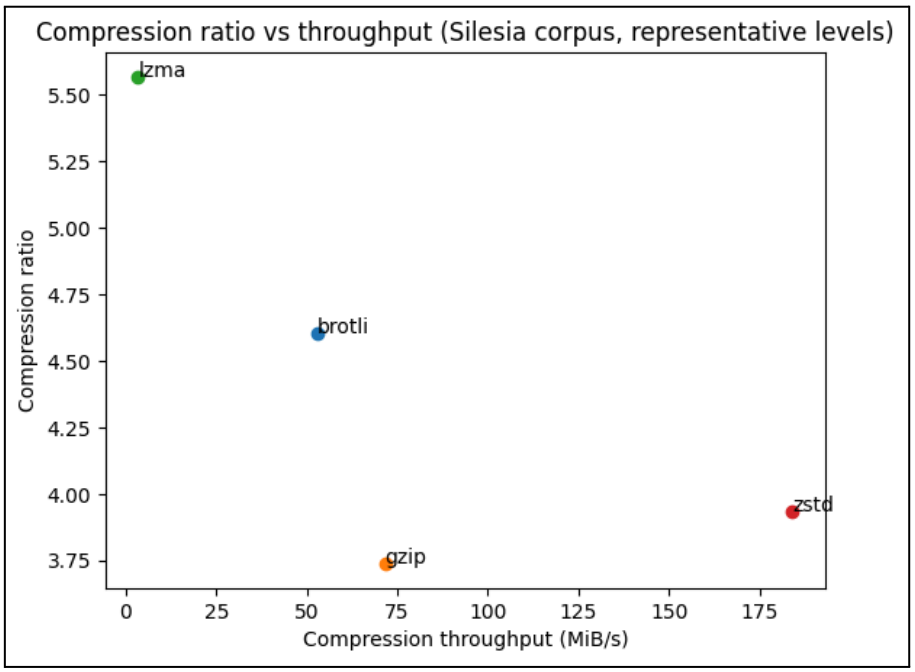

Figure 3. Compression ratio vs throughput (in general-purpose files)

- Increasing compression levels yields diminishing returns in compression ratio but rapidly increasing encoding cost.
- Zstd exhibits the smoothest trade-off curve, maintaining high throughput even as compression level increases.
- Brotli and LZMA show steep performance degradation at higher levels, making such configurations unsuitable for general use.
- Gzip scales predictably but remains capped in achievable compression ratio.

These trends are consistent across all twelve Silesia datasets and support the interpretation that moderate compression levels are optimal for most real-world scenarios.

Codec	Avg. compression ratio (×)	Avg. compression throughput (MiB/s)	Avg. decompression throughput (MiB/s)
brotli (5)	4.60	52.9	265.2
gzip (6)	3.74	71.6	330.0
lzma (6)	5.57	3.4	81.6
zstd (3)	3.93	184.1	359.1

Figure 4. Average performance on the Silesia corpus

Based on the complete Silesia corpus, Figure 4 summarizes the average performance of the evaluated codecs in terms of compression ratio, compression throughput, and decompression throughput, aggregated across all twelve datasets at representative compression levels, and the following comparative conclusions can be drawn:

- Zstandard is the most versatile codec, offering an excellent compromise between compression ratio, encoding speed, and decoding speed across all data types.
- Brotli is advantageous when compression ratio is prioritized over encoding speed, particularly for text and structured data.
- LZMA is best suited for archival use, where maximum compression ratio justifies very high computational cost.
- Deflate (gzip) remains relevant primarily for compatibility and legacy reasons.

This can be better seen in Figure 5 and Figure 6, which visualize the minimum and maximum compression ratios and compression throughputs achieved by each codec across all tested compression levels and data types. By explicitly showing the performance ranges instead of single-point measurements, these figures highlight both the tuning potential and the inherent limitations of each codec, making the trade-offs between compression efficiency and performance more apparent.

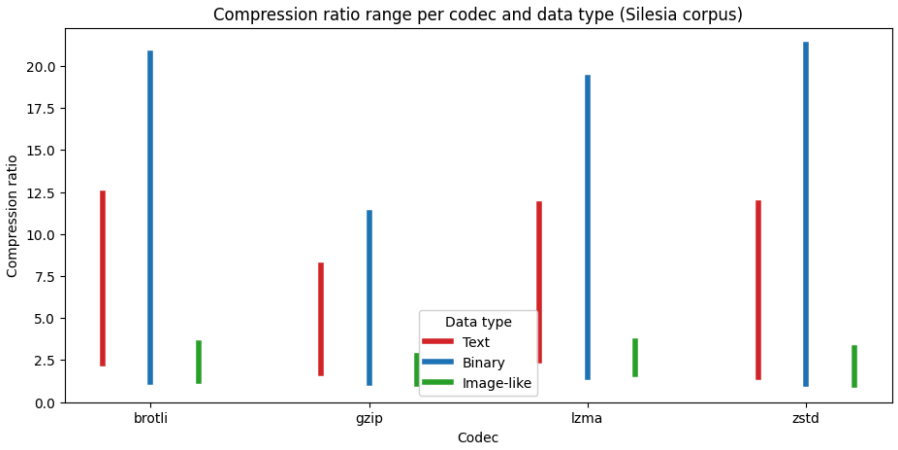

Figure 5. Compression ratio range per codec and data type (in general-purpose files)

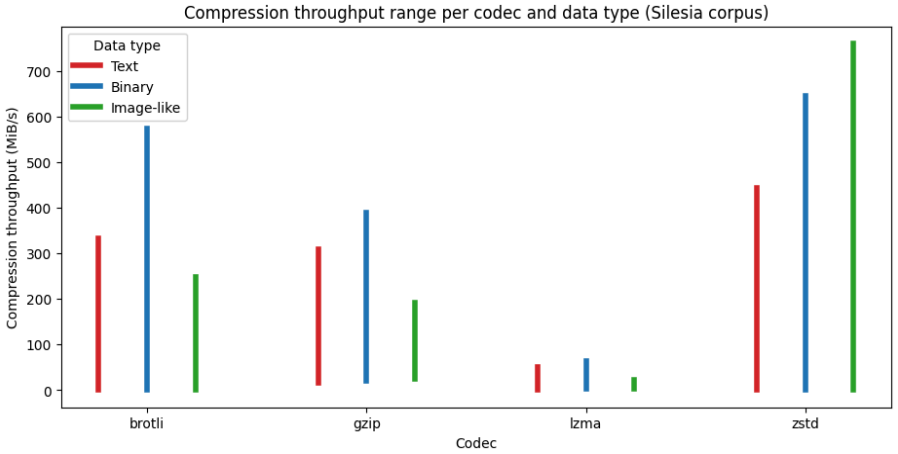

Figure 6. Compression throughput range per codec and data type (in general-purpose files)

9. Images (lossless and lossy)

Lossless image compression differs fundamentally from general-purpose compression in that the input data already exhibits relatively high entropy and strong spatial correlation. As a result, achievable compression ratios are typically much lower, and performance differences between codecs are often dominated by implementation efficiency rather than modeling depth.

To evaluate this scenario, experiments were conducted on a set of widely used photographic images from the Kodak PhotoCD and Tecnick datasets. All images were compressed using multiple lossless image codecs and all available compression levels. The evaluated codecs include PNG (via OptiPNG), WebP lossless, and JPEG XL lossless.

This can be better seen in Figure 7, which summarizes the minimum and maximum compression ratios achieved by each lossless image codec across all tested images and compression levels.

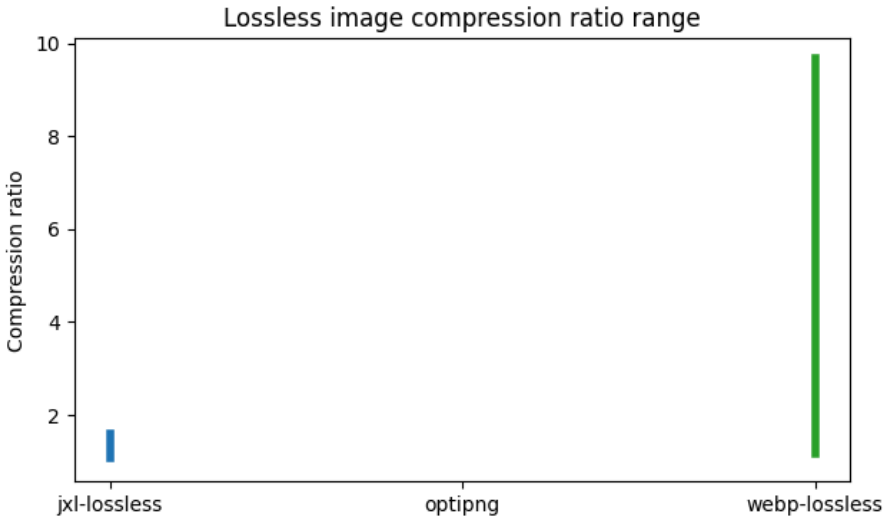

Figure 7. Compression ratio range for lossless image compression

Figure 7 shows that lossless image compression yields significantly smaller compression ratios than those observed for text or structured data, regardless of the codec used. This behavior is expected, as natural images contain less explicit redundancy and more high-frequency information.

Among the evaluated codecs:

- **WebP** lossless achieves the highest maximum compression ratios, with a wide operational range across images and compression levels. This indicates strong adaptability to image content and effective entropy coding.

- **JPEG XL** lossless provides moderate compression gains with a comparatively narrower range, suggesting more conservative but stable behavior.
- **OptiPNG** (PNG) exhibits very limited compression potential, with ratios close to 1× in many cases, reflecting the maturity and limited modeling flexibility of the PNG format.

These results highlight that, even in lossless mode, modern image codecs can outperform legacy formats in terms of compression efficiency.

The performance implications of these compression gains are illustrated in Figure 8, which presents the minimum and maximum compression throughput observed for each codec.

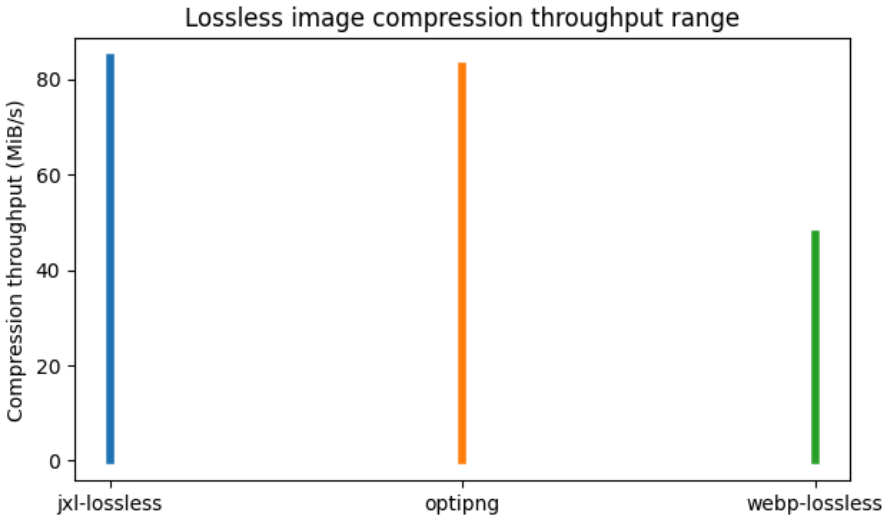

Figure 8. Compression throughput range for lossless image compression

Figure 8 reveals substantial differences in compression throughput across codecs:

- **JPEG XL** lossless achieves the highest peak compression throughput, indicating an implementation optimized for speed and scalability.
- **OptiPNG** shows a wide throughput range, with very low throughput at aggressive optimization settings and much higher throughput when minimal optimization is applied.
- WebP lossless exhibits lower peak throughput than JPEG XL but maintains more consistent performance across levels.

Notably, the throughput ranges span several orders of magnitude, demonstrating that compression level selection has a far greater impact on performance than on compression ratio in the lossless image domain.

Taken together, Figures 7 and Figure 8 illustrate a clear trade-off space for lossless image compression:

- **WebP** lossless prioritizes compression efficiency at the cost of lower throughput.
- **JPEG XL** lossless offers a more balanced profile, combining reasonable compression ratios with very high compression speed.
- **PNG** (OptiPNG) remains limited in compression efficiency and is primarily justified by compatibility rather than performance.

These observations support the conclusion that modern lossless image codecs provide meaningful advantages over traditional formats, particularly when performance and storage efficiency are both relevant.

Lossy image compression introduces a trade-off between compressed file size and computational cost, with codec parameters controlling the balance between compression efficiency and encoding performance. In this section, lossy image codecs are evaluated using natural photographic images from the Kodak and Tecnick datasets. The analysis focuses on output size and encoding throughput as a function of codec quality parameters, without considering perceptual quality metrics at this stage.

This can be better seen in Figure 9, which shows the median compressed output size across all evaluated images as a function of the codec quality parameter.

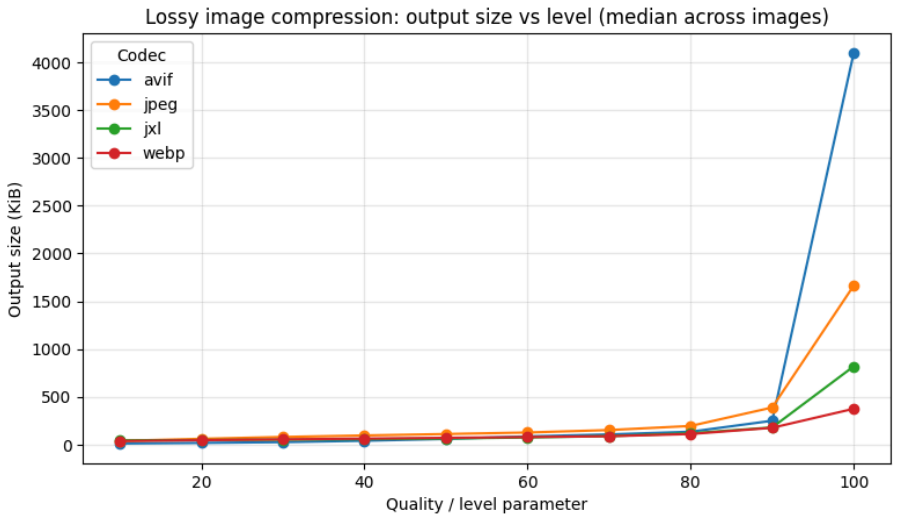

Figure 9. Lossy image compression: output size vs level

As codec quality increases, output size grows for all evaluated formats, although the rate of growth differs substantially between codecs. At moderate quality settings, all codecs produce relatively compact outputs suitable for common deployment constraints. At higher quality levels, some codecs exhibit a sharp increase in output size, indicating diminishing returns in compression efficiency when aggressive quality settings are used.

The computational impact of increasing codec quality is illustrated in Figure 10, which reports median encoding throughput as a function of the codec quality parameter.

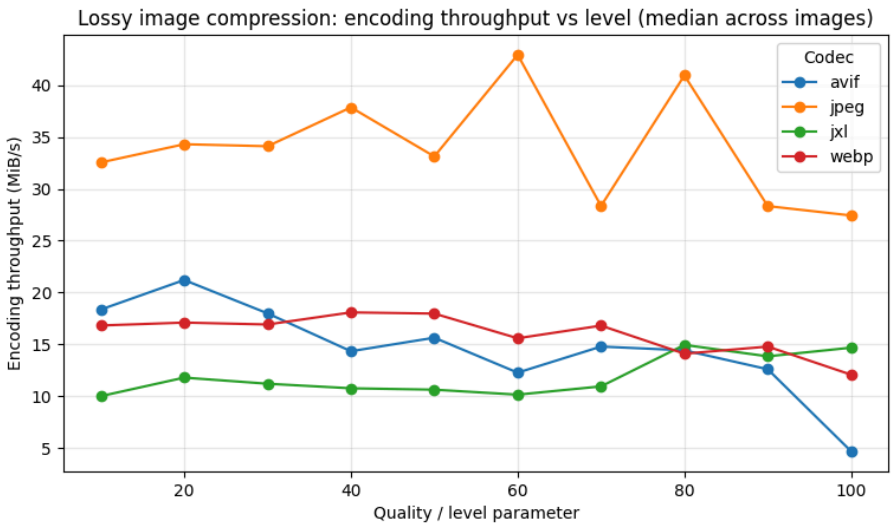

Figure 10. Lossy image compression: encoding throughput vs level

Encoding throughput decreases as quality settings increase, with the magnitude of this degradation depending strongly on the codec. Some codecs maintain relatively stable throughput across a wide range of parameters, while others experience a significant drop at higher quality levels. This behavior highlights the importance of considering encoding cost when selecting codec settings for performance-sensitive scenarios.

This can be better seen in Figure 11, which directly visualizes the trade-off between compressed output size and encoding throughput for all codecs and quality settings.

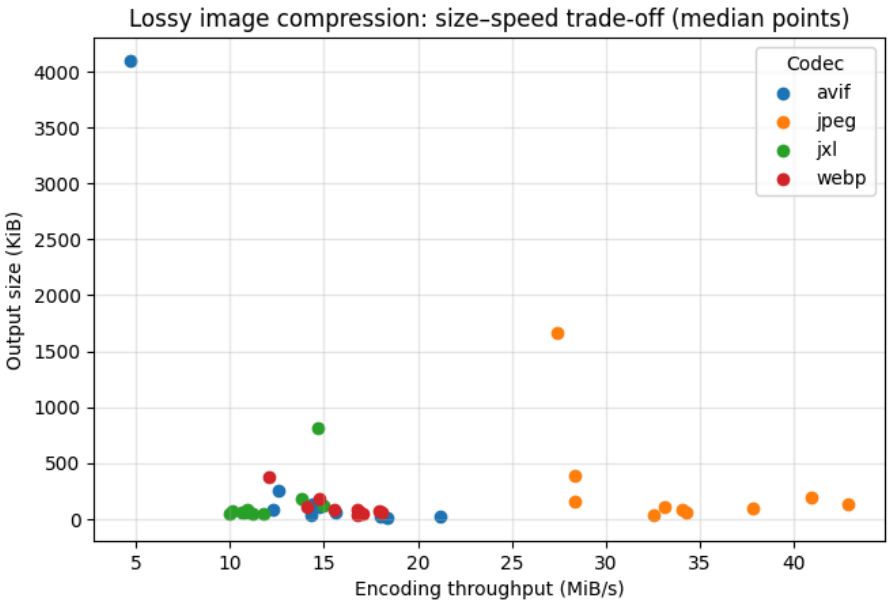

Figure 11. Lossy image compression: size–speed trade-off

Each point in Figure 11 represents a codec at a specific quality parameter setting. Lower values correspond to smaller output sizes, while higher values indicate faster encoding. The plot shows that no single codec dominates the trade-off space across all operating points. Instead, each codec occupies a distinct region, reflecting different design priorities between compression efficiency and computational performance.

Codec quality parameters are not standardized across formats, and identical numeric values do not imply equivalent visual quality. Consequently, the results in this section should be interpreted as illustrating codec behavior trends rather than direct perceptual equivalence. A quality-based comparison using perceptual metrics is addressed in a subsequent section.

10. Compatibility, Licensing, and Interoperability

While compression efficiency and performance are critical technical factors, real-world adoption is often constrained by compatibility, licensing, and interoperability considerations. A codec that offers superior compression may still be unsuitable if it lacks widespread support across operating systems, browsers, or tooling ecosystems.

For general-purpose file compression, legacy formats such as Deflate benefit from near-universal support across platforms, programming languages, and archival tools. This makes them a safe choice for interoperability, particularly when data must be exchanged between heterogeneous systems. However, this broad compatibility comes at the cost of lower compression efficiency compared to modern alternatives.

Modern codecs such as Zstandard and Brotli have achieved significant adoption in recent years. Zstandard is supported natively or via libraries in many operating systems, databases, and package managers, making it increasingly viable as a default compression format for general data. Brotli enjoys particularly strong support in web environments, where it is widely used as an HTTP content encoding and supported by all major browsers. Both codecs are released under permissive open-source licenses, facilitating integration in commercial and open-source projects.

LZMA, while offering strong compression ratios, presents interoperability challenges due to its higher resource requirements and slower performance. Its use is typically limited to archival scenarios or controlled environments where decoding cost is acceptable.

In the image domain, compatibility considerations are even more prominent. JPEG and PNG remain universally supported across browsers, operating systems, and image processing tools, which explains their continued dominance despite technical limitations. WebP and AVIF have reached broad but not universal browser support, generally exceeding 90% coverage, making them viable for web deployment when appropriate fallback strategies are implemented.

JPEG XL presents a more complex situation. Although it offers advanced technical features and strong compression performance, its adoption has been inconsistent, particularly in browser environments. As a result, JPEG XL is currently better suited for experimental, archival, or controlled deployments rather than as a primary web format.

Licensing and patent considerations also influence codec selection. All codecs evaluated in this project are available under open or royalty-free licenses suitable for research and practical use. However, historical concerns around patent

encumbrance in multimedia codecs highlight the importance of considering long-term legal clarity, especially for widely distributed applications.

To account for these factors, this project includes compatibility tables and deployment notes, emphasizing fallback strategies and hybrid approaches when a preferred codec is not universally supported.

11. Conclusions

This project has presented a systematic and reproducible evaluation of file and image compression techniques, focusing on practical trade-offs rather than theoretical optimality. By analyzing a representative set of widely used codecs across diverse datasets and realistic constraints, the work provides empirical evidence to support informed codec selection.

The results highlight the coexistence of legacy and modern compression formats, each occupying a distinct niche shaped by performance characteristics, resource requirements, and compatibility considerations. Modern codecs demonstrate clear advantages in many scenarios, but their adoption must be evaluated in the context of deployment constraints and interoperability requirements.

Beyond the immediate findings, this project establishes a reusable experimental framework that can be extended to future codecs, datasets, or evaluation metrics. Potential directions for future work include the evaluation of emerging compression standards, deeper analysis of energy consumption, and expanded perceptual studies involving human observers for image quality assessment.

Additionally, domain-specific compression techniques and adaptive pipelines that combine preprocessing, dictionary training, and codec selection represent promising areas for further investigation. As data volumes and performance demands continue to grow, such approaches are likely to play an increasingly important role in efficient data handling.

12. References

Bibliography and web resources:

[1] *RFC 1951 DEFLATE Compressed Data Format Specification version 1.3* [online]. P. Deutsch, 26 November 2021.
<https://www.rfc-editor.org/info/rfc1951> [accessed: 20 October 2025].

[2] *Zstandard* [online]. Meta Platforms, Facebook.
<https://facebook.github.io/zstd> [accessed: 20 October 2025].

[3] *Brotli Compressed Data Format* [online]. J. Alakuijala and Z. Szabadka, Google, July 2016.
<https://datatracker.ietf.org/doc/html/rfc7932> [accessed: 20 October 2025].

[4] *The .xz File Format* [online]. Tukaani.org, 8 April 2008.
<https://tukaani.org/xz/xz-file-format.txt> [accessed: 20 October 2025].

[5] *Overview of JPEG 1* [online]. JPEG.
<https://jpeg.org/jpeg> [accessed: 20 October 2025].

[6] *PNG (Portable Network Graphics) Specification Version 1.0* [online]. T. Boutell, Boutell.Com, March 1997.
<https://www.rfc-editor.org/rfc/rfc2083.html> [accessed: 20 October 2025].

[7] *An image format for the Web* [online]. Google, 7 August 2025.
<https://developers.google.com/speed/webp> [accessed: 20 October 2025].

[8] *What is AVIF?* [online]. Alliance for Open Media.
<https://aomedia.org/specifications/avif> [accessed: 20 October 2025].

[9] *Overview of JPEG XL* [online]. JPEG.
<https://jpeg.org/jpegxl> [accessed: 20 October 2025].

[10] *Silesia compression corpus* [online]. Sebastian Deorowicz, Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, Department of Algorithmics and Software.
<https://sun.aei.polsl.pl/~sdeor/index.php?page=silesia> [accessed: 3 November 2025].

[11] *Descriptions of the corpora* [online]. Matt Powell, 8 January 2001.
<https://corpus.canterbury.ac.nz/descriptions/#cantrbry> [accessed: 3 November 2025].

[12] *Kodak dataset* [online]. Sheryl, kaggle, 20 November 2018.

<https://www.kaggle.com/datasets/sherylmehta/kodak-dataset> [accessed: 3 November 2025]