Journal of Engineering and Sustainable Development
Vol. 30, No. 05, August 2026
https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd
ISSN 2520-0917
Research Article
https://doi.org/10.31272/jeasd.2643

Efficient Image Processing Through Compressive Sampling: Bridging the Gap between Data Compression and High-Quality Reconstruction

Ghassan Ahmad Ismaeel[1*]

Department of Clinical Laboratory Sciences, College of Pharmacy, University of Mosul

*Email: ghassanahmad@uomosul.edu.iq

Article Info

Received	**01/05/2024**
Revised	**22/08/2026**
Accepted	**22/08/2026**

Abstract

One of the longstanding problems in digital imaging is the compression of images with sufficient fidelity for reconstruction. Common file formats like JPEG and PNG are not well suited for higher compression, so there has been a growing interest in compressive sampling as an alternative. In this paper, the basic principles of compressive sampling are appraised, and its performance is compared with traditional methods of compression on a set of 129 animal images. The compression ratios of the Sparse Transform method on the test dataset were within the range of 9.26–93.44 (mean 45.02). Mean reconstruction error was 24.14, PSNR ranged from 18.98 to 27.07 dB (mean 23.27 dB), and SSIM from 0.68 to 0.96 (mean 0.85). The results presented here indicate that, among the compressive sampling variants tested, the Sparse Transform approach rose as the best combination of compression and reconstruction quality, whereas the conventional methods (including JPEG) have better reconstruction fidelity at each operating point. Finally, the paper raises a number of questions that need to be addressed in order to continue work in this area, particularly the applicability of the method to other more specialized imaging tasks.

Keywords: Compressive Sampling, Image Compression, Image Fidelity, Image Reconstruction, Sparse Transform Compression

1. Introduction

1.1 Background

Whether in medical diagnostics, satellite remote sensing, or computer vision, image analysis plays a vital role in a wide range of applications, each with its own requirements for the storage, transmission, and recovery of images. It therefore has a practical requirement in all these areas to efficiently retrieve and store that information – by compression [1]. At a moderate compression level, JPEG and PNG do a decent job of this, but both suffer significantly from compression at high compression ratios [2].

The drawbacks of conventional compression algorithms are well known for high compression ratios, and compressive sampling (CS) is proposed as one solution to these drawbacks. Classical signal acquisition requires sampling at the Nyquist rate [3] [4]; compressive sampling departs from this by exploiting signal sparsity to reconstruct images from far fewer measurements. This may result in lower storage and transmission costs with a corresponding decrease in quality [5].

This paper explores the possibility that compressive sampling can provide a viable alternative operating point (one that provides higher compression ratios than traditional formats, with a cost of performance in the reconstruction).

In this paper, we compare the five compressive sampling methods to one another and to three other conventional codecs, and discuss when they are practical for applications such as medical imaging, multimedia delivery, and remote sensing [6].

The precision versus compression ratio trade-off is not easy, and it has different levels of importance for different domains [7]. Medical imaging, for instance, can be adversely impacted by information loss resulting from compression [8]. In remote sensing, degradation may mask important features that are needed for scientific interpretation [9]. In these contexts, it is crucial to understand exactly what is lost and what stays in compression methods.

While JPEG and PNG are popular, they have an inherent problem: JPEG sees fine image detail lost with more aggressive compression [10]. Compressive sampling is thus different, however, such that it saves fewer samples during acquisition, but then exploits the signal's sparsity to recover it from the reduced samples [11].

In this paper, the author explores the potential of the compressive sampling operating point as an alternative to conventional formats, and what it costs in terms of reconstruction quality for the increased compression ratio.

1.2 Compressive Sampling Overview

Unlike the classical sampling theory, which requires sampling at the Nyquist rate, compressive sampling, also called compressed sensing, samples only as much as is necessary to recover a sparse signal. Since then, it has been used extensively in the fields of signal acquisition and, in particular, has been used in MRI acceleration, in radar imaging, and in lossy data compression [12].

This is because the vast majority of natural signals, such as images, are sparse in some transform domain, including, in the case of images, the coefficients capturing most of the energy (energy is the feature we are trying to compress) are relatively few [13]. The design of the measurement process can take advantage of this structure, so that the overall signal can be reconstructed using much less information than a simple measurement of the dimensions of the signal would indicate. This work belongs to three types of reconstruction algorithms: Basis Pursuit, which computes a solution that has the smallest $\ell 1$ norm of its coefficients while satisfying the measurements [14], and ISTA and FISTA, which achieve the same goal through iterative shrinkage-threshholding steps [15]; and greedy methods such as OMP and CoSaMP [16] [17] that iteratively select the most informative basis element at each step. The differences among these three families lie mainly in their speed vs. accuracy trade-offs, and their assumptions about signal structure.

In formal compressive sampling, a signal x is acquired through a linear measurement $y = \Phi x$; the sensing matrix Φ is designed so that x can be recovered from much fewer measurements than its full dimension [18]. Typically, $\ell 1$ minimization (also known as Basis Pursuit) is used to solve this problem [19] [20], and the sensing matrix is often designed to be a random matrix, as they meet the theoretical requirements for reliable recovery over a broad range of sparse signals.

There are many applications of compressive sampling in different areas, such as medical imaging and computed tomography, among other areas [21]. In image compression, its practical applications have been steadily expanding due to improvements in reconstruction algorithms, but the question of computational cost and quality of reconstruction is still open.

1.3 Comprehensive Survey of Related Work

In recent years, image compression has shifted its focus to deep learning, with CNN-based encoder and autoencoder architectures proving to be very effective on perceptual quality metrics. In addition to these architectural improvements, attention is also being paid to perceptual quality measurements that are designed to be more in line with human visual judgment than pixel error. Despite the potential gains, the deep learning techniques that have led to much of this progress do have real-world cost: they require large and diverse training sets, are slow to deploy on limited hardware, and frequently have poor performance when deployed on images that are not representative of their training distribution. A general-purpose, efficient, and reliable compression method is yet to be solved for a variety of image types.

The JPEG image format is by far the most widely used of the traditional ones. It does this by converting blocks of images from the spatial domain to frequencies in the DCT and discarding the high frequencies that are less seen by a human visual system. This data is further reduced when quantization and entropy coding are applied, providing a reasonable level of compression while maintaining a reasonable image quality. Its major drawback is the appearance of blocking artifacts, which are especially noticeable for high compression ratios [22]. PNG, on the other hand, is lossless, meaning that all the pixels are preserved by filtering and entropy coding. PNG is hence ideal for graphics, diagrams, and photos where lossy data is not important. The disadvantage is that the files are significantly larger than JPEG [23].

Wavelet-based compression is different as it decomposes an image into sub-bands of its frequencies and discards less significant sub-bands of the image by means of the Discrete Wavelet Transform. It is more demanding in terms of computation, but the technique is more effective than block-DCT-based techniques for fine detail, and may introduce ringing artifacts in high-frequency regions [24]. The latter, known as sparse coding and compressive sensing, are not based on fixed blocks of pixels but consider images as linear combinations of a few basis functions. Sparse coding means to recognize these basis functions from the data itself, whereas compressive sensing does more; it reduces the number of measurements and then reconstructs the image from sparse measurements. Both can attain even more effective compression than block-DCT methods for naturally sparse images in the domain chosen, but both fail badly for non-naturally sparse images, and both are not cost-effective in terms of computational costs. [25].

CNN-based compression models learn nonlinear image representations, which have performed better in perceptual metrics, especially at low bitrates, compared to DCT-based compression models. But they need more diverse training sets, take too long to deploy on limited devices, and are not always able to generalize across image types [26].

Each of these methods—JPEG, PNG, wavelet-based methods, and deep learning—has real trade-offs. Fast and widely used, JPEG and PNG create blocking artifacts for high compression levels. Deep Learning approaches come with better perceptual results but are more compute-intensive and need large training sets. Compressive sampling is located at the other end of this design space: it is an unsupervised approach, but its reconstruction costs and quality rely heavily on the sparsity of images.

1.4 Research Objectives and Scope

In medicine, remote sensing, and all consumer applications, the sheer number of digital images produced has increased significantly, and their storage and transmission requirements have thus become a topical, practical issue: how to compress images. Some formats are not necessarily a balance of reduced file size and loss of quality, especially when high compression ratios are required.

The aim of this paper is to consider five compressive sampling techniques and compare them with three conventional codecs on a set of 129 images and to determine which compressive sampling variant best balances compression ratio and reconstruction quality and where it is reasonably viable to use the variant.

The contributions of this study are as follows:

1. **Compression-Quality Trade-off Analysis**: This study compares the ability of compressive sampling techniques to address the compression-quality trade-off with quantitative metrics over a diverse set of images**.**
2. **Empirical Comparison:** For 129 images experimented on, the five compressive sampling variants are compared to select the one that achieves the best compression rate and good reconstruction results**.**
3. **Practical Guidance:** This study compares methods on the same dataset with the same condition, thereby determining their operating range, with one being or not being a practical choice.

The experimental setup, the dataset, and the evaluation metrics are described in Section 2. The results from the compressive sampling and traditional approaches are presented in Section 3. Conclusions are drawn in Section 5, and results are discussed in Section 4.

2. Materials and Methods

The experimental setup, image dataset, choices of compression methods, and comparison metrics are explained in this section. The entire process is illustrated in the block diagram in Fig. 1.

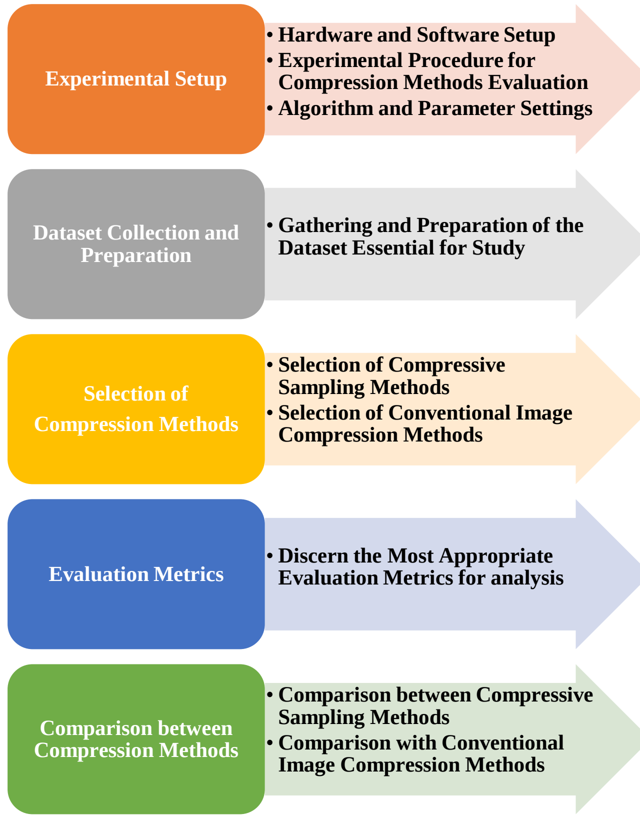


Figure 1. Methodology overview in block diagram format

2.1 Experimental Setup

2.1.1 Hardware and Software Setup

The tests were performed on a desktop computer with an Intel(R) Core(TM) i7-11800H processor, 16.0 GB of RAM, and Windows 11 Pro (version 23H2, OS build 22631.3296). All wavelet decomposition and reconstructions were performed on the 129-image dataset on the 8 physical cores of the Core i7-11800H without page or memory slowdowns. During the experiment, the OS build (22631.3296) was kept stable so as not to introduce variability due to background OS updates.

2.1.2 Experimental Procedure for Compression Methods Evaluation

OpenCV was used to read the images from disk sequentially. Each image was then fed into the compression methods being tested and then reconstructed, yielding one original image and one reconstructed image for use in computing the metric. In the case of compressive sampling methods, this involved applying the sensing matrix and then executing the algorithm selected for the reconstruction; for JPEG, WebP, and JPEG2000, the standard codec pipelines take care of it. The parameters were varied during each run to define the image quality – compression ratio relationship.

The main processing is done using three Python libraries: OpenCV for reading the images and converting them to grayscale, PyWavelets for computing the DWT and its inverse, and scikit-image for checking the results. Four metrics were calculated for each original-reconstructed pair: compression ratios, mean reconstruction error, PSNR, and SSIM. The first two were used to show the amount of data lost and how much distortion was caused; PSNR is a log scale of the distortion with respect to the peak pixel value; and SSIM measures more than just pixel difference—the degree to which the structural content of the image—edges, textures, local contrast—was preserved. For all of the images, metric measurements were taken, and the original and reconstructed image pairs were also checked visually for artifacts not measured by the metric.

2.1.3 Algorithm and Parameter Settings

In this study, the Sparse Transform method is used, in which Haar wavelet thresholding is performed. All the images were preprocessed by converting them to gray-level images and downscaling them, which greatly decreases the computational cost while keeping only the information needed for compression evaluation. The DWT then splits the grayscale image into an approximation sub-band (cA) and three details sub-bands (cH, cV, cD). Then, a sparsity criterion setting the threshold to 10% of the maximum coefficient value was imposed in the wavelet domain. This threshold was reduced to 50 for darker images (mean pixel value < 100) to avoid over-suppression of meaningful low-energy detail. Then, the reconstructed image was obtained by using the thresholded coefficients in the inverse DWT. The important parameters of the method are the type of wavelet (Haar), the values of the threshold (10% of the maximum coefficient), and the factor of adjustment for the dark images.

2.2 Dataset Collection and Preparation

The dataset was taken from Kaggle and contains 129 photos of animal subjects with a variety of species. A wildlife dataset was selected because it contains naturally occurring variations in textures, color distributions, contrast, and scene complexity, which are crucial for stress-testing compression algorithms; a collection of very similar pictures would provide an unrealistic range of compression algorithm performance metrics. File sizes ranged from 6 KB to 778 KB, and image dimensions from 100×100 to 1920×1080 pixels. Both types of file (JPG and PNG) were represented, and the evaluation was able to examine both lossy and lossless source formats and assess whether the compression methods differed in behavior with source file type. Each image was converted to grayscale prior to compression so that dimensionality is reduced, but the structural features are still relevant to the assessment of the image compression.

A few examples of images from the data set are displayed in Fig. 2.

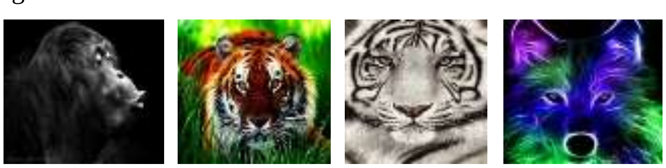


Figure 2. Random selection from animal dataset

2.3 Selection of Compression Methods

2.3.1 Selection of Compressive Sampling Methods

Five compressive sampling variants were chosen to test: Sparse Transform, Block-Based, Subsampling, Binary Sensing, and Structured Sampling. They were selected since they are particular methods for acquiring and reconstructing the signal and are frequently used in compressive sensing literature [27] [28].

Sparse Transform was chosen because it takes advantage of the sparsity of the wavelet and Fourier representations, which proves to be the ideal choice for images in which energy is focused in a few coefficients of the transform. Block-Based is the method for dividing the image into patches and compressing each block separately, which is suitable for localized compression instead of global compression.

Using subsampling to lower acquisition cost is done by a simple subsampling of the pixel data. The use of binary measurement matrices in Binary Sensing reduces the hardware complexity. Structured Sampling uses deterministic sensing matrices, based on knowledge of the signal structure, which may yield better reconstruction in some instances.

The methods were deliberately included within various acquisition strategies to ensure that the evaluations will identify not only which method was best in general and where each method's approach will be beneficial or disadvantageous.

2.3.2 Selection of Conventional Image Compression Methods

The conventional codecs selected to be used as benchmarks are: JPEG [29], WebP Compression [30], and JPEG2000 Compression [31]. JPEG is still the most popular lossy standard. Google created WebP as an alternative that is best optimized for the web. JPEG2000 provides scalable and lossless compression modes, and was employed in medical and archival imaging.

This is because the three codecs provided a solid performance reference that the compressive sampling results might be directly compared to on a like-for-like basis.

2.4 Evaluation Metrics

Throughout, four metrics were used. Compression ratio is the original file size divided by the compressed file size; ratios greater than 1.0 are compression, ratios less than 1.0 are expansions.

Reconstruction error is the mean absolute difference between the original image and the recovered image, which in this case is directly related to the amount of lost information [32].

The same can be presented in decibels relative to the maximum pixel value, and this is known as PSNR [33]. The reasons for introducing SSIM are that PSNR is not always correlated when human judgment is made in some situations. SSIM compares luminance, contrast, and local structure between two images, and it is known to be more closely correlated with human judgment [34]. These four values, applied together, provide a reasonably complete description of both the compression and the degradation of the image, which occur with the method.

2.5 Comparison between Compression Methods

2.5.1 Comparison between Compressive Sampling Methods

A comparison of Compressive Sampling Methods is undertaken. The first comparison was of the many compressive sampling variants themselves to find which performs well on the test dataset with regard to compression ratio vs. reconstruction quality.

2.5.2 Comparison with Conventional Image Compression Methods

The method that achieved the highest performance was then compared directly with JPEG, WebP, and JPEG2000 on all 129 images. The goal is to describe how the quality of reconstruction varies with the compression ratio of the Sparse Transform compared with other codecs, under its own settings.

3. Results

3.1 Results of Compressive Sampling Methods

Table 1 shows the compression ratio, reconstruction error, PSNR, and SSIM for every method for all 129 test images. The distribution of results is broad, and not all methods perform as compression techniques should: some generated outputs are larger than the inputs (this is discussed further below):

- **Sparse Transform Compressive Method:** The sparsity characteristics of each individual image had the greatest impact on the outputs of the Sparse Transform method, as evidenced by compression ratios ranging from 9.26 to 93.44 with a mean of 45.02 — the widest range and highest average of the five methods. Reconstruction errors were between 13.70 and 32.68, with an average of 24.14. PSNR ranged from 18.98 to 27.07 dB, with a mean of 23.27 dB. SSIM values ranged from 0.68 to 0.96, averaging 0.85.
- **Block-Based Compressive Sampling Method:** The Block-Based method resulted in a very small compression ratio of 1.0024 to 1.0628 (mean 1.0257), which means that the file sizes were only decreased by 2–6%. The number in the reconstruction errors (5.0655) was the lowest among the five methods, but this is at the expense of achieving almost no compression gain.
- **Subsampling Compressive Sampling Method:** The ratio of the compressed output to the original was only 0.25-0.2523 (average 0.2508) within the Subsampling Compressive Sampling Method. This method has the highest reconstruction errors (68.66) and poor results on both metrics in this set.
- **Binary Sensing Compressive Sampling Method:** The results of the Binary Sensing Compressive Sampling Method are compression ratios close to 0.5, indicating double the file size, and the highest reconstruction errors of all five methods (mean of ~102.48 and ~99.61, respectively).
- **Structured Sampling Compressive Sampling Method:** In this method, compression ratio is about 0.5 were obtained, that is in the case is similar to Binary Sensing. The average reconstruction error was approximately 99.61. PSNR and SSIM are similar to those of other methods. Consequently, neither would be a practical compression method under these conditions. Table 1 presents the full numerical comparison.

Table 1. Performance comparison of compressive sampling methods

Method	Compression Ratio (Range)	Compression Ratio (Mean)	Reconstruction Error (Range)	Reconstruction Error (Mean)
Sparse method	9.26 - 93.44	45.02	13.70 - 32.68	24.14
Block-Based method	1.0024 - 1.0628	1.0257	0.0161 - 33.8765	5.0655
Subsampling method	0.25 - 0.2523	0.2508	11.73 - 93.41	68.66
Binary Sensing method	Close to 0.5	Close to 0.5	Varies	~102.48
Structured method	Close to 0.5	Close to 0.5	Varies	~99.61

Fig. 3 presents the distributions of compression ratios and reconstruction errors for the five methods as bars, demonstrating the significant performance of the Sparse Transform method relative to the other four methods.

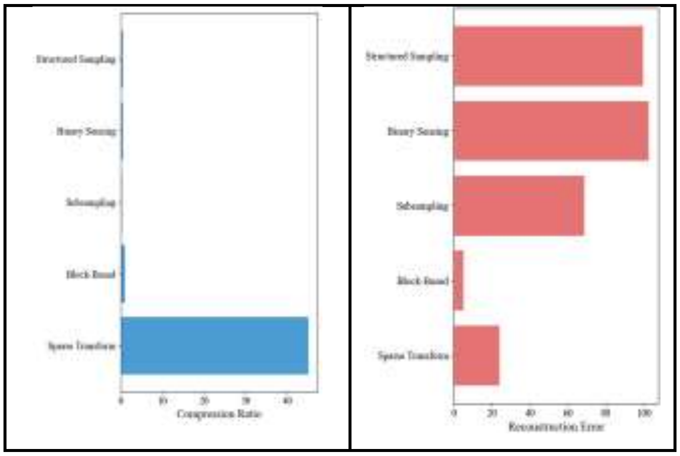


Figure 3. Compression ratios and reconstruction errors across sampling methods

3.1.1 Conclusion on Compression Sampling Methods

The results clearly substantiate one of the five variants tested: the Sparse Transform method. Thus, only one of them has a significant compression ratio (ratios much greater than 1.0 in most images) without high reconstruction errors. The Block-Based method yielded cleaner reconstructions, but it does not compress the data much. The other three methods (i.e., Subsampling, Binary Sensing, and Structured Sampling) grow instead of shrinking the data of the parameter settings employed here, and will be impractical without further tuning.

The side-by-side comparison of original and Sparse Transform-reconstructed images is displayed in Fig. 4, such that the main structural features are preserved although some of the fine texture details are lost, especially in darker areas of the images.

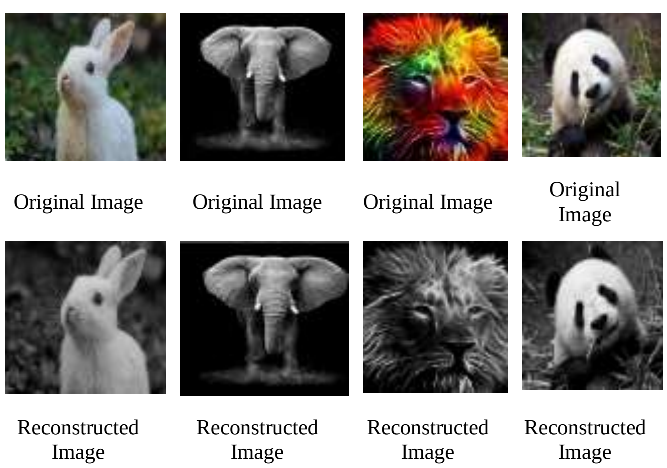


Figure 4. Sparse Transform Method - Image Reconstruction Comparison

3.1.2 T-Tests Statistical Analysis of Compressive Sampling Methods

This was performed using two-tailed t-tests [35] to determine whether the performance differences between Sparse Transform and the four methods were statistically significant or may be due to chance.

None of the comparisons for compression ratio were significant (all $p > 0.05$), so the differences in compression ratio between the methods cannot be confidently attributed to the method on this data size.

The situation was different for the reconstruction quality. The reconstruction errors of Sparse Transform are also significantly lower than those of Binary Sensing and Structured Sampling ($p < 0.05$), and significantly higher than those of both, respectively.

Yet, no significant differences were found by SSIM between any method pair, as all five methods performed in very similar ways when comparing their structural similarity scores to the originals despite the large differences in raw pixel error. These discrepancies in the PSNR and SSIM results should be explored more to see if they have more differences in low-level noise than in the cases of perceptible structure deterioration. The results from the conventional compression methods are as follows. Table 2 shows the compression ratio, reconstruction error, PSNR, and SSIM for JPEG, WebP, and JPEG2000 on the same 129-image dataset on which the CS method comparison was done.

Table 2. Summary of T-Test statistical analysis for compressive sampling methods

Aspect	Comparison	T-test Results	Interpretation
Compression Ratio	Sparse Transform vs. Other Methods	No significant differences ($p > 0.05$)	Sparse Transform method shows comparable compression ratios to other methods.
Reconstruction Error	Sparse Transform vs. Other Methods	Significant differences ($p < 0.05$)	Sparse Transform exhibits the lowest reconstruction errors when compared to Binary Sensing and Structured Sampling methods.
Peak Signal-to-Noise Ratio	Sparse Transform vs. Other Methods	Significant differences ($p < 0.05$)	Sparse Transform demonstrates the highest PSNR value compared to Binary Sensing and Structured Sampling methods.
Structural Similarity Index	Sparse Transform vs. Other Methods	No significant differences ($p > 0.05$)	All methods maintain a similar level of structural similarities between the original and reconstructed image.

3.2 Results of Conventional Compression Methods

Table 3 reports compression ratio, reconstruction error, PSNR, and SSIM for JPEG, WebP, and JPEG2000, measured on the same 129-image dataset used for the CS method comparisons.

3.2.1 Compression Metrics Overview

- **JPEG Compression:**

Compression ratios obtained from the JPEG compression technique were between 0.1827 and 0.5248, with an average compression ratio of 0.3273. Reconstruction error values spanned from 90.3216 to 1195.311, with an average of 279.5624. PSNR values ranged between 48.6282 and 57.7396 dB, with an average of 53.0054 dB. The JPEG SSIM values are also consistently high (0.9945–0.9996, mean 0.9971) even though the absolute pixel-level errors were larger, consistent with the design of JPEG, which emphasizes those frequency components that are relevant to the human perception of the images.

- **WebP Compression:**

WebP produced compression ratios of 2.33–12.20 (mean 7.12), reconstruction errors of 58.24–121.40 (mean 102.03), PSNR of 31.64–39.25 dB (mean 35.33 dB), and SSIM of 0.924–0.979 (mean 0.947). It compresses more aggressively compared to JPEG at the settings used, although still maintaining SSIM values over 0.92 on every test image, which broadly confirms its suitability for web delivery applications where moderate quality at highest compression is the objective.

- **JPEG2000 Compression:**

JPEG2000 returned compression ratios of 0.00981–0.01019 (mean ~0.01). This is an incredibly low ratio; it's typically a sign of an encoder parameter error and is unlikely to be a true characteristic of the algorithm. The error obtained in the reconstruction process was very low (mean of 0.71), and the PSNR varied from 18.27 to 33.03 dB (mean of 26.78 dB).

3.2.2 Comparison with Sparse Transform Compressive Sampling

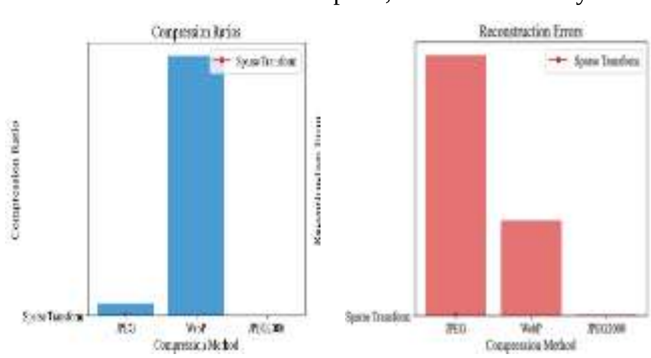
On the same 129-image dataset, the Sparse Transform method produced compression ratios of 9.26–93.44 (mean 45.02), reconstruction errors of 13.70–32.68 (mean 24.14), PSNR of 18.98–27.07 dB (mean 23.27 dB), and SSIM of 0.68–0.96 (mean 0.85).

Table 3. Performance Comparison of Conventional Compression Methods

Compression Method	Compression Ratio (Range)	Compression Ratio (Mean)	Reconstruction Error (Range)	Reconstruction Error (Mean)
JPEG	0.1827 - 0.5248	0.3273	90.3216 - 1195.311	279.5624
WebP	2.3265 - 12.2034	7.1191	58.2403 - 121.3966	102.0263
JPEG2000	0.009810 - 0.010185	0.009996	0.495612 - 0.860294	0.711915

In order to have a general eye view of how Sparse Transform compares to the three conventional codecs in each of the metrics, all four methods are plotted together in Fig. 5.

The relative compression ratios, reconstructed errors, PSNR, and SSIM are plotted in Fig. 5 for JPEG, WebP, JPEG200 and 2000 compression techniques. All the measurements are shown in the subplots, and the horizontal axis of each subplot corresponds to the conventional compression techniques. Of course, the Sparse Transform Compressive Sampling method is marked in red in each of the subplots, for reference only.



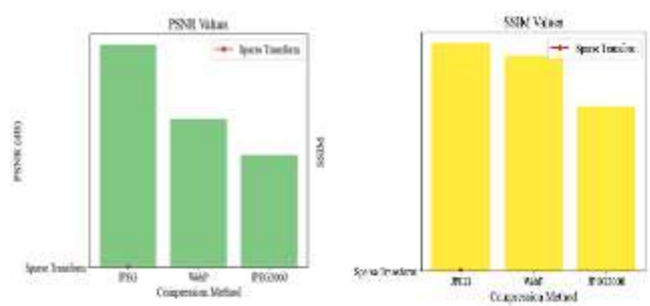


Figure 5. Comparative analysis of compression metrics, with sparse transform compressive sampling

4. Discussion

4.1 Advantages and Limitations of Sparse Transform Compressive Sampling

In this section, we discuss the advantages and limitations of Sparse Transform Compressive Sampling.

- **Theoretical Foundation:** STCS is based on the fact that natural images are sparse in the wavelet and other transforms, with most wavelet coefficients being close to zero and a few with most of the energy. This sparsity could be exploited to reconstruct the image within significantly fewer measurements than the number of pixels in the image, which is, in this case, the fundamental process that makes compression possible.

- **Practical Benefits**: Compression ratios achieved by STCS in this study are up to 93× on individual images, which is an indication of its capability to discard a large proportion of wavelet coefficients while retaining the dominant structure of images. This is applicable to any situation where storage or transmitting capacity is a limit. It has a main practical disadvantage, and that is its computations: iterative algorithms like sparse reconstruction are much slower than a JPEG encode-decode cycle, which makes it unfeasible for real-time applications.

- **Practical Implications:** The method as implemented currently is only used for grayscale images, and therefore cannot be directly used to perform tasks dependent on its color, like medical dermoscopy or satellite multispectral imaging, without adapting it further.

- **Real World Validation:** There was a reasonable degree of consistency across the 129 test images, yet the method seemed to be more or less effective on darker images and for images with fine, high-frequency texture, such as animal fur, feather detail, and complex backgrounds. The Haar wavelet's poor representation of the high-frequency structure resulted in loss of meaningful information during thresholding and in a visually “noisy” reconstruction in these cases. This is of more practical importance in other applications such as forensic imaging or medical dermatology, where its very fine surface detail is of diagnostic value.

- **Comparative Examination**: The main difference between STCS and JPEG or PNG is the location of sparsity utilized. JPEG operates block-by-block and discards high-frequency coefficients, whereas STCS works globally in the wavelet domain, potentially leading to better capture of long-range structure in the sparse image case. On the other hand, JPEG performs better compared to STCS in reconstructing the image on textured images or images with high frequency (meaning the sparsity assumption fails), as the experimental results of this study demonstrated.

- **Relevance to Research Goals**: The goal of characterizing the compression-quality trade-off provided by compressive sampling was achieved. Of the five CS variants, the Sparse Transform yielded the best result, yet the results also showed clearly that a higher compression ratio is at a tangible price in terms of reconstruction fidelity when compared to the traditional codecs. That is a valuable contribution in itself, because it provides the conditions below which STCS is and is not a viable option, which is what the research was designed to determine.

4.2 Comparative Analysis and Limitations of Compression Methods

A comparison between STCS and traditional image compression like JPEG, WebP, and JPEG 2000 was made based on the compression ratio, the reconstruction accuracy, PSNR, and SSIM. The specific goal is to understand the compression ratio and reconstruction quality of STCS compared to conventional codecs, and where STCS is a viable option. Comparing the compression ratios of the five compressive sampling methods, the t-test results in no statistically significant difference among the various methods (all $p > 0.05$), indicating that there are no reliable differences in compression ratio between the methods on this dataset. Both Binary Sensing and Structured Sampling had substantially larger mean errors, while STCS had significantly lower reconstruction error. The results of PSNR were the same; there were significantly higher PSNR values for Sparse Transform compared to Binary Sensing and Structured Sampling ($p < 0.05$), and no significant PSNR and SSIM difference was observed between Sparse Transform and Block-Based or Subsampling methods.

There are also definite limitations of STCS. Reconstruction algorithms such as Basis Pursuit and OMP are iterative and far from cheap, meaning they are hard to implement in real time on low-power equipment. It also fails on images with dense local detail, high contrast variation, or complex texture, which is when the assumption of sparsity also fails. For applications such as digital pathology or high-fidelity artwork reproduction, this is a meaningful limit due to the need for fine texture.

The selection of the transform/recover algorithm also affects performance, and tuning the parameters for the best performance will increase the implementation cost. There is a further practical concern with integrating STCS into the current imaging pipelines: the additional overhead of the pre-processing steps that are required for STCS, compared to simpler codecs; the need to process data in the presence of transmission channel noise; and the necessity to correct errors in the data.

These limitations notwithstanding, the results demonstrate that STCS has a unique and potentially valuable place in the compression design space: it achieves high compression ratios not offered by traditional codecs, but with lower reconstruction fidelity. The proportion might be acceptable for some applications but not for others, and the information in this study provides a solid foundation for making a decision.

5. Conclusion

129 animal images were used to test five compressive sampling methods and three conventional codecs. The Sparse Transform method was the only compressive sampling variant that gives compression ratios that are always greater than 1.0 and had a mean value of 45.02 and a range of 9.26–93.44. The mean reconstruction error, mean PSNR, and mean SSIM were 24.14, 23.27 dB, and 0.85, respectively. The results of these reconstruction quality figures are validated using t-tests, which were found to be significantly higher than Binary Sensing and Structured Sampling ($p < 0.05$), yet there was no compression

ratio difference among the methods. The Block-Based method results yielded lower reconstruction error and resulted in almost no real compression. Since all three subsampling methods, Binary Sensing, and Structured Sampling increased the file size, they were not suitable below the tested conditions.

The ratio of compression figures are not directly comparable against the conventional codecs, such that each computes the ratio differently: JPEG report the ratio of compressed to original bytes (mean of 0.33, meaning that the JPEG files are lower to be compressed than the originals); WebP reports the ratio of original to compressed bytes (mean of 7.12, meaning the WebP files are more compressed than the originals); Sparse Transform report the ratio of wavelet-coefficients to pixels (mean of 45.02, meaning the Sparse Transform files are less to be compressed than the originals). The data clearly show that all three conventional codecs performed significantly better in terms of the reconstruction quality: JPEG attains a mean PSNR of 53.00 dB and a mean SSIM of 0.997, which is much higher than the PSNR of 23.27 dB and SSIM of 0.85 reached by Sparse Transform. This is not a direct JPEG substitute for applications with quality requirements, but is at a different operating point—of a much higher compression ratio, but at a lower reconstruction fidelity. If storage or the bandwidth is limited and a moderate image quality is acceptable (bulk storage of non-diagnostic imagery, for example), the trade-off can be worth it. JPEG is more suitable for high-fidelity images.

However, there are three directions which seem to be the most promising of future research: testing other wavelet bases than Haar, which are more suitable for dealing with high-frequency texture; development of adaptive thresholding rules that will take account of the local image statistics in the place of applying fixed percentage of the maximum coefficient; and extension of the method to colour images (that were not considered in this study).

Acknowledgements

The author thanks the reviewers for their careful reading and constructive comments.

Conflict of interest

The author declares no conflict of interest.

References

[1] N. A. Awad and A. Mahmoud, "Improving Reconstructed Image Quality via Hybrid Compression Techniques," *Computers, Materials & Continua*, vol. 66, no. 3, pp. 3151–3160, 2021, doi:https://doi.org/10.32604/cmc.2021.014426

[2] A. J. Hussain, A. Al-Fayadh, and N. Radi, "Image compression techniques: A survey in lossless and lossy algorithms," *Neurocomputing*, vol. 300, pp. 44-69, 2018, doi: https://doi.org/10.1016/J.NEUCOM.2018.02.094.

[3] C. E. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949, doi: https://doi.org/10.1109/jrproc.1949.232969.

[4] R. Arie, A. Brand, and S. Engelberg, "Compressive sensing and sub-Nyquist sampling," *IEEE Instrumentation & Measurement Magazine*, vol. 23, no. 2, pp. 94-101, 2020, doi: https://doi.org/10.1109/MIM.2020.9062696.

[5] M. M. Abo-Zahhad, A. I. Hussein, and A. M. Mohamed, "Compressive sensing algorithms for signal processing applications: A survey," *International Journal of Communications, Network and System Sciences*, vol. 8, no. 6, pp. 197-216, 2015, doi: https://doi.org/10.4236/ijcns.2015.86021.

[6] A. Singh and K. G. Kirar, "Review of image compression techniques," 2017 International Conference on Recent Innovations in Signal Processing and Embedded Systems (RISE), Bhopal, India, 2017, pp. 172-174, doi: https://doi.org/10.1109/RISE.2017.8378148.

[7] J. Chen, Y. Gao, C. Ma, and Y. Kuo, "Compressive sensing image reconstruction based on multiple regulation constraints," *Circuits, Systems, and Signal Processing*, vol. 36, pp. 1621-1638, 2017, doi: https://doi.org/10.1007/s00034-016-0432-2.

[8] M. Lakshminarayana and M. Sarvagya, "Algorithm to balance compression and signal quality using novel compressive sensing in medical images," in *International Conference on Information and Communication Technology for Intelligent Systems*, 2016: Springer, pp. 317-327, doi: https://doi.org/10.1007/978-3-319-33622-0_29.

[9] B. Zhang, "One-two-one networks for compression artifacts reduction in remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 184-196, 2018, doi: https://doi.org/10.1016/j.isprsjprs.2018.01.003.

[10] A. J. I. Barbhuiya, T. A. Laskar, and K. Hemachandran, "An approach for color image compression of JPEG and PNG images using DCT and DWT," in *2014 International Conference on Computational Intelligence and Communication Networks*, 2014: IEEE, pp. 129-133, doi: https://doi.org/10.1109/CICN.2014.40.

[11] S. Zhu, B. Zeng, and M. Gabbouj, "Adaptive sampling for compressed sensing based image compression," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 94-105, 2015, doi: https://doi.org/10.1016/j.jvcir.2015.03.006.

[12] S. Bennett, "Compressive sampling using a pushframe camera," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 1069-1079, 2021, doi: https://doi.org/10.1109/TCI.2021.3114980.

[13] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21-30, 2008, doi: https://doi.org/10.1109/MSP.2007.914731.

[14] S. Chen and D. Donoho, "Basis pursuit," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, 1994, vol. 1: IEEE, pp. 41-44, doi: https://doi.org/10.1109/ACSSC.1994.471413.

[15] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183-202, 2009, doi: https://doi.org/10.1137/080716542.

[16] H. S. Goklani, J. N. Sarvaiya, and A. Fahad, "Image reconstruction using orthogonal matching pursuit (OMP) algorithm," in *2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking*, 2014: IEEE, pp. 1-5, doi: https://doi.org/10.1109/ET2ECN.2014.7044960.

[17] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009, doi: https://doi.org/10.1016/j.acha.2008.07.002

[18] J. Zhang, C. Zhao, and W. Gao, "Optimization-inspired compact deep compressive sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 4, pp. 765-774, 2020, doi: https://doi.org/10.1109/JSTSP.2020.2977507.

[19] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005, doi: https://doi.org/10.1109/TIT.2005.858979.

[20] T. Saha, S. Srivastava, S. Khare, P. S. Stanimirović, and M. D. Petković, "An improved algorithm for basis pursuit problem and its applications," *Applied Mathematics and Computation*, vol. 355, pp. 385-398, 2019, doi: https://doi.org/10.1016/j.amc.2019.02.073.

[21] Z. Hu, D. Liang, D. Xia, and H. Zheng, "Compressive sampling in computed tomography: Method and application," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 748, pp. 26-32, 2014, doi: https://doi.org/10.1016/j.nima.2014.02.026.

[22] S. Dhawan, "A review of image compression and comparison of its algorithms," *International Journal of Electronics & Communication Technology*, vol. 2, no. 1, pp. 22-26, 2011. Retrieved from https://ceng460.cankaya.edu.tr/uploads/files/file/Review%20paper.pdf.

[23] K. Sayood, *Introduction to Data Compression, 5th ed. Morgan Kaufmann, 2017.*

[24] M. Weeks and M. Bayoumi, "Discrete wavelet transform: architectures, design and performance issues," *Journal of VLSI signal processing systems*

for signal, image and video technology, vol. 35, pp. 155-178, 2003, doi: https://doi.org/10.1023/A:1023648531542.

[25] M. Dadkhah, M. J. Deen, and S. Shirani, "Compressive sensing image sensors-hardware implementation," *Sensors*, vol. 13, no. 4, pp. 4961-4978, 2013, doi: https://doi.org/10.3390/s130404961.

[26] A. S. Abd-Alzhra and M. S. Al-Tamimi, "Image compression using deep learning: methods and techniques," *Iraqi Journal of Science*, pp. 1299-1312, 2022, doi: https://doi.org/10.24996/ijs.2022.63.3.34.

[27] I. A. Kougioumtzoglou, I. Petromichelakis, and A. F. Psaros, "Sparse representations and compressive sampling approaches in engineering mechanics: A review of theoretical concepts and diverse applications," *Probabilistic Engineering Mechanics*, vol. 61, p. 103082, 2020, doi: https://doi.org/10.1016/j.probengmech.2020.103082.

[28] H. Gan, Y. Gao, C. Liu, H. Chen, T. Zhang, and F. Liu, "AutoBCS: Block-based image compressive sensing with data-driven acquisition and noniterative reconstruction," *IEEE Transactions on Cybernetics*, vol. 53, no. 4, pp. 2558-2571, 2021, doi: https://doi.org/10.1109/TCYB.2021.3127657.

[29] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics, vol. 38, no. 1, pp. xviii–xxxiv, 1992*, doi: https://doi.org/10.1109/30.125072

[30] E. Öztürk and A. Mesut, "Performance evaluation of JPEG standards, Webp and PNG in terms of compression ratio and time for lossless encoding," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 2021: IEEE, pp. 15-20, doi: https://doi.org/10.1109/UBMK52708.2021.9558922.

[31] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: an overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, 2000, doi: https://doi.org/10.1109/30.920468

[32] P. Khobragade and S. Thakare, "Image compression techniques-a review," *Int. J. Comput. Sci. Inf. Technol*, vol. 5, no. 1, pp. 272-275, 2014. Retrieved from: https://www.ijcsit.com/docs/Volume%205/vol5issue01/ijcsit2014050157.pdf

[33] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008, doi: https://doi.org/10.1049/el:20080522

[34] D. R. I. M. Setiadi, "PSNR vs SSIM: imperceptibility quality assessment for image steganography," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8423-8444, 2021, doi: https://doi.org/10.1007/s11042-020-10035-z.

[35] J.-H. Xue and D. M. Titterington, "t-tests, F -tests and Otsu's Methods for Image Thresholding," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2392-2396, 2011, doi: https://doi.org/10.1109/TIP.2011.2114358.