https://doi.org/10.31891/2307-5732-2026-363-90
УДК 004.421:004.932:519.217

SOLODKA VALENTYNA
State University of Intelligent Technologies and Communications
https://orcid.org/0000-0002-5357-6682
e-mail: valyaonas@gmail.com
PATLAIENKO MYKOLA
State University of Intelligent Technologies and Communications
https://orcid.org/0000-0002-5565-8956
e-mail: nick_msa@ukr.net
TOMASHEVSKYI IVAN
State University of Intelligent Technologies and Communications
e-mail: ti.rezerv@gmail.com
HOHNYAK OLEKSANDR
State University of Intelligent Technologies and Communications
e-mail: gohnayk@gmail.com

RECURSIVE WAVELET TRANSFORM ALGORITHM FOR IMAGE PROCESSING

This study presents the design and real-world implementation of advanced recursive image compression algorithms based on wavelet transform techniques. The proposed method aims to improve overall compression efficiency while preserving high image quality and optimizing computational performance. The visual quality of reconstructed images while simultaneously reducing computational demands to accelerate processing speed. A multilevel recursive wavelet decomposition scheme is formulated to provide adaptive, scale-sensitive analysis of image features. To improve the segmentation stage, a unified quadtree-based partitioning strategy is proposed, enabling flexible subdivision of image areas according to local texture characteristics and intensity distribution. Furthermore, the study presents a parametric segmentation method that relies on ranking block intersections. Candidate domain blocks are assessed using similarity measures and ordered through a prioritization mechanism, which significantly limits unnecessary comparisons and reduces the effective search space. This optimization decreases the total number of domain blocks involved in encoding, thereby improving computational efficiency without negatively affecting compression effectiveness.The study was conducted using a dataset comprising 102 flower categories, each containing at least 40 images. The photographs were captured from multiple perspectives and under varying conditions to ensure diversity and robustness of the data. lighting conditions to ensure variability. Performance evaluation considered objective similarity indicators, encoding time, and overall processing performance. Although the compression ratios achieved by all methods were comparable, examined models were nearly equivalent, noticeable differences were observed in processing speed. Among the deep learning architectures incorporated into the preprocessing stage, MobileNet V2 demonstrated the highest computational efficiency while maintaining strong reconstruction accuracy. As an outcome of this research, a dedicated software application was created to realize the recursive compression framework built upon wavelet transform techniques. The application provides fast and reliable image data compression by integrating content-independent compression techniques with adaptive segmentation procedures, making it well suited for large-scale image processing applications.

Keywords: Wavelet transform; image processing; recursive algorithm; compression ratio; coding time

СОЛОДКА ВАЛЕНТИНА, ПАТЛАЄНКО МИКОЛА, ТОМАШЕВСЬКИЙ ІВАН, ГОГНЯК ОЛЕКСАНДР
Державний університет інтелектуальних технологій та зв'язку

РЕКУРСИВНИЙ АЛГОРИТМ ВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ

У цій показані та наведені нові та перспективні схеми - рекурсивні зменшенн зображень, в виді стиснення на вейвлет-перетворень, які є основою. Показані схеми направлені не тільки на якість різкості, що підвищує досліджувальне зображень, але й на зменшення часу переробки за рахунок скорочення кількості необхідних обчислень. Запропоновано комплексний підхід до реалізації алгоритму квадродерева, а також розроблено й продемонстровано новий параметричний метод сегментації зображень із використанням ранжованих перетинів блоків. Показано, що запропонований алгоритм зменшує кількість доменних блоків. Для експерименту було використано набір даних, що складається зі 102 наборів зображень квітів, при цьому для кожної категорії було доступно щонайменше 40 зображень, отриманих з різних ракурсів та за різних умов освітлення. Відповідно до експериментальних результатів, найбільш придатну модель було обрано на основі рівня подібності та часу кодування, оскільки коефіцієнт стиснення для всіх алгоритмів був приблизно однаковим. Модель MobileNet V2 продемонструвала найвищу швидкість обробки.В цьому отримані результати підтверджує забезпечення швидкості та ефективності стиснення даних із використанням комбінації методів стиснення, незалежних від вмісту.

Ключові слова: *вейвлет-перетворення; обробка зображень; рекурсивний алгоритм; коефіцієнт стиснення; час кодування*

Стаття надійшла до редакції / Received 18.02.2026
Прийнята до друку / Accepted 11.03.2026
Опубліковано / Published 26.03.2026

Introduction

This section presents a comprehensive approach to implementing the quadtree algorithm and introduces a new parametric image segmentation method based on ranked block intersections. The developed algorithm minimizes. The quantity of domain blocks is reduced, which consequently decreases the number of comparisons between ranked and domain blocks, thereby enhancing the compression performance of the target image.

At the same time, several issues remain unresolved.issues persist, as existing ITU compression standards struggle to deliver acceptable quality when encoding high-definition (HD) and ultra-high-definition (UHD) images, especially under real-time transmission constraints. Contemporary compression methods designed for UHD resolutions

such as 2K, 4K, and 8K place stringent demands on both processing speed and computational efficiency. These demands must be satisfied while remaining within the hardware limitations and operational frameworks imposed by modern compression standards.

Consequently, modern recursive image compression techniques are designed not only to enhance visual quality but also to shorten processing time by significantly reducing computational complexity.

Materials and Methods

The proposed compression method is implemented in accordance with the fundamental standards MPEG-2, MPEG-4 AVC, and UHDTV, as defined in ITU-R Recommendations BT.1737 and BT.1870. These recommendations outline video coding technologies for digital terrestrial broadcasting systems utilizing MPEG-2, MPEG-4 AVC, and MPEG-H HEVC formats. The principal encoding specifications and parameters are additionally clarified in ITU-R Recommendation BT.2073., which defines the essential HEVC configuration for HDTV and UHDTV broadcasting environments [1].

Wavelet-based image coding is standardized in JPEG-2000 [3], which addresses still-image compression and includes extended tools for image processing. The Motion JPEG-2000 standard [6] extends this methodology to video compression by applying wavelet coding independently to each frame in a video sequence.

Image quality evaluation in video conversion and encoding systems commonly relies with particular emphasis on the peak signal-to-noise ratio (PSNR) as a primary quality assessment metric. Furthermore, additional guidance is provided in ITU-R Recommendation BT.1683 [4] and ITU-T Recommendation J.144, which address objective methods for evaluating video quality. [5] introduce the enhanced PSNR (EPSNR) metric, which is specifically designed to assess noise behavior near object boundaries. These recommendations outline the principles for applying EPSNR, particularly in wavelet-based image processing, where it is often considered a more suitable quality indicator than conventional PSNR.

It is worth noting that integrating the intraframe coding tools of the JPEG-2000 standard [6] into video coding standards, alongside existing techniques defined in MPEG-based frameworks, could have further increased overall compression efficiency.

Wavelet-based methods underpin several compression standards developed for specialized applications. JPEG-2000 relies entirely on wavelet transforms, and numerous studies present theoretical models for spectral image coding using edge-adaptive wavelet bases. When properly implemented, these techniques demonstrate notable gains in compression efficiency, especially for high-resolution images reconstructed with minimal distortion.

At present, H.265/HEVC represents the most advanced video compression standard, incorporating decades of research in digital image and video coding. A simplified functional diagram of an HEVC encoder is shown in Figure 1.1, while its operational principles are described in detail in [2].

During encoding, reconstructed reference frames are stored in a buffer and reused for predicting subsequent frames. As a result, the sequence of frames in the compressed bitstream does not necessarily match the temporal order of the original video.

One of the key characteristics of the wavelet transform is its capacity to concentrate the signal's energy within a limited set of coefficients. Following the transformation process, the signal is decomposed into multiple subbands, where most of the energy is typically localized in the low-frequency components. (LF) subband. This characteristic makes scalar quantization particularly effective, as fewer bits are required to encode regions with low energy. However, this approach does not fully exploit the structural dependencies present in the high-frequency (HF) subbands. Strong signal components often reappear across multiple subbands because they originate from image edges. If the wavelet basis cannot adequately represent these features at higher decomposition levels, some of the energy spreads into HF coefficients.

Smooth image regions are well approximated by the LF wavelet components, leading to dense clusters of coefficients in the low-frequency band. This behavior repeats at successive decomposition levels, resulting in a noticeable similarity between subbands. Such interband relationships can be explained by prior knowledge of typical image content, which consists of smooth areas, textures, and edges.

To exploit this structure, the algorithm organizes wavelet coefficients into a spatial tree. Each tree begins at a selected root coefficient in a given subband and branches into four child coefficients located in the corresponding subband of the previous decomposition level.

Following the wavelet transform, entropy coding is applied to convert the quantized coefficient blocks into a compressed bitstream.

The SPIHT (Set Partitioning in Hierarchical Trees) algorithm is a high-efficiency coding method that encodes wavelet coefficients using a bit-plane representation. Wavelet transforms facilitate this process by ensuring strong energy compaction. Nevertheless, the residual structure within the high-frequency coefficients must still be addressed. Regions with high energy tend to preserve their spatial location and shape across subbands, and imperfect representation by the transform causes energy leakage into HF bands. In contrast, smooth regions are accurately modeled by the LF basis, producing compact low-frequency coefficient distributions. This pattern persists across multiple decomposition levels, reinforcing the similarity between subbands.

Understanding image characteristics—such as edges, textures, and smooth regions—helps explain these hierarchical relationships. Zero-tree coding techniques leverage this structure by combining interband dependencies with efficient representation of insignificant coefficients, achieving high compression performance.

The Embedded Zero Tree (EZW) algorithm, introduced by Shapiro, encodes both significant and insignificant coefficients. At low bitrates, the significance map may dominate the bitstream; however, this map exhibits substantial

redundancy that can be efficiently compressed. If a parent coefficient has low magnitude, its descendants are likely insignificant as well, leading to the formation of zero-trees. When the probability of this event approaches one, the associated information content becomes minimal, requiring very few bits to encode. In cases where one or more descendants are significant, an isolated zero symbol is produced, which occurs infrequently but requires additional bits.

EZW generates hierarchical, embedded bitstreams that support progressive image transmission. This means that the image quality improves as more bits are received. For two transmission rates $R_1 > R_2$, the bitstream at rate R_2 forms a prefix of the bitstream at rate R_1.

Such embedded coding provides several advantages: precise control over the output bitrate; optimal image reconstruction for any truncated bitstream; efficient adaptation to channels with varying bandwidth; fast preview and browsing of large image databases.

In the EZW algorithm, coefficients are evaluated using a threshold T. Coefficients with magnitudes greater than T are classified as significant, while the others are considered insignificant. The scanning process proceeds from low-frequency to high-frequency subbands. Binary symbols indicate the significance state and spatial relationship of coefficients, including a root zero-tree symbol that represents an insignificant coefficient and all of its descendants. After significance coding, the corresponding bits are entropy-coded, typically using arithmetic coding. If the available bitrate allows, the threshold T is reduced by half and the procedure is repeated. Figure 1 illustrates this iterative process. Zero-trees prevent repeated transmission of insignificant coefficients, and once a coefficient becomes significant, its quantized magnitude is transmitted.

SPIHT is an enhanced version of the EZW algorithm, designed to provide both highly efficient compression and optimal progressive transmission. Its primary objective is to maximize reconstructed image quality for a given bitrate while maintaining efficient decoding.

A key feature of SPIHT is its embedded (nested) coding structure. The encoder produces a sequence of bitstreams in which any shorter stream of m bits corresponds exactly to the first m bits of a longer stream, ensuring optimal performance at all truncation points.

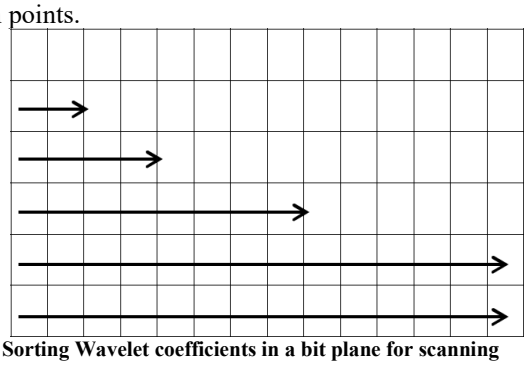

Fig. 1. Sorting Wavelet coefficients in a bit plane for scanning

In practical implementations, the encoding process usually ends when the iteration index reaches $n = 0$; however, the encoder may terminate earlier if required. In such cases, tThe least informative components—specifically, the lower-order bits of the wavelet coefficients—are intentionally omitted during transmission, thereby decreasing computational burden. This deliberate and controlled data reduction resembles scalar quantization in principle; however, it achieves better performance because the wavelet coefficients are arranged and encoded in an optimized hierarchical sequence. While the encoder has the capability to send every coefficient in full precision, the decoding procedure can be stopped at any stage once the reconstructed image reaches a predefined quality threshold. This threshold may either be user-defined or adaptively selected according to time or resource constraints.

Recursive image compression methods originate from the framework of iterative function systems (IFS). A pioneering role in this field was played by Michael Barnsley, who showed that IFS theory could effectively describe image structures. Expanding upon these ideas, Jacquin proposed a fractal compression algorithm based on partitioning the image into blocks. In this scheme, the image is segmented into non-overlapping range blocks and a larger set of overlapping domain blocks.

For each range block, the algorithm searches for an appropriate domain block and determines an affine transformation that provides the closest approximation. Instead of storing pixel data directly, the image is represented as the fixed point of a contractive mapping. Through iterative application of this mapping, the image progressively converges to its reconstructed form. By virtue of Banach's fixed-point theorem, any contractive transformation guarantees convergence to a single, unique fixed point corresponding to the decoded image. The primary limitation of fractal compression lies in its computational intensity, as the encoding phase demands exhaustive comparisons between numerous pairs of blocks. Nevertheless, fractal-based techniques remain a meaningful alternative to traditional Fourier-based compression methods. At present, fractal regression approaches are mainly used in archival systems, such as digital encyclopedias, where data is encoded once and decoded repeatedly. They are also applied in specialized domains, including satellite imagery and medical imaging. In medical diagnostics, for example, fractal-based processing of X-ray images can enhance the visibility of clinically relevant details.

For discrete image data, one of the most widely used wavelet-based techniques is the Mallat algorithm, named after Stéphane Mallat. This method decomposes an image into two parts: a set of high-frequency components that capture abrupt intensity changes, and a low-frequency, downsampled approximation of the original image. The

decomposition is performed using a pair of filters, each operating at half the original spatial resolution. Typically, finite impulse response (FIR) filters are employed, where pixel values within a small neighborhood are multiplied by predefined coefficients, summed, and then shifted across the image to generate successive outputs. Although wavelets are not explicitly present in the algorithm, they arise implicitly through repeated application of these filters to an image containing a single impulse.

Since images are two-dimensional, filtering is carried out in both horizontal and vertical directions. This procedure is applied iteratively, with each stage using the previously obtained low-frequency approximation as its input. The resulting detail subbands usually consist of sharp edges and large regions with near-zero values. If fine details are not essential, coefficients with small magnitudes can be discarded, leading to significant compression gains. The original image is recovered using the inverse Mallat transform, which applies the same filter pairs in reverse order.

Recursive and wavelet-based image compression methods thus provide powerful tools for reducing data size while preserving essential visual information. Given $M \times N$ image from $\Omega = (0, M] \times (0, N] \in R^2$ pixels .

In this case, each pixel is associated with a specific area:

$$\{(\xi, \eta) \in \Omega, \xi \in (j-1, j], \eta \in (i-1,]\} \tag{1}$$

The specified point $(i, j) \in \Omega$ is the upper right corner of the pixel area.

We will look for a mapping of the type: $A: X \to X$.

The main point of this compression is that the original image is given by the $z = f(\xi, \eta)$ function , using some compression map, the given points correspond to the original image. The set of vectors with X, R^n, where $n = M \times N$ are the coordinates of the vectors, corresponding to the pixel

$x_i = f(1 + (i-1) \bmod N, \left\lfloor \frac{i}{N} \right\rfloor + 1)$, $i = 1, \ldots, n$, $\lfloor \ \rfloor$ - this means rounding.

To create such a display, it is necessary to split the image into a set of R data blocks , $R_1, R_2, \ldots, R_r$ where . $R_i \subseteq \Omega,\ i = 1, \ldots, r$

Д – blocks $D_1, D_2 \ \ldots D_1, (D_i \subseteq \Omega)$- a square (possibly intersecting) $2B \times 2B$ fragment of pixels and $\cup_i \ D_i = \Omega$

Given function $F_i : R_i \to R,\ i = 1, \ldots, r$ are the brightness characteristics of each pixel. The average brightness value - (ξ, η)

$$F_i(\xi, \eta) = s_i \cdot \frac{1}{4} \sum (f(\xi^{\sim}, \eta^{\sim}) : (\xi^{\sim}, \eta^{\sim}) \in W^{-1}((\xi - 1, \xi] \times (\eta - 1, \eta]) \cap Z^2) \tag{2}$$

S_i- the contrast parameter, and this is O_i for brightness.

$$x_k^{'} = F_{i(k)}(\xi(k), \eta(k))$$

де $\xi(k) = 1 + (k-1)N$, $\eta(k) = \left\lfloor \frac{k}{N} \right\rfloor + 1$ the vector coordinates are the data of the upper right corner of the K pixel area and $i(k)$ the number of the block containing the point.

In recurrent coding, the map is chosen such that the distance between a vector and its image is minimal according to some metric.

According to Banach's theorem, this is a compressed map where the complete metric space is the metric:

$$x_A = A(x_A) \tag{3}$$

converges to a single fixed transformation point $x_0 \in X$ for any point $x_A \in X$:

It is shown that the reconstructed image only needs to remember the reflection.

$$\rho(x, y) \le L\rho(A(x),)A(y)), \tag{4}$$

where ρ is a metric with norm l^{∞} . For convergence of the interaction process, it is sufficient that the parameters S_i are $S_i < 1 \ \forall i = 1, \ldots, r$. N is the norm of the indices. It should be noted $L < 1$ that is not mandatory for recovery. N - norm does not always coincide exactly with the image $A(x)$. N - norm, but if some conditions are met on the operator A , then it is x_A guaranteed to be similar to a value close x to as and $A(x)$.

The following theorem [7] gives an estimate between (x) and $A(x)$.

Let us show this theorem:

$$\rho(x, x_A) \le \frac{1}{1-L} \rho(x, A(x)),$$

where $L < 1$ is the Lipschitz constant that compresses the operator A in a certain metric ρ.

For a Euclidean metric, the Lipschitz constant of the operator can be calculated as follows [8].

$$L = \max_{1 \le j \le d} \sqrt{\frac{1}{4} \sum (s_i^2 : j(i) = j)} \tag{5}$$

In the algorithm proposed below, the Euclidean metric is used as a parameter $A(x)$close to the original image vector. In other words, it is necessary to find A a mapping that is minimal $\rho(A(x), x)$. This is equivalent to the problem of finding, for each block R, $R_i, i = 1, \ldots, r$, a triple such $(W_i, D_{j(i)}, F_i)$.

When solving this problem for any variant and $D_{j(i)}$ and W_i, the least squares linear regression method can be used to select the optimal parameters s_i and o_i.

$$z_1 = f(\xi_1, \eta_1), z_2 f(\xi_2, \eta_2), \ldots, z_k = f(\xi_{B^2}, \eta_{B^2}), \tag{6}$$

$$z_1' = \frac{1}{4}\sum (f(\xi^{\sim}, \eta^{\sim}) : (\xi^{\sim}, \eta^{\sim}) \in W^{-1}((\xi_1 - 1, \xi_1] \times (\eta_1 - 1, \eta_1]) \cap Z^2) \tag{7}$$

$$z_k' = \frac{1}{4}\sum (f(\xi^{\sim}, \eta^{\sim}) : (\xi^{\sim}, \eta^{\sim}) \in W^{-1}((\xi_k - 1, \xi_k] \times (\eta_k - 1, \eta_k]) \cap Z^2) \tag{8}$$

$(\xi_k, \eta_k) \in R_i$ and Z_k' average brightness reading of the prototype pixel(ξ_k, η_k) $\forall k = 1, \ldots, B^2$.

For analysis, it is necessary to consider a function that is minimized at the point where the private derivative of the parameter $\sigma^2_{R_i}$ and the parameter S_i and O_i is equal to the minimum value 0. After differentiation and transformation for the parameters, we see a system of 2 simultaneous equations with 2 unknowns, from which we obtain S_i and O_i the following equation.

$$s_i = \frac{\left[B^2 \cdot \left(\sum_{k=1}^{B^2} z_k' z_k\right) - \left(\sum_{k=1}^{B^2} z_k'\right) \cdot \left(\sum_{k=1}^{B^2} z_k\right)\right]}{B^2 \cdot \sum_{k=1}^{B^2} z_k'^2 - \left(\sum_{k=1}^{B^2} z_k'\right)^2} \; o_i = \frac{1}{B^2}\left(\sum_{k=1}^{B^2} z_k - s_i \cdot \sum_{k=1}^{B_2} z_k'\right) \tag{9}$$

which coincides with the equation of the original block. If $s_i = 0$, то $o_i = \frac{1}{B^2}\sum_{k=1}^{B^2} z_k$

The idea of these algorithms is to classify blocks *D* and *R* and to find blocks *D* that are in the same class as the ranked regions. This is done as follows.

The output block is divided into 4 parts. The mean value A_i and variance of pixels are calculated for the part V_i. Then the block is classified according to the following principles. Let's define three main types of blocks. Type 1 - $A_1 \geq A_2 \geq A_3 \leq A_4$, Type 2 - $A_1 \geq A_2 \geq A_4 \geq A_3$, Type 3 - $A_1 \geq A_4 \geq A_2 \geq A_3$.

It is obvious that any block can be transformed into a form corresponding to any of these types by means of the appropriate affine quadratic transformation. After determining the 3 main classes, the blocks are classified by variants.

Thus, each of the three classes has 24 subclasses, for a total of 72 classes; the search for D blocks close to R blocks is carried out by a specific search class.

Genetic algorithms are algorithmic approaches to solving extreme single criteria selection problems based on modeling the main factors of the evolutionary development of a population.

If we use a genetic algorithm (GA) to find the optimal solution for each element x∈X, the optimization space looks like this: shown as a vector $b \in$ B of *N* binary letters *A* = {0,1}, where $B = A^N$. *X* denotes the optimization space, which must also consist of a finite number of elements.

Population Π = ($\chi^1, \chi^2, \ldots, \chi$M) is a vector in the space B^M, and its coordinates are the genotypic individuals of this population.

The next stage. Transition to GA is a transition from the current generation to another, Π_{t+1} that is Π_t, obtaining a new population. The operator participates in the construction of the next individual of the new population Select crossover, mutation and random selection: *Select*: $B^M \rightarrow$ {1,…,M} selects the identification number of the parent when generating the next generation.

To define this algorithm, c must specify the intersection operator *Cross*: B×B $\rightarrow$ B×B and the mutation operator *Mut*: B→B.

The crossover $(\chi', \tau') = Cross(\chi, \tau)$ operation consists in randomly selecting position *j*, which is evenly distributed from 1 to *N* - 1, and the result has the following form

$\chi' = (\chi_1, \chi_2, \ldots, \chi_j, \tau_{j+1}, \ldots, \tau_N)$, $\tau' = (\tau_1, \tau_2, \ldots, \tau_j, \chi_{j+1}, \ldots, \chi_N)$.

The shutter effect is set using the basic probability parameters P_{cross} that trigger this operator (otherwise it does not change at all).

The mutation operator at each position of the argument replaces its contents with A, a random element of the lifetime alphabet, chosen based on a uniform distribution, with a given probability.

The basic function of this problem is replaced by a non-negative genotype compatibility function in GA, $\Phi(\chi)$, $\chi \in$B.

The algorithmic process itself is a sequential change of generations, at each stage of which the population is replenished with pairs of descendants from individual individuals of the population according to the formula

$$(\chi_k^{t+1}, \chi_{k+1}^{t+1}) = Mut(Cross(\chi_{Select(\Pi_t)}^t, \chi_{Select(\Pi_t)}^t)), (\chi_k^{t+1}, \chi_{k+1}^{t+1})$$

Individuals with the lowest fitness level in the population are Π_t in Π_{t+1} That is, individuals are removed after mating, and then in pairs, and mutations accumulate. If you change the parameters of mutation and crossover probability, you will see permission to tune the behavior of GA and adapt to specific tasks.

The presented scheme of the GA algorithm is applied to the fractal compression problem. As the genotype of the GA, it is convenient to take a vector whose components are the number of encoding pixel coordinates and the affine transformation of the region of the $D_{j(i)}$ original image defined on the toroidal Surface W_i. There are 8 ways to glue a square to a square: rotate it from all sides or reflect and rotate it from all sides. So, 3 bits are enough to encode this transformation. Let's have a fitness function like this

$$\Phi = \frac{1}{1 + \sum([f(\xi,\eta) - F_i(\xi,\eta)]^2 : (\xi,\eta) \in R_i \cap Z^2))} \quad (10)$$

This function satisfies the GA requirement (non-negative) and is suitable for the Crossingoffer selection operator $\chi^{i,t}$.

In this operator, each individual from the population Π_{t+1}has its own probability, which is calculated

$$Pselect\ (\chi^{i,t}) = \frac{\Phi\ (\chi^{i,t})}{\sum_j\ \Phi\ (\chi^{j,t})} \quad (11)$$

The specified chromosomes identifying a given genotype are represented in such a way that any vector in the decision space is acceptable, and the fitness is never zero. The mutation operator in the algorithm is standard, while the crossover operator has been modified as follows:

Crossovers can only occur at bifurcation points in the binary coordinate representation.

As described above, the main disadvantage of this compression method is that the computational effort required to encode the image is enormous. Programs implementing this method use both traditional acceleration techniques, such as integer arithmetic and assembly language routines, as well as specialized approaches that employ only one type of affine transformation or various strategies for segmenting the display area.

Below is a schematic diagram of two algorithms that can significantly reduce the computational complexity for several classes of images. The parameters of the first algorithm are the level of loss during encoding and the minimum block size. This algorithm ensures uniform encoding quality across the entire image.

The second algorithm uses the number of blocks as its main parameter, which directly affects the computational complexity but does not influence the accuracy of encoding individual image regions.

Results

A series of flower images, consisting of 102 datasets of different flower species, was selected for the experiment. Each category contains at least 40 images taken from various angles and under different lighting conditions. The dataset was sourced from Kaggle [3]. For the experiment, 100 color classes were selected, each represented by a single image.

The test image was divided into 8×8 blocks of 1,500 pixels each. A total of 20,000 copies were generated as auxiliary parts.

Table 1 shows the results of encoding the test images using models with different architectures.

Figures 3 - 4 show graphs of the experimental results. In these graphs, the similarity measure (SSIM) has been scaled by a factor of 83% for visualization purposes.

The figures display a series of embossments in blue, while similarity values are shown in red.

Figures 3 - 4 show graphs of the experimental results. In these graphs, the similarity measure (SSIM) has been scaled by a factor of 83% for visualization purposes.

Fig. 2. Test image, 200×300 pixels, data size: 180,128 bytes

The figures display a series of embossments in blue, while similarity values are shown in red.

Based on the experimental results, the most suitable model was selected according to similarity levels and encoding time, as the compression ratios of all algorithms were approximately the same. The MobileNet V2 model demonstrated the highest speed.

Other models showed similar performance: the first model was slightly faster, while the second achieved slightly better similarity scores.

The number of such fragments in the database plays an important role in the developed encoding method. In this study, we examined how the quality of the decoded image and the encoding time depend on the number of fragments requiring similar images.

Table 2 presents the results of this study.

Table 1

Model Comparison

Model	Encoding time, seconds	Output size, bytes	Compression percentage, %	Similarity, SSIM
MobileNet V2	33.03	3829	97.87	0.53
Inception V3	50.46	3777	97.9	0.84
NASNet-A (mobile)	79.47	3773	97.9	0.76

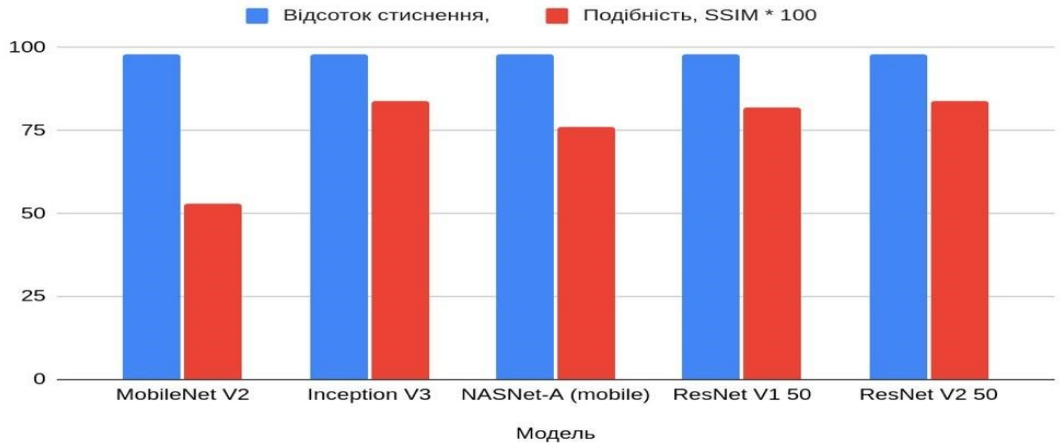

Fig. 3. Compression level and similarity graph

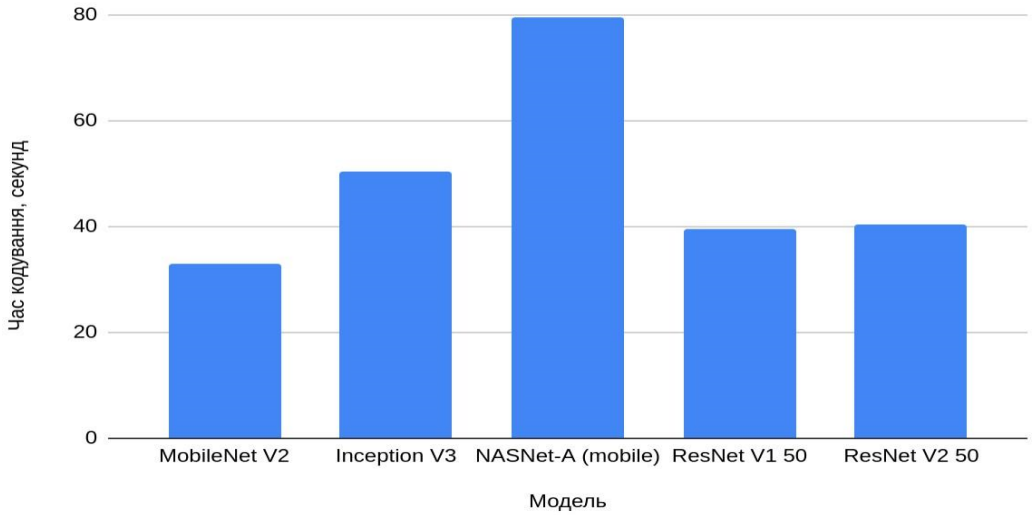

Fig. 4. Encoding time graph

Table 2

Experimental Results

Number of clusters seconds	Encoding time, seconds	Output size, bytes	Compression percentage, %	Similarity, SSIM
20 000	15.55	3733	97.93	0.84
50 000	27.56	4069	97.74	0.86
100 000	44.67	4677	97.40	0.88
150 000	63.36	4905	97.28	0.89
200 000	82.83	5005	97.22	0.89

The experimental results indicate that the proposed approach is less effective than the standard JPG and JPEG formats when evaluated by absolute compression efficiency. Nevertheless, it demonstrates a very high similarity coefficient. Concurrently, the approach proposed in this study demonstrates a noticeably higher compression efficiency when benchmarked against conventional reference methods.

It should be noted, however, that the reconstructed image generated by the developed algorithm occupies a file size roughly 3.5 times greater than the initial JPG version and approximately 3.6 times larger than the corresponding original JPEG file.

A comparative evaluation was performed against JPEG and MJPEG compression schemes. While JPEG and MJPEG achieved the highest image similarity index of 0.73, the developed method reached a value of 0.97, indicating a significantly improved ability to retain the visual and structural characteristics of the original image.

Throughout this research, the image compression process was thoroughly examined. In accordance with the stated problem, the research objectives and tasks were formulated, and appropriate methods for their implementation were identified and applied.

When examining how image processing impacts compression outcomes, it becomes clear that each graphics editor applies unique transformations tailored to specific visual effects. This variability makes it challenging to anticipate the precise modifications introduced into an image. Similarly, determining which editor provides the most effective compromise between compression efficiency and visual fidelity is far from straightforward.

The task of comparing different compression techniques is further complicated by the absence of uniform datasets and evaluation standards. Certain specialized fields, such as security-sensitive applications, may require bespoke compression approaches to meet their specific needs. The findings also indicate that, in general, lossless compression methods excel at maintaining the original data's integrity compared to lossy alternatives.

As a result of this investigation, a software application was developed that enables rapid and efficient data compression through the combined use of content-independent compression techniques.

Reference

1. Versatile video coding (VVC) **:** ITU-T Recommendation H.266 ; ISO/IEC MPEG-I Part 3. – ITU-T ; ISO/IEC JTC 1/SC 29/WG 11, 2023. – URL: https://www.itu.int/rec/T-REC-H.266

2. High efficiency video coding (HEVC) : ITU-T Recommendation H.265 ; ISO/IEC 23008-2. – ITU-T ; ISO/IEC JTC 1/SC 29/WG 11, 2013. – URL: https://www.itu.int/rec/T-REC-H.265

3. Image parameter values for HDR-TV standards for production and international programme exchange : ITU-R Recommendation BT.2100-2. – ITU-R, 2023. – URL: https://www.itu.int/rec/R-REC-BT.2100

4. Methodology for the subjective assessment of the quality of television pictures **:** ITU-R Recommendation BT.500-15. – ITU-R, 2022. – URL: https://www.itu.int/rec/R-REC-BT.500

5. JPEG XS image coding system : ISO/IEC 21122. – ISO/IEC JTC 1/SC 29/WG 1, 2021. – URL: https://www.iso.org/standard/77596.html

6. JPEG XL image coding system : ISO/IEC 18181. – ISO/IEC JTC 1/SC 29/WG 1, 2021. – URL: https://www.iso.org/standard/75114.html

7. Information technology — JPEG 2000 image coding system: core coding system : ISO/IEC 15444-1:2020 ; ITU-T Rec. H.800. – ISO/IEC JTC 1/SC 29/WG 1, 2020. – URL: https://www.iso.org/standard/78321.html

8. Bross B., Wang Y.-K., Ye Y., Liu S.-L., Chen J., Sullivan G. J., Ohm J. Overview of the versatile video coding (VVC) standard and its applications // *IEEE Transactions on Circuits and Systems for Video Technology*. – 2021. – Vol. 31, No. 10. – P. 3736–3764. – DOI: https://doi.org/10.1109/TCSVT.2021.3101953

9. Yang W., Huang H., Hu Y., Duan L.-Y., Liu J. Video coding for machines: compact visual representation compression for intelligent collaborative analytics // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2024. – Vol. 46, No. 7. – P. 5174–5191. – DOI: https://doi.org/10.1109/TPAMI.2024.3367293

10. Zhou L., Ruan C., Ling N., Chen Z., Wang W., Jiang W. TVC: tokenized video compression with ultra-low bit rate // *Visual Intelligence*. – 2025. – Vol. 3. – Art. 25. – DOI: https://doi.org/10.1007/s44267-025-00098-7